User's Manual

32

RX26T Group

User's Manual: Hardware

RENESAS 32-Bit MCU
RX Family／RX200 Series

Renesas Electronics
www.renesas.com

Rev.1.10 Aug 2023

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 or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving 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, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. 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 shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended 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; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

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

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

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

(Rev.5.0-1 October 2020)

Corporate Headquarters

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

Contact Information

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

Trademarks

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

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

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

1. Precaution against Electrostatic Discharge (ESD)
   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on
   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state
   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins
   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.
5. Clock signals
   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin
   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses
   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products
   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

How to Use This Manual

1. Purpose and Target Readers

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

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

The revision history summarizes the locations of revisions and additions. It does not list all revisions. Refer to the text of the manual for details.

The following documents apply to the RX26T Group. Make sure to refer to the latest versions of these documents. The latest versions of the listed documents are available from the Renesas Electronics website.

Document Type	Description	Document Title	Document No.
Datasheet	Hardware overview and electrical characteristics	RX26T Group Datasheet	R01DS0407EJ
User's Manual: Hardware	Hardware specifications (pin assignments, memory maps, peripheral specifications, electrical characteristics, and timing charts) and operation description Note: Refer to the application notes for details on using peripheral functions.	RX26T Group User's Manual: Hardware	This User's manual
User's Manual: Software	Description of CPU instruction set	RX Family RXv3 Instruction Set Architecture User's Manual: Software	R01US0316EJ
Application Note	Notes on Printed Circuit Board Patterns	RX Family Hardware Design Guide	R01AN1411EJ
	Examples of register initial setting	RX26T Group Initial Setting Examples	R01AN6567EJ
	Information on using peripheral functions and application examples Sample programs	Available from Renesas Electronics website.	
Renesas Technical Update	Product specifications, updates on documents, etc.		

2. Notation of Numbers and Symbols

Each register description includes a bit chart, illustrating the arrangement of bits, and a table of bits, describing the meanings of the bit settings. The standard format and notation for bit charts and tables are described below.

X.X.X ... Register

Address(es): xxxx xxxxh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	...[1:0]		...4	—	—	—	...0
Value after reset:	x	0	0	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	...0	.....	0: ..... 1: Setting prohibited (3)	R/W (1)
b3 to b1	—	Reserved (2)	These bits are read as 0. The write value should be 0.	R/W
b4	...4	.....	0: 1:	R
b6, b5	...[1:0]	.....	0 0: ..... 0 1: ..... Settings other than above are prohibited. (3)	R/(W)*1
b7	—	Reserved	The read value is undefined. Writing to this bit has no effect.	R

(1) R/W: The bit or field is readable and writable.
R/(W): The bit or field is readable and writable. However, writing to this bit or field has some limitations. For details on the limitations, see the description or notes of respective registers.
R: The bit or field is readable. Writing to this bit or field has no effect.

(2) Reserved.
Use the specified value when writing to this bit or field; otherwise, the correct operation is not guaranteed.

(3) Setting prohibited. The correct operation is not guaranteed if such a setting is performed.

3. List of Abbreviations and Acronyms

Abbreviation	Full Form
ACIA	Asynchronous Communications Interface Adapter
bps	bits per second
CRC	Cyclic Redundancy Check
DMA	Direct Memory Access
DMAC	Direct Memory Access Controller
GSM	Global System for Mobile Communications
Hi-Z	High Impedance
IEBus	Inter Equipment Bus
I/O	Input / Output
IrDA	Infrared Data Association
LSB	Least Significant Bit
MSB	Most Significant Bit
NC	Non-Connect
PLL	Phase Locked Loop
PWM	Pulse Width Modulation
SFR	Special Function Registers
SIM	Subscriber Identity Module
UART	Universal Asynchronous Receiver / Transmitter
VCO	Voltage Controlled Oscillator

Contents

120-MHz, 32-bit RX MCU, on-chip FPU, 709 CoreMark, Supportive of 5V power supply, up to 512-KB flash memory, up to 64-KB SRAM, 16-KB data flash memory, various communications interfaces including CAN FD, Simultaneous sampling with 3 units of 12-bit A/D converter (up to 7 channels), Analog comparator (6 channels), 120 MHz PWM (4 channels for 3-phase complementary, 2 channels for 5-phase complementary, 10 channels for single-phase complementary), 4-channel high-resolution PWM with resolution of 260 ps at the minimum, Encryption functions

Features

■ 32-bit RXv3 CPU core
- Maximum operating frequency: 120 MHz
  Capable of 709 CoreMark in operation at 120 MHz
- A collective register bank save function is available.
- Supports the memory protection unit (MPU)
- JTAG and FINE (one-line) debugging interfaces

■ Low-power design and architecture
- Operation from a single 2.7- to 5.5-V supply
- Three low-power modes

■ On-chip code flash memory
- Supports versions with up to 512 Kbytes of ROM
- Operation at 120 MHz (with no waiting)
- User code is programmable by on-board or off-board programming.
- Programming/erasing as background operations (BGOs)
- A dual-bank structure allows exchanging the start-up bank.

■ On-chip data flash memory
- 16 Kbytes, reprogrammable up to 100,000 times
- Programming/erasing as background operations (BGOs)

■ On-chip SRAM
- 64 K/48Kbytes of SRAM (with no waiting)

■ Data transfer
- DMACAa: 8 channels
- DTCb: 1 channel

■ ELC
- Module operation can be initiated by event signals without using interrupts
- Linked operation between modules is possible when the CPU is in sleep mode

■ Reset and supply management
- Power-on reset (POR)
- Low voltage detection (LVD) with voltage settings

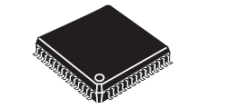
■ Clock functions
- The main clock oscillator is connectable to an 8- to 24-MHz external crystal resonator and usable as the PLL reference clock.
- Internal 240-kHz LOCO and HOCO selectable from 16, 18, and 20 MHz
- 120-kHz clock for the IWDTa

■ Independent watchdog timer
- 120-kHz IWDT-dedicated on-chip oscillator clock operation

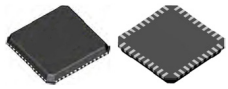
■ Useful functions for IEC60730 compliance
- Oscillation-stoppage detection, functions for self-diagnosis and detection of disconnection for the A/D converter, clock frequency accuracy measurement circuit, independent watchdog timer, RAM test-assisting function by DOC, and CRCA, etc.
- Register write protection function can protect values in important registers against overwriting.

■ Encryption functions (Trusted Secure IP Lite)
- 128- or 256-bit key length of AES for ECB, CBC, GCM, others
- True random number generator
- Unauthorized access to the encryption engine is disabled and imposture and falsification of information are prevented
- Safe management of keys

■ Up to 83 pins for general I/O ports
- 5-V tolerance, open drain, input pull-up, switchable driving ability, and retention of the port output



PLQP0100KB-B 14 × 14 mm, 0.5 mm pitch
PLQP0080KB-B 12 × 12 mm, 0.5 mm pitch
PLQP0064KB-C 10 × 10 mm, 0.5 mm pitch
PLQP0048KB-B 7 × 7 mm, 0.5 mm pitch



PWQN0064KF-A 9 × 9 mm, 0.5 mm pitch
PWQN0048KC-A 7 × 7 mm, 0.5 mm pitch

■ Various communications interfaces
- CAN FD: Compliant with ISO11898-1:2015, standard frame and extended frame (1 channel)
- SCIk and SCIh with multiple functionalities (up to 4 channels)
  Choose from among asynchronous mode, clock-synchronous mode, smart-card interface mode, simplified SPI, simplified I^2C, and extended serial mode.
- Up to three RSCIs with Manchester encoding and HBS functionality
- I^2C bus interface (RIICa) for transfer at up to 400 kbps (fast mode), capable of SMBus operation (1 channel)
- I3C bus interface (RI3C) for the single data rate (SDR) mode (1 channel)
- RSPId (1 channel) for transfer at up to 30 Mbps

■ Up to 29 extended-function timers
- 32-bit (products with 64 Kbytes of RAM) or 16-bit (products with 48 Kbytes of RAM) GPTWa (8 channels): operation at 120 MHz, input capture, output compare, PWM waveforms: 10 output channels in single-phase complementary PWM mode/3 output channels in 3-phase complementary PWM mode/2 output channels in 5-phase complementary PWM mode, phase-counting mode, linkage with comparator (counting operation, PWM negate control)
- 16-bit MTU3d (9 channels): operation at 120 MHz, input capture, output compare, PWM waveforms: 2 output channels in 3-phase complementary PWM mode, phase-counting mode
- 8-bit TMRb (8 channels)
- 16-bit CMT (4 channels)

■ High-resolution PWM waveform generation circuit (HRPWM): 4 channels
- Controlling the timing of rising or falling of the PWM output waveform for 32-bit GPTWa is realized with minimum of 260 ps resolution (in operation at 120 MHz)

■ 12-bit A/D converter (S12ADH)
- Products with 64 Kbytes of RAM
  Three 12-bit units of sample-and-hold circuit included:
  Unit 0 (4 channels for 3 sample-and-hold circuits),
  Unit 1 (4 channels for 3 sample-and-hold circuits),
  Unit 2 (14 channels)
- Products with 48 Kbytes of RAM
  Two 12-bit units of sample-and-hold circuit included:
  Unit 0 (7 channels for 3 sample-and-hold circuits),
  Unit 2 (8 channels)

■ Analog Comparator (CMPCa): 6 channels

■ 12-bit D/A converter: 2 channels
- Usable as a reference voltage for the analog comparator

■ Temperature sensor for measuring temperature within the chip

■ Recommended operating temp. range (Topr)
- D-version: –40°C to +85°C
- G-version: –40°C to +105°C

1. Overview

1.1 Outline of Specifications

Table 1.1 lists the specifications in outline, and Table 1.2 gives a comparison of the functions of products in different packages.

Table 1.1 shows the outline of maximum specifications. The peripheral functions and the number of their channels vary depending on the number of pins of the package, and the RAM capacity. For details, see Table 1.2, Comparison of Functions for Different Packages.

Table 1.1 Outline of Specifications (1/9)

Classification	Module/Function	Description
CPU	CPU	• Maximum operating frequency: 120 MHz • 32-bit RX CPU (RXv3) • Minimum instruction execution time: One instruction per state (cycle of the system clock) • Address space: 4-Gbyte linear • Register set of the CPU General purpose: Sixteen 32-bit registers Control: Ten 32-bit registers Accumulator: Two 72-bit registers • 113 instructions (products with 64 Kbytes of RAM), 111 instructions (products with 48 Kbytes of RAM) Standard provided instructions: 111 Basic instructions: 77 Single precision floating point instructions: 11 DSP instructions: 23 Instructions for register bank save function: 2 (only supported by products with 64 Kbytes of RAM) • Addressing modes: 11 • Data arrangement Instructions: Little endian Data: Selectable as little endian or big endian • On-chip 32-bit multiplier: 32 × 32 → 64 bits • On-chip divider: 32/32 → 32 bits • Barrel shifter: 32 bits
	FPU	• Single-precision (32-bit) floating-point number • Data types and floating-point exceptions in conformance with the IEEE754 standard
	Register bank save function	• Fast collective saving and restoration of the values of CPU registers • 16 save register banks
Memory	Code flash memory	• Capacity: 512 Kbytes, 256 Kbytes, 128 Kbytes • 120 MHz No-wait access • On-board programming: Three types • Instructions are executable only for the program stored in the TM target area by using the Trusted Memory (TM) function and protection against data reading is realized. • A dual-bank structure allows programming during reading or exchanging the start-up areas
	Data flash memory	• Capacity: 16 Kbytes • Programming/erasing: 100,000 times
	Unique ID	• 12-byte unique ID for the device
	RAM	• Capacity: 64 Kbytes, 48 Kbytes • 120 MHz No-wait access • SED (single error detection)
Operating modes		• Operating modes by the mode-setting pins at the time of release from the reset state Single-chip mode Boot mode (SCI interface) Boot mode (FINE interface) • Selection of operating mode by register setting Single-chip mode • Endian selectable

Table 1.1 Outline of Specifications (2/9)

Classification	Module/Function	Description
Clock	Clock generation circuit	• Main clock oscillator, low-speed/high-speed on-chip oscillator, PLL frequency synthesizer, and IWDT-dedicated on-chip oscillator • The peripheral module clocks can be set to frequencies above that of the system clock. • Main-clock oscillation stoppage detection • Separate frequency-division and multiplication settings for the system clock (ICLK), peripheral module clocks (PCLKA, PCLKB, PCLKC, PCLKD), and flash-IF clock (FCLK) The CPU and other bus masters run in synchronization with the system clock (ICLK): Up to 120 MHz Peripheral modules of MTU (Internal peripheral bus), GPTW (Internal peripheral bus), HRPWM (Internal peripheral bus), RSPI, RSPIA, RSCI, RI3C, and the ECC function control registers in the CAN FD module run in synchronization with PCLKA, which operates at up to 120 MHz. Other peripheral modules run in synchronization with PCLKB: Up to 60 MHz MTU (counter reference clocks), GPTW (counter reference clocks), and HRPWM (reference clocks) are synchronized with PCLKC: Up to 120 MHz ADCLK in the S12AD runs in synchronization with PCLKD: Up to 60 MHz Flash IF run in synchronization with the flash-IF clock (FCLK): Up to 60 MHz • Multiplication is possible with using the high-speed on-chip oscillator (HOCO) as a reference clock of the PLL circuit
Reset		Eight types of reset • RES# pin reset: Generated when the RES# pin is driven low. • Power-on reset: Generated when the RES# pin is driven high and VCC rises. • Voltage-monitoring 0 reset: Generated when VCC falls. • Voltage-monitoring 1 reset: Generated when VCC falls. • Voltage-monitoring 2 reset: Generated when VCC falls. • Independent watchdog timer reset: Generated when the independent watchdog timer underflows, or a refresh error occurs. • Watchdog timer reset: Generated when the watchdog timer underflows, or a refresh error occurs. • Software reset: Generated by register setting.
Power-on reset		If the RES# pin is at the high level when power is supplied, an internal reset is generated. After VCC has exceeded the voltage detection level and the specified period has elapsed, the reset is cancelled.
Voltage detection circuit (LVDA)		Monitors the voltage being input to the VCC pin and generates an internal reset or internal interrupt. • Voltage detection circuit 0 Capable of generating an internal reset The option-setting memory can be used to select enabling or disabling of the reset. Voltage detection level: Selectable from two different levels • Voltage detection circuits 1 and 2 Voltage detection level: Selectable from five different levels Digital filtering (1/2, 1/4, 1/8, and 1/16 LOCO frequency) Capable of generating an internal reset • Two types of timing are selectable for release from reset An internal interrupt can be requested. • Detection of voltage rising above and falling below thresholds is selectable. • Maskable or non-maskable interrupt is selectable Voltage detection monitoring Event linking
Low power consumption	Low power consumption facilities	• Module stop function • Three low power consumption modes Sleep mode, all-module clock stop mode, and software standby mode
Interrupt	Interrupt controller (ICUG)	• Interrupt vectors: 256 • External interrupts: 16 (pins IRQ0 to IRQ15) • Software interrupts: 2 sources • Non-maskable interrupts: 7 sources • Sixteen levels specifiable for the order of priority • Method of interrupt source selection: The interrupt vectors consist of 256 vectors, with 128 having fixed sources. The other 133 sources can be assigned to the remaining 128 vectors as required.

Table 1.1 Outline of Specifications (3/9)

Classification	Module/Function	Description
DMA	DMA controller (DMACAa)	• 8 channels • Three transfer modes: Normal transfer, repeat transfer, and block transfer • Request sources: Software trigger, external interrupts, and interrupt requests from peripheral functions
	Data transfer controller (DTCb)	• Three transfer modes: Normal transfer, repeat transfer, and block transfer • Request sources: External interrupts and interrupt requests from peripheral functions
I/O ports	Programmable I/O ports	• I/O ports for the 100-pin LFQFP I/O pins: 82 Input pin: 1 Pull-up resistors: 82 Open-drain outputs: 82 5-V tolerance: 2 Large current output: 15 • I/O ports for the 80-pin LFQFP I/O pins: 62 Input pin: 1 Pull-up resistors: 62 Open-drain outputs: 62 5-V tolerance: 2 Large current output: 14 • I/O ports for the 64-pin LFQFP, 64-pin HWQFN I/O pins: 49 Input pin: 1 Pull-up resistors: 49 Open-drain outputs: 49 5-V tolerance: 2 Large current output: 14 • I/O ports for the 48-pin LFQFP, 48-pin HWQFN I/O pins: 37 Input pin: 1 Pull-up resistors: 37 Open-drain outputs: 37 5-V tolerance: 2 Large current output: 13
Event link controller (ELC)		• Event signals such as interrupt request signals can be interlinked with the operation of functions such as timer counting, eliminating the need for intervention by the CPU to control the functions. • 183 internal event signals can be freely combined for interlinked operation with connected functions. • Event signals from peripheral modules can be used to change the states of output pins (of ports B and E). • Changes in the states of pins (of ports B and E) being used as inputs can be interlinked with the operation of peripheral modules.
Timers	8-bit timers (TMRb)	• (8 bits × 2 channels) × 4 units • Select from among seven internal clock signals (PCLKB/1, PCLKB/2, PCLKB/8, PCLKB/32, PCLKB/64, PCLKB/1024, PCLKB/8192) and one external clock signal • Capable of output of pulse trains with desired duty cycles or of PWM signals • The 2 channels of each unit can be cascaded to create a 16-bit timer • Generation of triggers for A/D converter conversion • Capable of generating baud-rate clocks for SCI5, SCI6, and SCI12 • Event linking by the ELC
	Compare match timer (CMT)	• (16 bits × 2 channels) × 2 units • Select from among four internal clock signals (PCLKB/8, PCLKB/32, PCLKB/128, PCLKB/512) • Event linking by the ELC
	Compare match timer W (CMTW)	• (32 bits × 1 channel) × 2 units • Compare-match, input-capture input, and output-comparison output are available. • Select from among four internal clock signals (PCLKB/8, PCLKB/32, PCLKB/128, PCLKB/512) Interrupt requests can be output in response to compare-match, input-capture, and output-comparison events.

Table 1.1 Outline of Specifications (4/9)

Classification	Module/Function	Description
Timers	Watchdog timer (WDTA)	• 14 bits × 1 channel • Select from among 6 counter-input clock signals (PCLKB/4, PCLKB/64, PCLKB/128, PCLKB/512, PCLKB/2048, PCLKB/8192)
	Independent watchdog timer (IWDTa)	• 14 bits × 1 channel • Counter-input clock: IWDT-dedicated on-chip oscillator • Dedicated clock/1, dedicated clock/16, dedicated clock/32, dedicated clock/64, dedicated clock/128, dedicated clock/256 • Window function: The positions where the window starts and ends are specifiable (the window defines the timing with which refreshing is enabled and disabled). • Event linking by the ELC
	Multifunction timer pulse unit 3 (MTU3d)	• 9 channels (16 bits × 9 channels) • Maximum of 28 pulse-input/output and 3 pulse-input possible • Select from among 14 counter-input clock signals for each channel (PCLKC/1, PCLKC/2, PCLKC/4, PCLKC/8, PCLKC/16, PCLKC/32, PCLKC/64, PCLKC/256, PCLKC/1024, MTCLKA, MTCLKB, MTCLKC, MTCLKD, MTIOC1A) 11 of the signals are available for channels 1, 3, 4, 12 are available for channel 2, and 10 are available for channel 5. • 43 output compare/input capture registers • Counter clear operation (synchronous clearing by compare match/input capture) • Simultaneous writing to multiple timer counters (TCNT) • Simultaneous register input/output by synchronous counter operation • Buffered operation • Support for cascade-connected operation • 45 interrupt sources • Automatic transfer of register data • Pulse output mode Toggle/PWM/complementary PWM/reset-synchronized PWM • Complementary PWM output mode Outputs non-overlapping waveforms for controlling 3-phase inverters Automatic specification of dead times PWM duty cycle: Selectable as any value from 0% to 100% Delay can be applied to requests for A/D conversion. Non-generation of interrupt requests at peak or trough values of counters can be selected. Double buffer configuration • Reset synchronous PWM mode Three phases of positive and negative PWM waveforms can be output with desired duty cycles. • Phase-counting mode: 16-bit mode (channels 1 and 2); 32-bit mode (channels 1 and 2) • Counter functionality for dead-time compensation • Generation of triggers for A/D converter conversion The timing of the generation of requests to start A/D conversion can be monitored by an external pin. • A/D conversion start triggers can be skipped • Digital filter function for signals on the input capture and external counter clock pins • Event linking by the ELC • Internal peripheral bus clock: PCLKA • Counter reference clock: PCLKC • Frequency ratio: PCLKA to PCLKC = 1: N (N = 1 or 2)
	Port output enable 3 (POE3D)	• Control of the high-impedance state of the MTU/GPTW's waveform output pins, and control of switching to the general I/O port pin • 7 pins for input from signal sources: POE0, POE4, POE8, POE9, POE10, POE11, POE12 • Initiation by detection of short-circuited outputs (detection of PWM outputs that have become an active level simultaneously) • Initiation by comparator detection/oscillation stop detection/software • Additional programming of output control target pins is enabled

Table 1.1 Outline of Specifications (5/9)

Classification	Module/Function	Description
Timers	General PWM timer (GPTWa)	• 32 bits × 8 channels (products with 64 Kbytes of RAM) • 16 bits × 8 channels (products with 48 Kbytes of RAM) • Counting up or down (sawtooth-wave), counting up and down (triangle-wave) selectable for all channels • Clock sources independently selectable for each channel • 2 input/output pins per channel • 2 output compare/input capture registers per channel • For the 2 output compare/input capture registers of each channel, 4 registers are provided as buffer registers and are capable of operating as comparison registers when buffering is not in use. • In output compare operation, buffer switching can be at peaks or troughs, enabling the generation of laterally asymmetrically PWM waveforms. • Registers for setting up frame intervals on each channel (with capability for generating interrupts on overflow or underflow) • Generation of dead times in PWM operation • Capable of synchronous start, stop, or clearing of counter for any channel • Capable of a start, stop, clearing, or up-/down-counting of the counter supporting maximum of 8 ELC events • Capable of a start, stop, clearing, or up-/down-counting of the counter supporting input level comparison • Capable of a start, stop, clearing, or up-/down-counting of the counter supporting maximum of 4 external triggers • Output pin disabling function by a dead time error or a short circuit detection among output pins • Capable of generating conversion start triggers for the A/D converters as well as monitoring external pins for a start timing of conversion. • Capable of outputting events, such as compare-match from A to F and overflow/ underflow, to ELC • Capable of using noise filter of input capture • Periodic counting • Internal peripheral bus clock: PCLKA • Counter reference clock: PCLKC • Frequency ratio: PCLKA to PCLKC = 1: N (N = 1 or 2)
	High resolution PWM (HRPWM)	• Capable of generating the PWM waveform that is generated by GPTW0 through GPTW3 with resolution of minimum of 260 ps.
	Port output enable for GPTW (POEG)	• Controlling the output disable for GPTW waveform output • Initiation by input level detection of GTETRG pins • Initiation by output disable request from GPTW • Initiation by detection of comparator interrupt request • Initiation by detection of oscillation stop or by software

Table 1.1 Outline of Specifications (6/9)

Classification	Module/Function	Description
Communication function	Serial communications interfaces (SCIk, SCIh)	• 4 channels SCIk: SCI1, SCI5, SCI6 SCIh: SCI12 • SCIk, SCIh Serial communications modes: Asynchronous, clock synchronous, and smart-card interface Multi-processor function On-chip baud rate generator allows selection of the desired bit rate Choice of LSB-first or MSB-first transfer Average transfer rate clock can be input from TMR timers for SCI5, SCI6, and SCI12 Start-bit detection: Level or edge detection is selectable. Simple I^2C Simple SPI 7, 8, 9-bit transfer mode Bit rate modulation Double-speed mode Data match detection (SCI12 is not supported) Event linking by the ELC (supported by SCI5 only) RXD input signal select function (supported by SCI5 only) • SCIk Only Data match detection Adjustment of the timing of sampling of the RXD signals • SCIh Only Supports the serial communications protocol, which contains the start frame and information frame Supports the LIN format
	Serial communications interfaces (RSCI)	• 3 channels (RSCI8, RSCI9, RSCI11) • Serial communications modes: Asynchronous, clock synchronous, and smart-card interface • Multi-processor function • On-chip baud rate generator allows selection of the desired bit rate • Choice of LSB-first or MSB-first transfer • Start-bit detection: Level or edge detection is selectable. • Simple I^2C • Simple SPI • 9-bit transfer mode • Bit rate modulation • Double-speed mode • Event linking by the ELC (only RSCI11) • RXD input signal select function • Supports the serial communications protocol, which contains the start frame and information frame • Supports the LIN format (RSCI9, RSCI11) • Data can be transmitted or received in sequence by the 32-byte FIFO buffers of the transmission and reception unit (only RSCI11) • Manchester encoding is supported. • RSCI has some home bus system (HBS) functionality. • Data match detection • Adjustment of the timing of sampling of the RXD signals
	I^2C bus interface (RIICa)	• 1 channel Communication formats I^2C bus format/SMBus format Supports the multi-master Max. transfer rate: 400 kbps • Event linking by the ELC

Table 1.1 Outline of Specifications (7/9)

Classification	Module/Function	Description
Communication function	I3C bus interface (RI3C)	• 1 channel Supports the SDR mode Supports the legacy I^2C message Supports the multi-master • Event linking by the ELC
	CAN FD module (CANFD)	• 1 channel • Compliance with the ISO11898-1:2015 specification (standard frame and extended frame)
	Serial peripheral interface (RSPId)	• 1 channel • RSPI transfer facility Using the MOSI (master out, slave in), MISO (master in, slave out), SSL (slave select), and RSPCK (RSPI clock) signals enables serial transfer through SPI operation (four lines) or clock-synchronous operation (three lines) Capable of handling serial transfer as a master or slave • Data formats Switching between MSB first and LSB first The number of bits in each transfer can be changed to any number of bits from 8 to 16, 20, 24, or 32 bits. 128-bit buffers for transmission and reception Up to four frames can be transmitted or received in a single transfer operation (with each frame having up to 32 bits) Transmit/receive data can be swapped in byte units • Buffered structure Double buffers for both transmission and reception • RSPCK can be stopped with the receive buffer full for master reception. • Event linking by the ELC
	Serial peripheral interface (RSPIA)	• 1 channel • RSPI transfer facility Using the MOSI (master out, slave in), MISO (master in, slave out), SSL (slave select), and RSPCK (RSPI clock) signals enables serial transfer through SPI operation (four lines) or clock-synchronous operation (three lines) Capable of handling serial transfer as a master or slave • Data formats Switching between MSB first and LSB first The number of bits in each transfer can be changed to any number of bits from 8 to 16, 20, 24, or 32 bits. 128-bit buffers for transmission and reception Up to four frames can be transmitted or received in a single transfer operation (with each frame having up to 32 bits) Transmit/receive data can be swapped in byte units • Buffered structure The transmission and reception sections have 4-stage and 32-bit-wide FIFO buffers for the sequential transmission and reception of data. • RSPCK can be stopped with the receive buffer full for master reception. • Event linking by the ELC • Communications protocol: RSPIA supports the Texas Instruments Synchronous Serial Protocol (TI SSP).

Table 1.1 Outline of Specifications (8/9)

Classification	Module/Function	Description
12-bit A/D converter (S12ADH) (Products with 64 Kbytes of RAM)		• 12 bits (4 channels × 2 units, 14 channels × 1 unit) • 12-bit resolution • Minimum conversion time 0.9 µs per channel (when ADCLK operates at 60 MHz) • Operating mode Scan mode (single scan mode, continuous scan mode, or 3 group scan mode) Group A priority control (only for 3 group scan mode) • Sample-and-hold function channel-dedicated sample-and-hold function (unit 0 × 3 channels, unit 1 × 3 channels) included • Sampling variable Sampling time can be set up for each channel. • Conversion function in order of arbitrarily selected channels (Serial conversion of the same channel cannot be allowed) • Double trigger mode (A/D conversion data duplicated) • Three ways to start A/D conversion Software trigger, synchronous trigger (MTU, GPTW, TMR, ELC), external trigger • Prioritization in group scanning can be controlled among group A, B, and C. • Digital comparison Method: Comparison to detect voltages above or below thresholds and window comparison Measurement: Comparison of two results of conversion or comparison of a value in the comparison register and a result of conversion • Self-diagnostic function • Detection of analog input disconnection • Event linking by the ELC • Input signal amplification function by the programmable gain amplifier (unit 0 × 3 channels, unit 1 × 3 channels)
12-bit A/D converter (S12ADH) (Products with 48 Kbytes of RAM)		• 12 bits (7 channels × 1 unit, 8 channels × 1 unit) • 12-bit resolution • Minimum conversion time 0.9 µs per channel (when ADCLK operates at 60 MHz) • Operating mode Scan mode (single scan mode, continuous scan mode, or 3 group scan mode) Group A priority control (only for 3 group scan mode) • Sample-and-hold function channel-dedicated sample-and-hold function (unit 0 × 3 channels) included • Sampling variable Sampling time can be set up for each channel. • Conversion function in order of arbitrarily selected channels (Serial conversion of the same channel cannot be allowed) • Double trigger mode (A/D conversion data duplicated) • Three ways to start A/D conversion Software trigger, synchronous trigger (MTU, GPTW, TMR, ELC), external trigger • Prioritization in group scanning can be controlled among group A, B, and C. • Digital comparison Method: Comparison to detect voltages above or below thresholds and window comparison Measurement: Comparison of two results of conversion or comparison of a value in the comparison register and a result of conversion • Self-diagnostic function • Detection of analog input disconnection • Event linking by the ELC
12-bit D/A converter (R12DAb)		• 2 channels • 12-bit resolution • Output voltage: 0 V to AVCC2 • Capable of providing as a reference voltage for comparator • Event linking by the ELC
Comparator C (CMPCa)		• 6 channels • Function to compare the reference voltage and the analog input voltage • Reference voltage is selectable from 4 inputs • Analog input voltage is selectable from 4 inputs • Digital filtering

Table 1.1 Outline of Specifications (9/9)

Classification	Module/Function	Description
Temperature sensor		• 1 channel • Relative precision: ±1.0°C • The voltage of the temperature is converted into a digital value by the 12-bit A/D converter (unit 2).
Arithmetic unit for trigonometric functions (TFUv2)		• Calculation of sine, cosine, arctangent, and hypotenuse Simultaneous calculation of sine and cosine Simultaneous calculation of arctangent and hypotenuse
Safety	Memory protection unit (MPU)	• Protection area: Eight areas (max.) can be specified in the range from 0000 0000h to FFFF FFFFh. • Minimum protection unit: 16 bytes • Reading from, writing to, and enabling the execution access can be specified for each area. • An access exception occurs when the detected access is not in the permitted area.
	Trusted Memory (TM) Function	• Programs in the TM target area in the code flash memory are protected against reading • Instruction fetching by the CPU is the only form of access to these areas when the TM function is enabled.
	Register write protection function	• Protects important registers from being overwritten for in case a program runs out of control.
	CRC calculator (CRCA)	• Generation of CRC codes for 8-/32-bit data 8-bit data Selectable from the following three polynomials $X^8 + X^2 + X + 1$, $X^{16} + X^{15} + X^2 + 1$, $X^{16} + X^{12} + X^5 + 1$ 32-bit data Selectable from the following two polynomials $X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$, $X^{32}+X^{28}+X^{27}+X^{26}+X^{25}+X^{23}+X^{22}+X^{20}+X^{19}+X^{18}+X^{14}+X^{13}+X^{11}+X^{10}+X^9+X^8+X^6+1$ • Generation of CRC codes for use with LSB-first or MSB-first communications is selectable
	Main clock oscillation stop detection function	• Main clock oscillation stop detection: Available
	Clock frequency accuracy measurement circuit (CAC)	• Monitors the clock output from the main clock oscillator, low- and high-speed on-chip oscillators, IWDT-dedicated on-chip oscillator, and PCLKB.
	Data operation circuit (DOCA)	• This handles the comparison, addition, subtraction, comparison in terms of which is larger or smaller, or window comparison of 32-bit values.
Encryption functions	Trusted Secure IP (TSIP-Lite)	• Access management circuit • Encryption engine 128- or 256-bit key sizes of AES Block cipher mode of operation: GCM, ECB, CBC, CMAC, XTS, CTR, GCTR • Hash function • True random number generator • Prevention from illicit copying of a key
Operating frequency		Up to 120 MHz
Power supply voltage		VCC = 2.7 to 5.5V AVCC0 = AVCC1 = AVCC2 = 3.0 to 5.5V (VCC ≤ AVCC0 = AVCC1 = AVCC2) VSS = AVSS0 = AVSS1 = AVSS2 = 0V
Operating temperature		D-version: –40 to +85°C G-version: –40 to +105°C
Package		100-pin LFQFP 0.5 mm pitch 80-pin LFQFP 0.5 mm pitch 64-pin LFQFP 0.5 mm pitch 64-pin HWQFN 0.5 mm pitch 48-pin LFQFP 0.5 mm pitch 48-pin HWQFN 0.5 mm pitch
Debugging interfaces		• JTAG and One-line FINE interfaces

Table 1.2 Comparison of Functions for Different Packages (1/2)

Module/Functions		RX26T Group					
		Products with 64 Kbytes of RAM				Products with 48 Kbytes of RAM	
		100 Pins	80 Pins	64 Pins	48 Pins	64 Pins	48 Pins
CPU	Register Bank Save Function	Available				Not available	
Code Flash Memory	Code flash memory capacity	128 Kbytes/256 Kbytes/512 Kbytes				128 Kbytes/256 Kbytes	
	Dual bank function	Available*1				Not available	
	BGO function	Available					
Data Flash Memory		16 Kbytes					
RAM		64 Kbytes				48 Kbytes	
External interrupts	NMI	Available					
	IRQ	16 channels	13 channels	12 channels	10 channels	12 channels	10 channels
DMA	DMA controller	Available					
	Data transfer controller	Available					
Timers	Multifunction timer pulse unit 3	9 channels (Ch. 0 to 7, Ch. 9)					
	General PWM timer	32 bits × 8 channels				16 bits × 8 channels	
	High resolution PWM	4 channels				Not available	
	Port output enable 3	Available					
	Port Output Enable for GPTW	Available					
	8-bit timer	2 channels × 4 units					
	Compare match timer	2 channels × 2 units					
	Compare match timer W	1 channel × 2 units					
	Watchdog timer	Available					
	Independent watchdog timer	Available					
Communication functions	Serial communications interfaces (SCIk)	Ch. 1, 5, and 6					
	Serial communications interfaces (SCIh)	Ch. 12					
	Serial communications interfaces (RSCI)	Ch. 8, 9, and 11				Not available	
	I2C bus interfaces (RIIC)	1 channel					
	I3C bus interfaces (RI3C)	1 channel				Not available	
	Serial peripheral interface (RSPI)	Ch. 0					
	Serial peripheral interface (RSPIA)	Ch. 0				Not available	
	CAN FD module (CANFD)	1 channel					
12-bit A/D Converter		Unit 0: 4 channels Unit 1: 4 channels Unit 2: 14 channels	Unit 0: 4 channels Unit 1: 4 channels Unit 2: 11 channels	Unit 0: 4 channels Unit 1: 4 channels Unit 2: 7 channels	Unit 0: 4 channels Unit 1: 1 channels Unit 2: 5 channels	Unit 0: 7 channels Unit 2: 8 channels	Unit 0: 5 channels Unit 2: 5 channels
	3 channels simultaneous sampling function	Available (unit 0, 1)			Available (unit 0)		
	Programmable gain amplifier	6 channels			4 channels	Not available	
Comparator C		6 channels			5 channels	4 channels	
D/A converter		2 channels					
Temperature sensor		1 channel					

Table 1.2 Comparison of Functions for Different Packages (2/2)

Module/Functions	RX26T Group					
	Products with 64 Kbytes of RAM				Products with 48 Kbytes of RAM	
	100 Pins	80 Pins	64 Pins	48 Pins	64 Pins	48 Pins
Arithmetic unit for trigonometric functions (TFU)	Available					
CRC calculator (CRC)	Available					
Data operation circuit (DOC)	Available					
Clock frequency accuracy measurement circuit (CAC)	Available					
Trusted Secure IP (TSIP-Lite)	Available/Not available				Not available	
Event link controller (ELC)	Available					
Packages	100-pin LFQFP	80-pin LFQFP	64-pin LFQFP 64-pin HWQFN	48-pin LFQFP 48-pin HWQFN	64-pin LFQFP	48-pin LFQFP

Note 1. The products with 512 Kbytes of the code flash memory only support this function.

1.2 List of Products

Table 1.3 is a list of products, and Figure 1.1 shows how to read the product part no.

Table 1.3 List of Products (1/3)

Group	Part No.	Package	ROM Capacity	RAM Capacity	CANFD	TSIP-Lite	Operating temperature
RX26T (D-version)	R5F526T9ADFP	PLQP0100KB-B	128 Kbytes	64 Kbytes	Available*1	Not available	-40 to 85°C
	R5F526T9BDFP	PLQP0100KB-B	128 Kbytes	64 Kbytes	Available*1	Available	-40 to 85°C
	R5F526TBADFP	PLQP0100KB-B	256 Kbytes	64 Kbytes	Available*1	Not available	-40 to 85°C
	R5F526TBBDFP	PLQP0100KB-B	256 Kbytes	64 Kbytes	Available*1	Available	-40 to 85°C
	R5F526TBCDFP	PLQP0100KB-B	256 Kbytes	64 Kbytes	Available	Not available	-40 to 85°C
	R5F526TBDDFP	PLQP0100KB-B	256 Kbytes	64 Kbytes	Available	Available	-40 to 85°C
	R5F526TFADFP	PLQP0100KB-B	512 Kbytes	64 Kbytes	Available*1	Not available	-40 to 85°C
	R5F526TFBDFP	PLQP0100KB-B	512 Kbytes	64 Kbytes	Available*1	Available	-40 to 85°C
	R5F526TFCDFP	PLQP0100KB-B	512 Kbytes	64 Kbytes	Available	Not available	-40 to 85°C
	R5F526TFDDFP	PLQP0100KB-B	512 Kbytes	64 Kbytes	Available	Available	-40 to 85°C
	R5F526T9ADFN	PLQP0080KB-B	128 Kbytes	64 Kbytes	Available*1	Not available	-40 to 85°C
	R5F526T9BDFN	PLQP0080KB-B	128 Kbytes	64 Kbytes	Available*1	Available	-40 to 85°C
	R5F526TBADFN	PLQP0080KB-B	256 Kbytes	64 Kbytes	Available*1	Not available	-40 to 85°C
	R5F526TBBDFN	PLQP0080KB-B	256 Kbytes	64 Kbytes	Available*1	Available	-40 to 85°C
	R5F526TBCDFN	PLQP0080KB-B	256 Kbytes	64 Kbytes	Available	Not available	-40 to 85°C
	R5F526TBDDFN	PLQP0080KB-B	256 Kbytes	64 Kbytes	Available	Available	-40 to 85°C
	R5F526TFADFN	PLQP0080KB-B	512 Kbytes	64 Kbytes	Available*1	Not available	-40 to 85°C
	R5F526TFBDFN	PLQP0080KB-B	512 Kbytes	64 Kbytes	Available*1	Available	-40 to 85°C
	R5F526TFCDFN	PLQP0080KB-B	512 Kbytes	64 Kbytes	Available	Not available	-40 to 85°C
	R5F526TFDDFN	PLQP0080KB-B	512 Kbytes	64 Kbytes	Available	Available	-40 to 85°C
	R5F526T8ADFM	PLQP0064KB-C	128 Kbytes	48 Kbytes	Available*1	Not available	-40 to 85°C
	R5F526T9ADFM	PLQP0064KB-C	128 Kbytes	64 Kbytes	Available*1	Not available	-40 to 85°C
	R5F526T9BDFM	PLQP0064KB-C	128 Kbytes	64 Kbytes	Available*1	Available	-40 to 85°C
	R5F526TAADFM	PLQP0064KB-C	256 Kbytes	48 Kbytes	Available*1	Not available	-40 to 85°C
	R5F526TACDFM	PLQP0064KB-C	256 Kbytes	48 Kbytes	Available	Not available	-40 to 85°C
	R5F526TBADFM	PLQP0064KB-C	256 Kbytes	64 Kbytes	Available*1	Not available	-40 to 85°C
	R5F526TBBDFM	PLQP0064KB-C	256 Kbytes	64 Kbytes	Available*1	Available	-40 to 85°C
	R5F526TBCDFM	PLQP0064KB-C	256 Kbytes	64 Kbytes	Available	Not available	-40 to 85°C
	R5F526TBDDFM	PLQP0064KB-C	256 Kbytes	64 Kbytes	Available	Available	-40 to 85°C
	R5F526TFADFM	PLQP0064KB-C	512 Kbytes	64 Kbytes	Available*1	Not available	-40 to 85°C
	R5F526TFBDFM	PLQP0064KB-C	512 Kbytes	64 Kbytes	Available*1	Available	-40 to 85°C
	R5F526TFCDFM	PLQP0064KB-C	512 Kbytes	64 Kbytes	Available	Not available	-40 to 85°C
	R5F526TFDDFM	PLQP0064KB-C	512 Kbytes	64 Kbytes	Available	Available	-40 to 85°C
	R5F526T9ADND	PWQN0064KF-A	128 Kbytes	64 Kbytes	Available*1	Not available	-40 to 85°C
	R5F526T9BDND	PWQN0064KF-A	128 Kbytes	64 Kbytes	Available*1	Available	-40 to 85°C
	R5F526TBADND	PWQN0064KF-A	256 Kbytes	64 Kbytes	Available*1	Not available	-40 to 85°C
	R5F526TBBDND	PWQN0064KF-A	256 Kbytes	64 Kbytes	Available*1	Available	-40 to 85°C
	R5F526TBCDND	PWQN0064KF-A	256 Kbytes	64 Kbytes	Available	Not available	-40 to 85°C
	R5F526TBDDND	PWQN0064KF-A	256 Kbytes	64 Kbytes	Available	Available	-40 to 85°C
	R5F526TFADND	PWQN0064KF-A	512 Kbytes	64 Kbytes	Available*1	Not available	-40 to 85°C
	R5F526TFBDND	PWQN0064KF-A	512 Kbytes	64 Kbytes	Available*1	Available	-40 to 85°C
	R5F526TFCDND	PWQN0064KF-A	512 Kbytes	64 Kbytes	Available	Not available	-40 to 85°C
	R5F526TFDDND	PWQN0064KF-A	512 Kbytes	64 Kbytes	Available	Available	-40 to 85°C

Table 1.3 List of Products (2/3)

Group	Part No.	Package	ROM Capacity	RAM Capacity	CANFD	TSIP-Lite	Operating temperature
RX26T (D-version)	R5F526T8ADFL	PLQP0048KB-B	128 Kbytes	48 Kbytes	Available*1	Not available	-40 to 85°C
	R5F526T9ADFL	PLQP0048KB-B	128 Kbytes	64 Kbytes	Available*1	Not available	-40 to 85°C
	R5F526T9BDFL	PLQP0048KB-B	128 Kbytes	64 Kbytes	Available*1	Available	-40 to 85°C
	R5F526TAADFL	PLQP0048KB-B	256 Kbytes	48 Kbytes	Available*1	Not available	-40 to 85°C
	R5F526TACDFL	PLQP0048KB-B	256 Kbytes	48 Kbytes	Available	Not available	-40 to 85°C
	R5F526TBADFL	PLQP0048KB-B	256 Kbytes	64 Kbytes	Available*1	Not available	-40 to 85°C
	R5F526TBBDFL	PLQP0048KB-B	256 Kbytes	64 Kbytes	Available*1	Available	-40 to 85°C
	R5F526TBCDFL	PLQP0048KB-B	256 Kbytes	64 Kbytes	Available	Not available	-40 to 85°C
	R5F526TBDDFL	PLQP0048KB-B	256 Kbytes	64 Kbytes	Available	Available	-40 to 85°C
	R5F526TFADFL	PLQP0048KB-B	512 Kbytes	64 Kbytes	Available*1	Not available	-40 to 85°C
	R5F526TFBDFL	PLQP0048KB-B	512 Kbytes	64 Kbytes	Available*1	Available	-40 to 85°C
	R5F526TFCDFL	PLQP0048KB-B	512 Kbytes	64 Kbytes	Available	Not available	-40 to 85°C
	R5F526TFDDFL	PLQP0048KB-B	512 Kbytes	64 Kbytes	Available	Available	-40 to 85°C
	R5F526T9ADNE	PWQN0048KC-A	128 Kbytes	64 Kbytes	Available*1	Not available	-40 to 85°C
	R5F526T9BDNE	PWQN0048KC-A	128 Kbytes	64 Kbytes	Available*1	Available	-40 to 85°C
	R5F526TBADNE	PWQN0048KC-A	256 Kbytes	64 Kbytes	Available*1	Not available	-40 to 85°C
	R5F526TBBDNE	PWQN0048KC-A	256 Kbytes	64 Kbytes	Available*1	Available	-40 to 85°C
	R5F526TBCDNE	PWQN0048KC-A	256 Kbytes	64 Kbytes	Available	Not available	-40 to 85°C
	R5F526TBDDNE	PWQN0048KC-A	256 Kbytes	64 Kbytes	Available	Available	-40 to 85°C
	R5F526TFADNE	PWQN0048KC-A	512 Kbytes	64 Kbytes	Available*1	Not available	-40 to 85°C
	R5F526TFBDNE	PWQN0048KC-A	512 Kbytes	64 Kbytes	Available*1	Available	-40 to 85°C
	R5F526TFCDNE	PWQN0048KC-A	512 Kbytes	64 Kbytes	Available	Not available	-40 to 85°C
	R5F526TFDDNE	PWQN0048KC-A	512 Kbytes	64 Kbytes	Available	Available	-40 to 85°C
RX26T (G-version)	R5F526T9AGFP	PLQP0100KB-B	128 Kbytes	64 Kbytes	Available*1	Not available	-40 to 105°C
	R5F526T9BGFP	PLQP0100KB-B	128 Kbytes	64 Kbytes	Available*1	Available	-40 to 105°C
	R5F526TBAGFP	PLQP0100KB-B	256 Kbytes	64 Kbytes	Available*1	Not available	-40 to 105°C
	R5F526TBBGFP	PLQP0100KB-B	256 Kbytes	64 Kbytes	Available*1	Available	-40 to 105°C
	R5F526TBCGFP	PLQP0100KB-B	256 Kbytes	64 Kbytes	Available	Not available	-40 to 105°C
	R5F526TBDGFP	PLQP0100KB-B	256 Kbytes	64 Kbytes	Available	Available	-40 to 105°C
	R5F526TFAGFP	PLQP0100KB-B	512 Kbytes	64 Kbytes	Available*1	Not available	-40 to 105°C
	R5F526TFBGFP	PLQP0100KB-B	512 Kbytes	64 Kbytes	Available*1	Available	-40 to 105°C
	R5F526TFCGFP	PLQP0100KB-B	512 Kbytes	64 Kbytes	Available	Not available	-40 to 105°C
	R5F526TFDGFP	PLQP0100KB-B	512 Kbytes	64 Kbytes	Available	Available	-40 to 105°C
	R5F526T9AGFN	PLQP0080KB-B	128 Kbytes	64 Kbytes	Available*1	Not available	-40 to 105°C
	R5F526T9BGFN	PLQP0080KB-B	128 Kbytes	64 Kbytes	Available*1	Available	-40 to 105°C
	R5F526TBAGFN	PLQP0080KB-B	256 Kbytes	64 Kbytes	Available*1	Not available	-40 to 105°C
	R5F526TBBGFN	PLQP0080KB-B	256 Kbytes	64 Kbytes	Available*1	Available	-40 to 105°C
	R5F526TBCGFN	PLQP0080KB-B	256 Kbytes	64 Kbytes	Available	Not available	-40 to 105°C
	R5F526TBDGFN	PLQP0080KB-B	256 Kbytes	64 Kbytes	Available	Available	-40 to 105°C
	R5F526TFAGFN	PLQP0080KB-B	512 Kbytes	64 Kbytes	Available*1	Not available	-40 to 105°C
	R5F526TFBGFN	PLQP0080KB-B	512 Kbytes	64 Kbytes	Available*1	Available	-40 to 105°C
	R5F526TFCGFN	PLQP0080KB-B	512 Kbytes	64 Kbytes	Available	Not available	-40 to 105°C
	R5F526TFDGFN	PLQP0080KB-B	512 Kbytes	64 Kbytes	Available	Available	-40 to 105°C
	R5F526T8AGFM	PLQP0064KB-C	128 Kbytes	48 Kbytes	Available*1	Not available	-40 to 105°C
	R5F526T9AGFM	PLQP0064KB-C	128 Kbytes	64 Kbytes	Available*1	Not available	-40 to 105°C
	R5F526T9BGFM	PLQP0064KB-C	128 Kbytes	64 Kbytes	Available*1	Available	-40 to 105°C
	R5F526TAAGFM	PLQP0064KB-C	256 Kbytes	48 Kbytes	Available*1	Not available	-40 to 105°C
	R5F526TACGFM	PLQP0064KB-C	256 Kbytes	48 Kbytes	Available	Not available	-40 to 105°C

Table 1.3 List of Products (3/3)

Group	Part No.	Package	ROM Capacity	RAM Capacity	CANFD	TSIP-Lite	Operating temperature
RX26T (G-version)	R5F526TBAGFM	PLQP0064KB-C	256 Kbytes	64 Kbytes	Available*1	Not available	-40 to 105°C
	R5F526TBBGFM	PLQP0064KB-C	256 Kbytes	64 Kbytes	Available*1	Available	-40 to 105°C
	R5F526TBCGFM	PLQP0064KB-C	256 Kbytes	64 Kbytes	Available	Not available	-40 to 105°C
	R5F526TBDGFM	PLQP0064KB-C	256 Kbytes	64 Kbytes	Available	Available	-40 to 105°C
	R5F526TFAGFM	PLQP0064KB-C	512 Kbytes	64 Kbytes	Available*1	Not available	-40 to 105°C
	R5F526TFBGFM	PLQP0064KB-C	512 Kbytes	64 Kbytes	Available*1	Available	-40 to 105°C
	R5F526TFCGFM	PLQP0064KB-C	512 Kbytes	64 Kbytes	Available	Not available	-40 to 105°C
	R5F526TFDGFM	PLQP0064KB-C	512 Kbytes	64 Kbytes	Available	Available	-40 to 105°C
	R5F526T9AGND	PWQN0064KF-A	128 Kbytes	64 Kbytes	Available*1	Not available	-40 to 105°C
	R5F526T9BGND	PWQN0064KF-A	128 Kbytes	64 Kbytes	Available*1	Available	-40 to 105°C
	R5F526TBAGND	PWQN0064KF-A	256 Kbytes	64 Kbytes	Available*1	Not available	-40 to 105°C
	R5F526TBBGND	PWQN0064KF-A	256 Kbytes	64 Kbytes	Available*1	Available	-40 to 105°C
	R5F526TBCGND	PWQN0064KF-A	256 Kbytes	64 Kbytes	Available	Not available	-40 to 105°C
	R5F526TBDGND	PWQN0064KF-A	256 Kbytes	64 Kbytes	Available	Available	-40 to 105°C
	R5F526TFAGND	PWQN0064KF-A	512 Kbytes	64 Kbytes	Available*1	Not available	-40 to 105°C
	R5F526TFBGND	PWQN0064KF-A	512 Kbytes	64 Kbytes	Available*1	Available	-40 to 105°C
	R5F526TFCGND	PWQN0064KF-A	512 Kbytes	64 Kbytes	Available	Not available	-40 to 105°C
	R5F526TFDGND	PWQN0064KF-A	512 Kbytes	64 Kbytes	Available	Available	-40 to 105°C
	R5F526T8AGFL	PLQP0048KB-B	128 Kbytes	48 Kbytes	Available*1	Not available	-40 to 105°C
	R5F526T9AGFL	PLQP0048KB-B	128 Kbytes	64 Kbytes	Available*1	Not available	-40 to 105°C
	R5F526T9BGFL	PLQP0048KB-B	128 Kbytes	64 Kbytes	Available*1	Available	-40 to 105°C
	R5F526TAAGFL	PLQP0048KB-B	256 Kbytes	48 Kbytes	Available*1	Not available	-40 to 105°C
	R5F526TACGFL	PLQP0048KB-B	256 Kbytes	48 Kbytes	Available	Not available	-40 to 105°C
	R5F526TBAGFL	PLQP0048KB-B	256 Kbytes	64 Kbytes	Available*1	Not available	-40 to 105°C
	R5F526TBBGFL	PLQP0048KB-B	256 Kbytes	64 Kbytes	Available*1	Available	-40 to 105°C
	R5F526TBCGFL	PLQP0048KB-B	256 Kbytes	64 Kbytes	Available	Not available	-40 to 105°C
	R5F526TBDGFL	PLQP0048KB-B	256 Kbytes	64 Kbytes	Available	Available	-40 to 105°C
	R5F526TFAGFL	PLQP0048KB-B	512 Kbytes	64 Kbytes	Available*1	Not available	-40 to 105°C
	R5F526TFBGFL	PLQP0048KB-B	512 Kbytes	64 Kbytes	Available*1	Available	-40 to 105°C
	R5F526TFCGFL	PLQP0048KB-B	512 Kbytes	64 Kbytes	Available	Not available	-40 to 105°C
	R5F526TFDGFL	PLQP0048KB-B	512 Kbytes	64 Kbytes	Available	Available	-40 to 105°C
	R5F526T9AGNE	PWQN0048KC-A	128 Kbytes	64 Kbytes	Available*1	Not available	-40 to 105°C
	R5F526T9BGNE	PWQN0048KC-A	128 Kbytes	64 Kbytes	Available*1	Available	-40 to 105°C
	R5F526TBAGNE	PWQN0048KC-A	256 Kbytes	64 Kbytes	Available*1	Not available	-40 to 105°C
	R5F526TBBGNE	PWQN0048KC-A	256 Kbytes	64 Kbytes	Available*1	Available	-40 to 105°C
	R5F526TBCGNE	PWQN0048KC-A	256 Kbytes	64 Kbytes	Available	Not available	-40 to 105°C
	R5F526TBDGNE	PWQN0048KC-A	256 Kbytes	64 Kbytes	Available	Available	-40 to 105°C
	R5F526TFAGNE	PWQN0048KC-A	512 Kbytes	64 Kbytes	Available*1	Not available	-40 to 105°C
	R5F526TFBGNE	PWQN0048KC-A	512 Kbytes	64 Kbytes	Available*1	Available	-40 to 105°C
	R5F526TFCGNE	PWQN0048KC-A	512 Kbytes	64 Kbytes	Available	Not available	-40 to 105°C
	R5F526TFDGNE	PWQN0048KC-A	512 Kbytes	64 Kbytes	Available	Available	-40 to 105°C

Note 1. Products with this part number support only CAN 2.0 protocol.

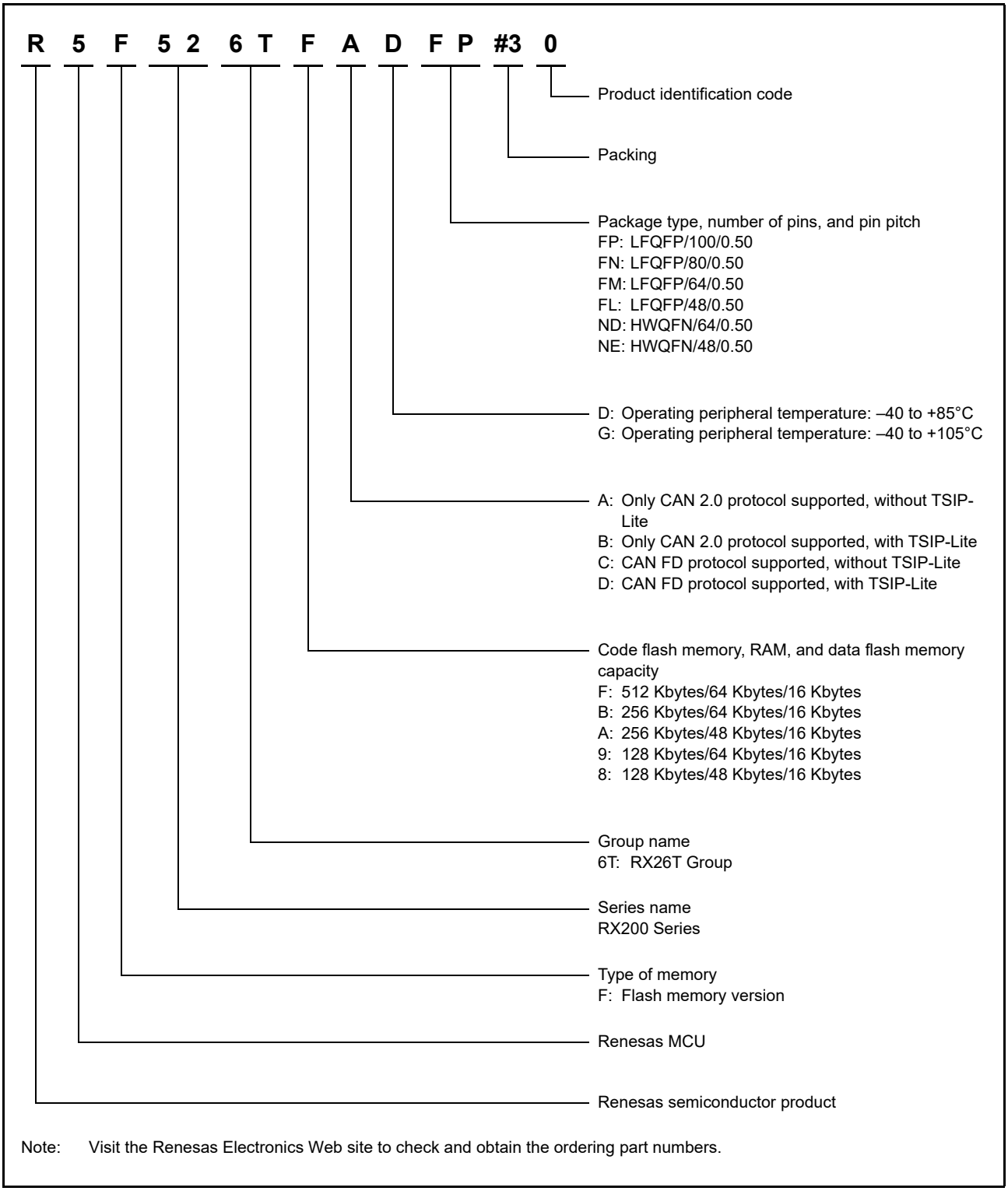


Figure 1.1 How to Read the Product Part Number

1.3 Block Diagram

Figure 1.2 and Figure 1.3 show block diagrams.

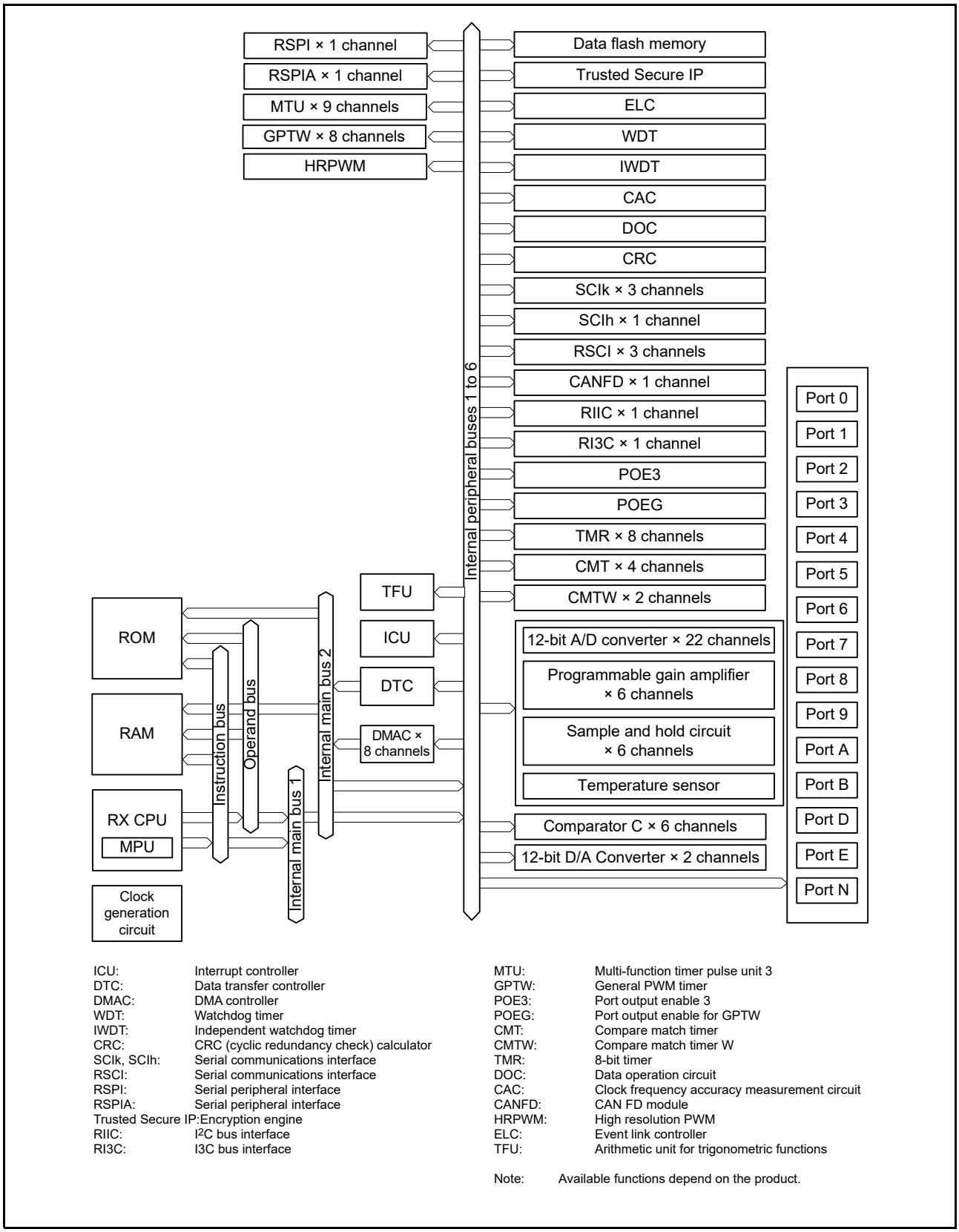


Figure 1.2 Block Diagram (Products with 64 Kbytes of RAM)

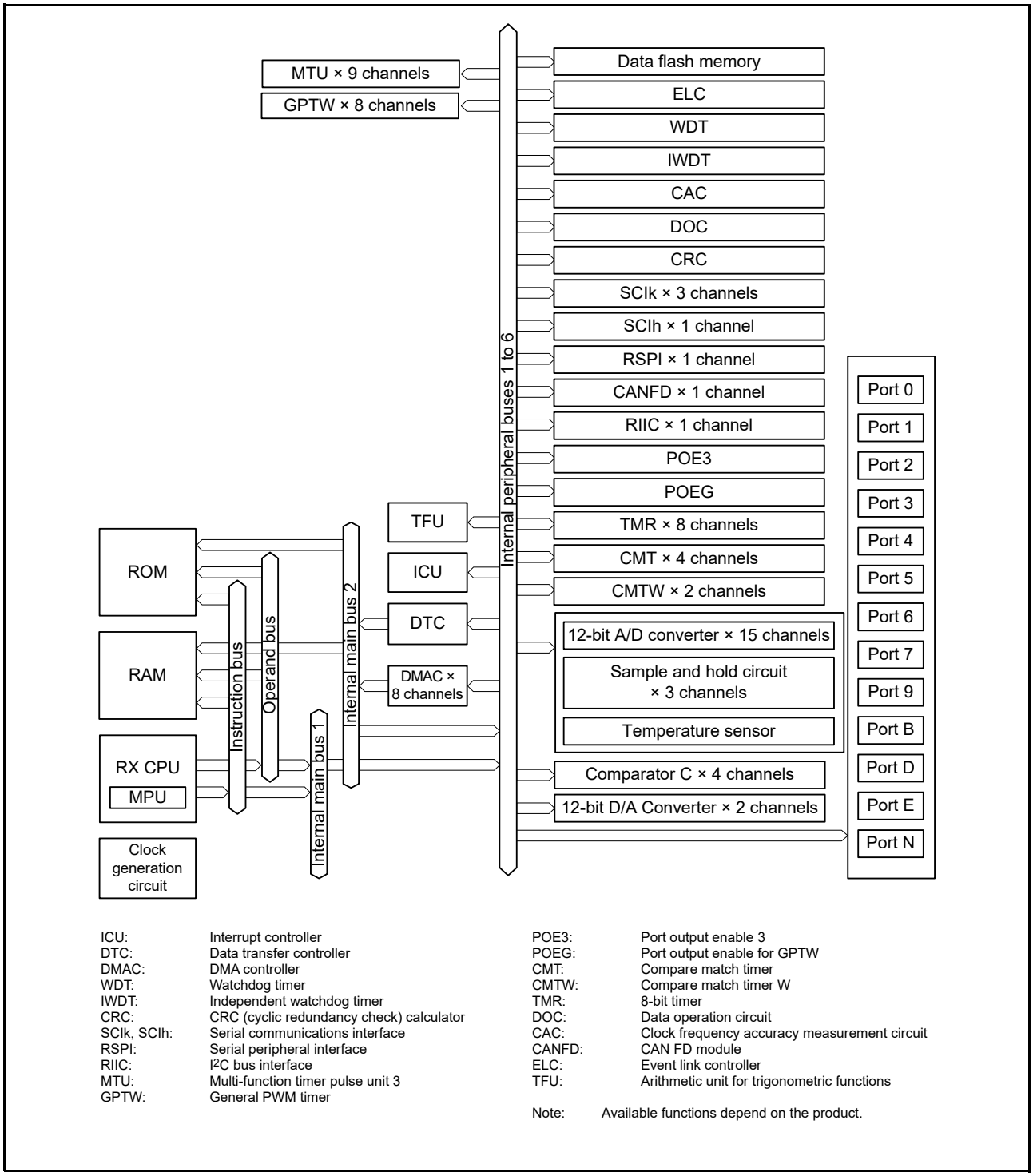


Figure 1.3 Block Diagram (Products with 48 Kbytes of RAM)

1.4 Pin Functions

Table 1.4 lists the pin functions.

Table 1.4 Pin Functions (1/5)

Classifications	Pin Name	I/O	Description
Digital power supply	VCC	Input	Power supply pin. Connect this pin to the system power supply. Connect the pin to VSS via a 0.1-μF multilayer ceramic capacitor. The capacitor should be placed close to the pin.
	VCL	Input	Connect this pin to VSS via a 0.47-μF smoothing capacitor used to stabilize the internal power supply. The capacitor should be placed close to the pin.
	VSS	Input	Ground pin. Connect it to the system power supply (0 V).
Clock	XTAL	Output	Pins for a crystal resonator. An external clock signal can be input through the EXTAL pin.
	EXTAL	Input	
CAC	CACREF	Input	Input pin for the clock frequency accuracy measurement circuit.
Operating mode control	MD	Input	Pins for setting the operating mode. The signal levels on these pins must not be changed during operation.
System control	RES#	Input	Reset pin. This MCU enters the reset state when this signal goes low.
	EMLE	Input	Input pin for the on-chip emulator enable signal. When the on-chip emulator is used, this pin should be driven high. When not used, it should be driven low.
On-chip emulator	FINED	I/O	FINE interface pin.
	TRST#	Input	Pins for the on-chip emulator. When the EMLE pin is driven high, these pins are dedicated for the on-chip emulator.
	TMS	Input	
	TDI	Input	
	TCK	Input	
	TDO	Output	
Interrupt	NMI	Input	Non-maskable interrupt request pin
	IRQ0 to IRQ15	Input	Maskable interrupt request pins
Multi-function timer pulse unit 3	MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D	I/O	The TGRA0 to TGRD0 input capture input/output compare output/PWM output pins
	MTIOC0A#, MTIOC0B#, MTIOC0C#, MTIOC0D#	I/O	The TGRA0 to TGRD0 input capture inverted input/output compare inverted output/PWM inverted output pins.
	MTIOC1A, MTIOC1B	I/O	The TGRA1 and TGRB1 input capture input/output compare output/PWM output pins.
	MTIOC1A#, MTIOC1B#	I/O	The TGRA1 and TGRB1 input capture inverted input/output compare inverted output/PWM inverted output pins.
	MTIOC2A, MTIOC2B	I/O	The TGRA2 and TGRB2 input capture input/output compare output/PWM output pins.
	MTIOC2A#, MTIOC2B#	I/O	The TGRA2 and TGRB2 input capture inverted input/output compare inverted output/PWM inverted output pins.
	MTIOC3A, MTIOC3B, MTIOC3C, MTIOC3D	I/O	The TGRA3 to TGRD3 input capture input/output compare output/PWM output pins.
	MTIOC3A#, MTIOC3B#, MTIOC3C#, MTIOC3D#	I/O	The TGRA3 to TGRD3 input capture inverted input/output compare inverted output/PWM inverted output pins.
	MTIOC4A, MTIOC4B, MTIOC4C, MTIOC4D	I/O	The TGRA4 to TGRD4 input capture input/output compare output/PWM output pins
	MTIOC4A#, MTIOC4B#, MTIOC4C#, MTIOC4D#	I/O	The TGRA4 to TGRD4 input capture inverted input/output compare inverted output/PWM inverted output pins.

Table 1.4 Pin Functions (2/5)

Classifications	Pin Name	I/O	Description
Multi-function timer pulse unit 3	MTIC5U, MTIC5V, MTIC5W	Input	The TGRU5, TGRV5, and TGRW5 input capture input/external pulse input pins
	MTIC5U#, MTIC5V#, MTIC5W#	Input	The TGRU5, TGRV5, and TGRW5 input capture inverted input/ external pulse inverted input pins.
	MTIOC6A, MTIOC6B, MTIOC6C, MTIOC6D	I/O	The TGRA6 to TGRD6 input capture input/output compare output/PWM output pins
	MTIOC6A#, MTIOC6B#, MTIOC6C#, MTIOC6D#	I/O	The TGRA6 to TGRD6 input capture inverted input/output compare inverted output/PWM inverted output pins.
	MTIOC7A, MTIOC7B, MTIOC7C, MTIOC7D	I/O	The TGRA7 to TGRD7 input capture input/output compare output/PWM output pins
	MTIOC7A#, MTIOC7B#, MTIOC7C#, MTIOC7D#	I/O	The TGRA7 to TGRD7 input capture inverted input/output compare inverted output/PWM inverted output pins.
	MTIOC9A, MTIOC9B, MTIOC9C, MTIOC9D	I/O	The TGRA9 to TGRD9 input capture input/output compare output/PWM output pins
	MTIOC9A#, MTIOC9B#, MTIOC9C#, MTIOC9D#	I/O	The TGRA9 to TGRD9 input capture inverted input/output compare inverted output/PWM inverted output pins.
	MTCLKA, MTCLKB, MTCLKC, MTCLKD	Input	Input pins for the external clock.
	MTCLKA#, MTCLKB#, MTCLKC#, MTCLKD#	Input	Inverted input pins for the external clock.
	ADSM0, ADSM1	Output	A/D conversion start request frame synchronization signal output pins.
General PWM timer	GTETRGA, GTETRGB, GTETRGC, GTETRGD	Input	External trigger input pins
	GTIOC0A to GTIOC7A, GTIOC0B to GTIOC7B	I/O	Input capture input/output compare output/PWM output pins
	GTIOC0A# to GTIOC7A#, GTIOC0B# to GTIOC7B#	I/O	Input capture inverted input/output compare inverted output/ PWM inverted output pins
	GTCPPO0, GTCPPO4	Output	Synchronized PWM output
	GTIU, GTIV, GTIW	Input	Hall sensor input pins
	GTOUUP	Output	A three-phase PWM output for controlling a brushless DC motor (positive U-phase)
	GTOULO	Output	A three-phase PWM output for controlling a brushless DC motor (negative U-phase)
	GTOVUP	Output	A three-phase PWM output for controlling a brushless DC motor (positive V-phase)
	GTOVLO	Output	A three-phase PWM output for controlling a brushless DC motor (negative V-phase)
	GTOWUP	Output	A three-phase PWM output for controlling a brushless DC motor (positive W-phase)
	GTOWLO	Output	A three-phase PWM output for controlling a brushless DC motor (negative W-phase)
	GTADSM0, GTADSM1	Output	A/D conversion start request monitoring output pins
8-bit timer	TMO0 to TMO7	Output	Compare match output pins.
	TMCI0 to TMCI7	Input	Input pins for the external clock to be input to the counter.
	TMRI0 to TMRI7	Input	Counter reset input pins.
Compare match timer W	TIC0 to TIC3	Input	Input pins for CMTW
	TOC0 to TOC3	Output	Output pins for CMTW
Port output enable 3	POE0#, POE4#, POE8#, POE9#, POE10#, POE11#, POE12#	Input	Input pins for request signals to switch the MTU and GPTW pins between the high impedance state

Table 1.4 Pin Functions (3/5)

Classifications	Pin Name	I/O	Description
Serial communications interface (SCIk)	• Asynchronous mode/clock synchronous mode		
	SCK1, SCK5, SCK6	I/O	Input/output pins for the clock
	RXD1, RXD5, RXD6	Input	Input pins for received data
	TXD1, TXD5, TXD6	Output	Output pins for transmitted data
	CTS1#, CTS5#, CTS6#	Input	Input pins for controlling the start of transmission and reception.
	RTS1#, RTS5#, RTS6#	Output	Output pins for controlling the start of transmission and reception.
	• Simple I^2C mode		
	SSCL1, SSCL5, SSCL6	I/O	Input/output pins for the I^2C clock.
	SSDA1, SSDA5, SSDA6	I/O	Input/output pins for the I^2C data.
	• Simple SPI mode		
	SCK1, SCK5, SCK6	I/O	Input/output pins for the clock
	SMISO1, SMISO5, SMISO6	I/O	Input/output pins for slave transmit data.
	SMOSI1, SMOSI5, SMOSI6	I/O	Input/output pins for master transmit data.
	SS1#, SS5#, SS6#	Input	Chip-select input pins.
Serial communications interface (SCIh)	• Asynchronous mode/clock synchronous mode		
	SCK12	I/O	Input/output pin for the clock
	RXD12	Input	Input pin for received data
	TXD12	Output	Output pin for transmitted data
	CTS12#	Input	Input pin for controlling the start of transmission and reception
	RTS12#	Output	Output pin for controlling the start of transmission and reception
	• Simple I^2C mode		
	SSCL12	I/O	Input/output pin for the I^2C clock
	SSDA12	I/O	Input/output pin for the I^2C data
	• Simple SPI mode		
	SCK12	I/O	Input/output pin for the clock
	SMISO12	I/O	Input/output pin for slave transmission of data
	SMOSI12	I/O	Input/output pin for master transmission of data
	SS12#	Input	Chip-select input pin
	• Extended serial mode		
	RXDX12	Input	Input pin for received data
	TXDX12	Output	Output pin for transmitted data
	SIOX12	I/O	Input/output pin for received or transmitted data
Serial communications interface (RSCI)	• Asynchronous mode/clock synchronous mode		
	SCK008, SCK009, SCK011	I/O	Input/output pins for the clock
	RXD008, RXD009, RXD011	Input	Input pins for received data
	TXD008, TXD009, TXD011	Output	Output pins for transmitted data
	CTS008#, CTS009#, CTS011#	Input	Input pins for controlling the start of transmission and reception
	RTS008#, RTS009#, RTS011#	Output	Output pins for controlling the start of transmission and reception
	DE008, DE009, DE011	Output	DriveEnable output pins
	• Simple I^2C mode		
	SSCL008, SSCL009, SSCL011	I/O	Input/output pins for the I^2C clock
	SSDA008, SSDA009, SSDA011	I/O	Input/output pins for the I^2C data

Table 1.4 Pin Functions (4/5)

Classifications	Pin Name	I/O	Description
Serial communications interface (RSCI)	• Simple SPI mode		
	SCK008, SCK009, SCK011	I/O	Input/output pins for the clock
	SMISO008, SMISO009, SMISO011	I/O	Input/output pins for slave transmission of data
	SMOSI008, SMOSI009, SMOSI011	I/O	Input/output pins for master transmission of data
	SS008#, SS009#, SS011#	Input	Chip-select input pins
	• HBS support mode		
	RXD008, RXD009, RXD011	Input	Input pins for received data
	TXDA008, TXDA009, TXDA011	Output	Output pins for transmitted data
	TXDB008, TXDB009, TXDB011		
I^2C bus interface	SCL0	I/O	Input/output pin for I^2C bus interface clocks. Bus can be directly driven by the N-channel open drain output.
	SDA0	I/O	Input/output pin for I^2C bus interface data. Bus can be directly driven by the N-channel open drain output.
I3C bus interface	SCL00	I/O	Input/output pin for I3C bus interface clocks.
	SDA00	I/O	Input/output pin for I3C bus interface data.
CAN FD module	CRX0	Input	Input pins
	CTX0	Output	Output pins
Serial peripheral interface	RSPCKA	I/O	Input/output pin for the RSPI clock.
	MOSIA	I/O	Input/output pin for transmitting data from the RSPI master.
	MISOA	I/O	Input/output pin for transmitting data from the RSPI slave.
	SSLA0	I/O	Input/output pin to select the slave for the RSPI.
	SSLA1 to SSLA3	Output	Output pins to select the slave for the RSPI.
Serial peripheral interface (RSPIA)	RSPCK0	I/O	Input/output pin for the RSPIA clock.
	MOSI0	I/O	Input/output pin for transmitting data from the RSPIA master.
	MISO0	I/O	Input/output pin for transmitting data from the RSPIA slave.
	SSL00	I/O	Input/output pin to select the slave for the RSPIA.
	SSL01 to SSL03	Output	Output pins to select the slave for the RSPIA.
12-bit A/D converter	AN000 to AN006, AN100 to AN103, AN200 to AN211, AN216, AN217	Input	Input pins for the analog signals to be processed by the A/D converter.
	ADST0, ADST1, ADST2	Output	Output pins for A/D conversion status.
	ADTRG0#, ADTRG1#, ADTRG2#	Input	Input pins for the external trigger signals that start the A/D conversion.
12-bit D/A converter	DA0, DA1	Output	Output pins for the analog signals to be processed by the D/A converter
Comparator C	COMP0 to COMP5	Output	Comparator detection result output pins.
	CVREFC0, CVREFC1	Input	Analog reference voltage supply pins for comparator C.
	CMPCnm	Input	Analog input pin for CMPCnm (n = 0 to 5, m = 0 to 3)

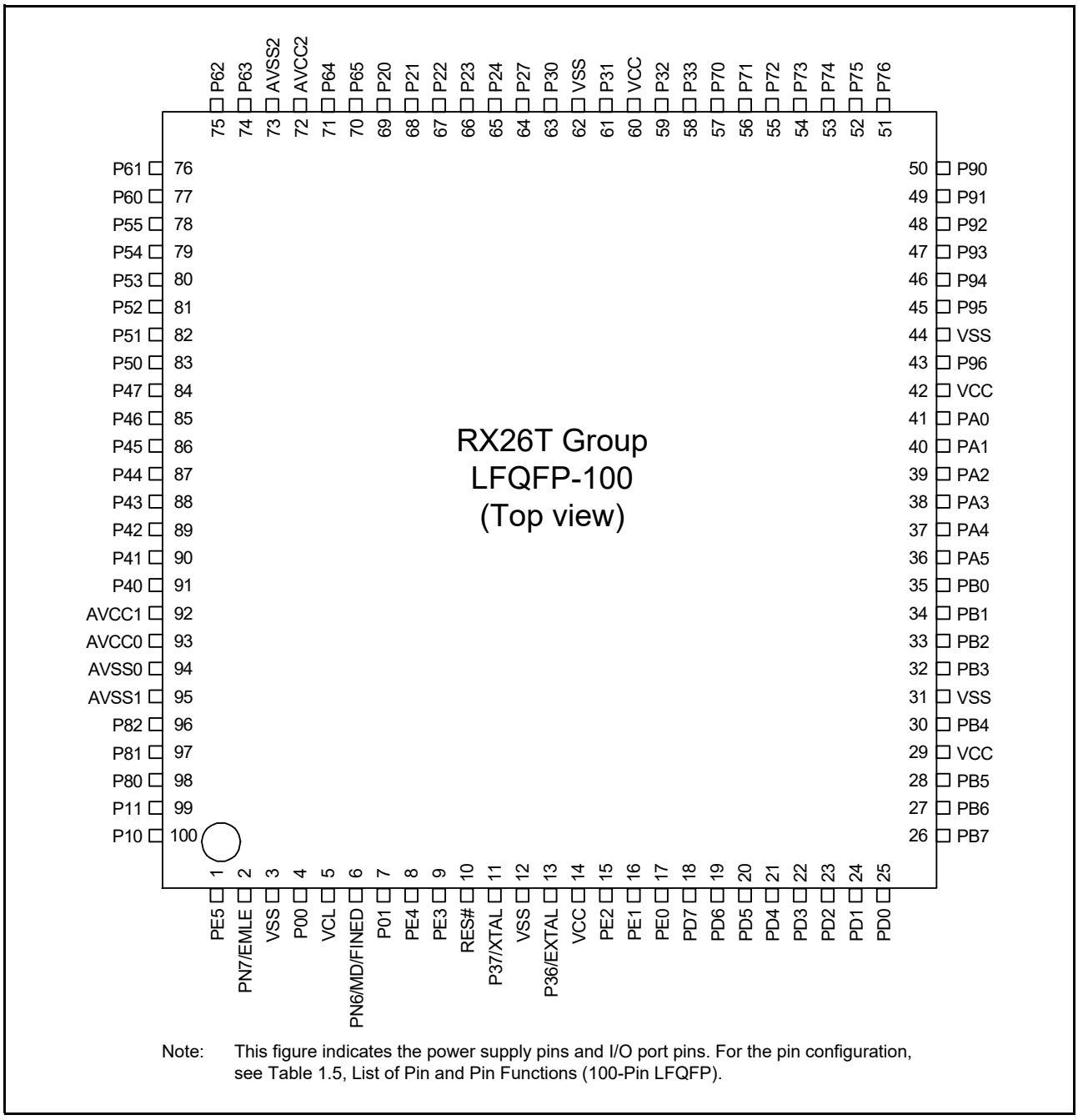
Table 1.4 Pin Functions (5/5)

Classifications	Pin Name	I/O	Description
Analog power supply	AVCC0	Input	Analog voltage supply pin for 12-bit A/D converter unit 0. Connect the AVCC0 pin to AVCC1 or AVCC2 when 12-bit A/D converter unit 0 is not used.
	AVSS0	Input	Analog ground pin for 12-bit A/D converter unit 0. Connect the AVSS0 pin to AVSS1 or AVSS2 when 12-bit A/D converter unit 0 is not used.
	AVCC1	Input	Analog voltage supply pin for 12-bit A/D converter unit 1. Connect this pin to AVCC0 when not using the 12-bit A/D converter 1 but using the 12-bit A/D converter 0. Connect this pin to AVCC2 when not using the 12-bit A/D converter 0 and the 12-bit A/D converter 1.
	AVSS1	Input	Analog ground pin for 12-bit A/D converter unit 1. Connect this pin to AVSS0 when not using the 12-bit A/D converter 1 but using the 12-bit A/D converter 0. Connect this pin to AVSS2 when not using the 12-bit A/D converter 0 and the 12-bit A/D converter 1.
	AVCC2	Input	Analog voltage supply pin for the 12-bit A/D converter unit 2, reference voltage supply pin for the 12-bit D/A converter, analog voltage supply pin for the comparator C, and analog voltage supply pin for the temperature sensor. Connect this pin to either of AVCC0 or AVCC1 when not using the 12-bit A/D converter unit 2, 12-bit D/A converter, comparator C, and temperature sensor.
	AVSS2	Input	Analog ground pin for the 12-bit A/D converter unit 2, reference ground pin for the D/A converter, analog ground pin for the comparator C, and analog ground pin for the temperature sensor. Connect this pin to either of AVSS0 or AVSS1 when not using the 12-bit A/D converter unit 2, 12-bit D/A converter, comparator C, and temperature sensor.
I/O ports	P00, P01	I/O	General-purpose input/output pins
	P10, P11	I/O	General-purpose input/output pins
	P20 to P24, P27	I/O	General-purpose input/output pins
	P30 to P33, P36, P37	I/O	General-purpose input/output pins
	P40 to P47	I/O	General-purpose input/output pins
	P50 to P55	I/O	General-purpose input/output pins
	P60 to P65	I/O	General-purpose input/output pins
	P70 to P76	I/O	General-purpose input/output pins
	P80 to P82	I/O	General-purpose input/output pins
	P90 to P96	I/O	General-purpose input/output pins
	PA0 to PA5	I/O	General-purpose input/output pins
	PB0 to PB7	I/O	General-purpose input/output pins
	PD0 to PD7	I/O	General-purpose input/output pins
	PE0 to PE5	I/O	General-purpose input/output pins (PE2: input pin)
	PN6*1, PN7*2	I/O	General-purpose input/output pins

Note: When not using any of the A/D converter, D/A converter, comparator C and temperature sensor, connect the AVCC0, AVCC1 and AVCC2 pins to VCC, and connect the AVSS0, AVSS1 and AVSS2 pins to VSS, respectively.
Note 1. This pin functions as MD after release from the reset state, and the pull-up resistor connected to the MD pin is enabled.
Note 2. This pin functions as EMLE after release from the reset state, and the pull-down resistor connected to the EMLE pin is enabled.

1.5 Pin Assignments

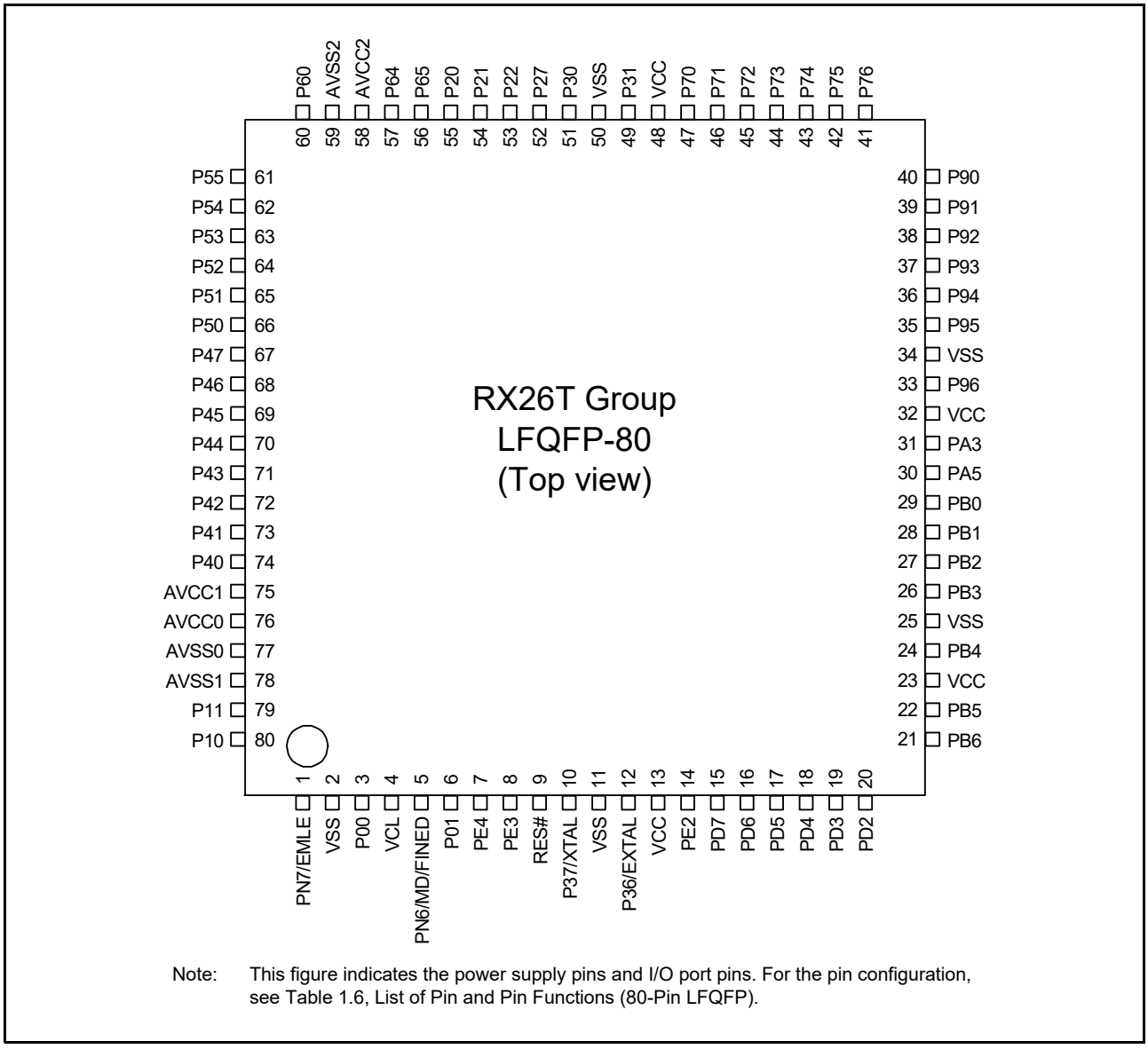
1.5.1 100-Pin LFQFP



Note: This figure indicates the power supply pins and I/O port pins. For the pin configuration, see Table 1.5, List of Pin and Pin Functions (100-Pin LFQFP).

Figure 1.4 Pin Assignment (100-pin LFQFP)

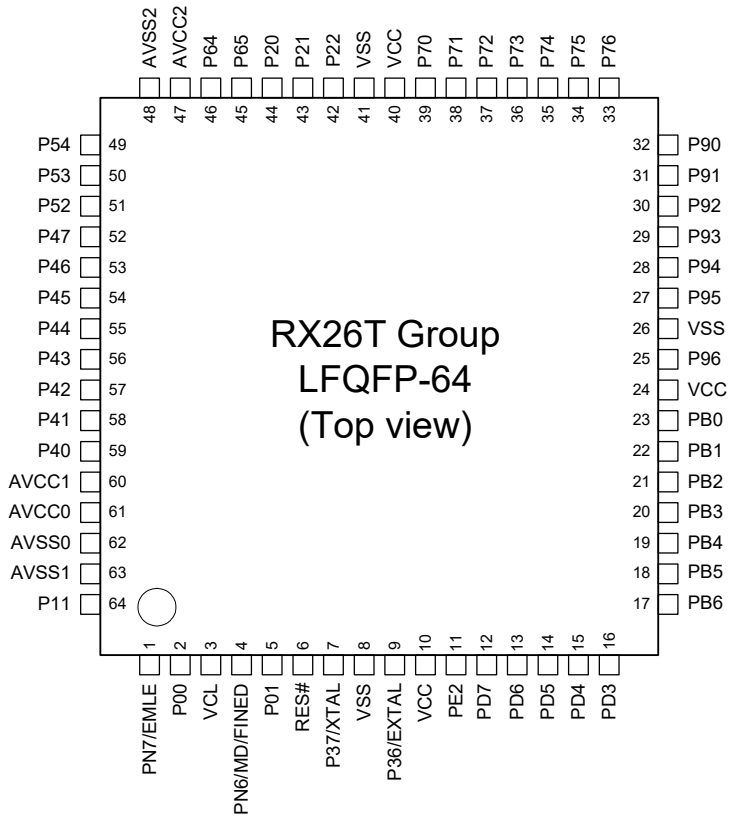
1.5.2 80-Pin LFQFP



Note: This figure indicates the power supply pins and I/O port pins. For the pin configuration, see Table 1.6, List of Pin and Pin Functions (80-Pin LFQFP).

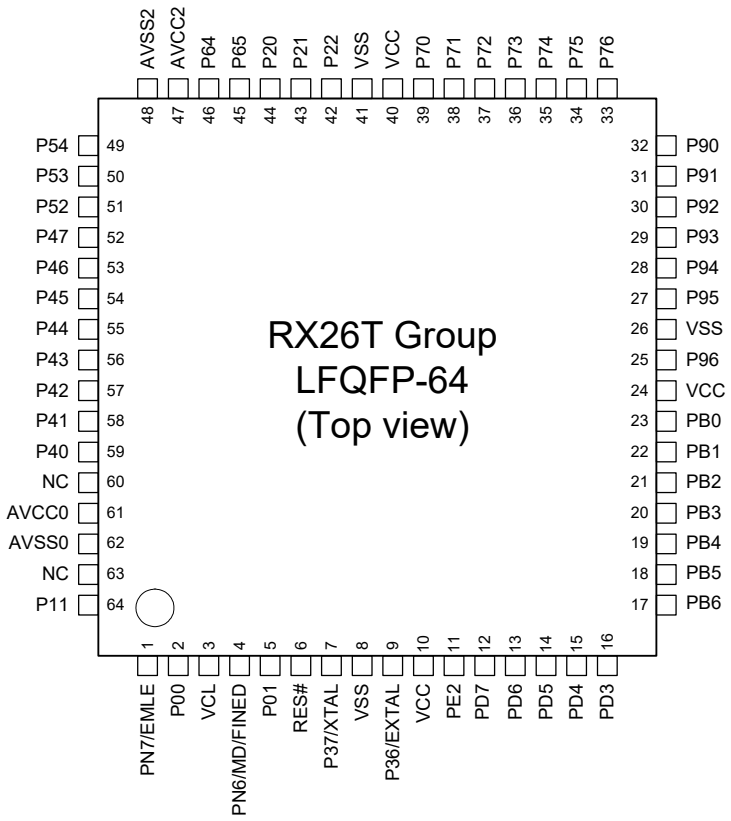
Figure 1.5 Pin Assignment (80-pin LFQFP)

1.5.3 64-Pin LFQFP and 64-Pin HWQFN



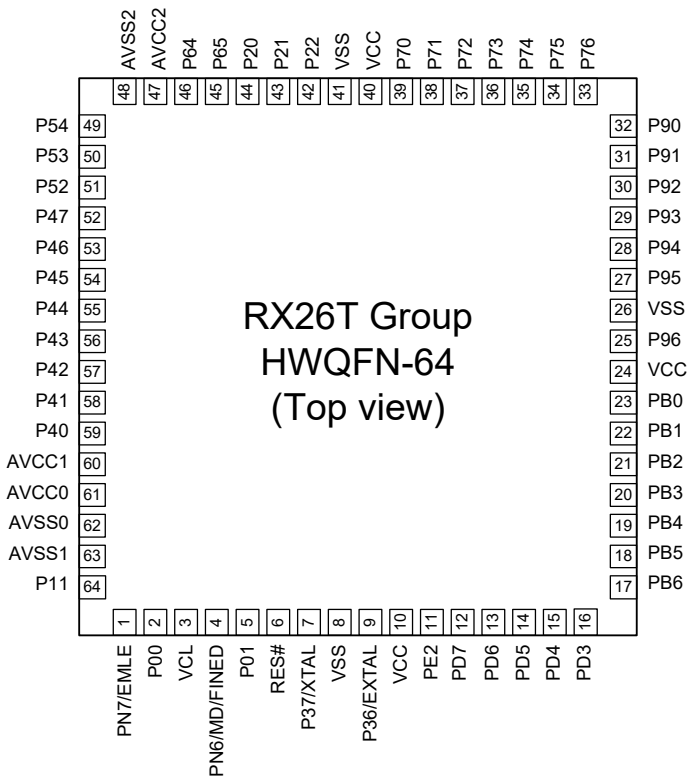
Note: This figure indicates the power supply pins and I/O port pins. For the pin configuration, see Table 1.7, List of Pin and Pin Functions (64-Pin LFQFP, 64-Pin HWQFN) (Products with 64 Kbytes of RAM).

Figure 1.6 Pin Assignment (64-pin LFQFP) (Products with 64 Kbytes of RAM)



Note: This figure indicates the power supply pins and I/O port pins. For the pin configuration, see Table 1.8, List of Pin and Pin Functions (64-Pin LFQFP) (Products with 48 Kbytes of RAM).

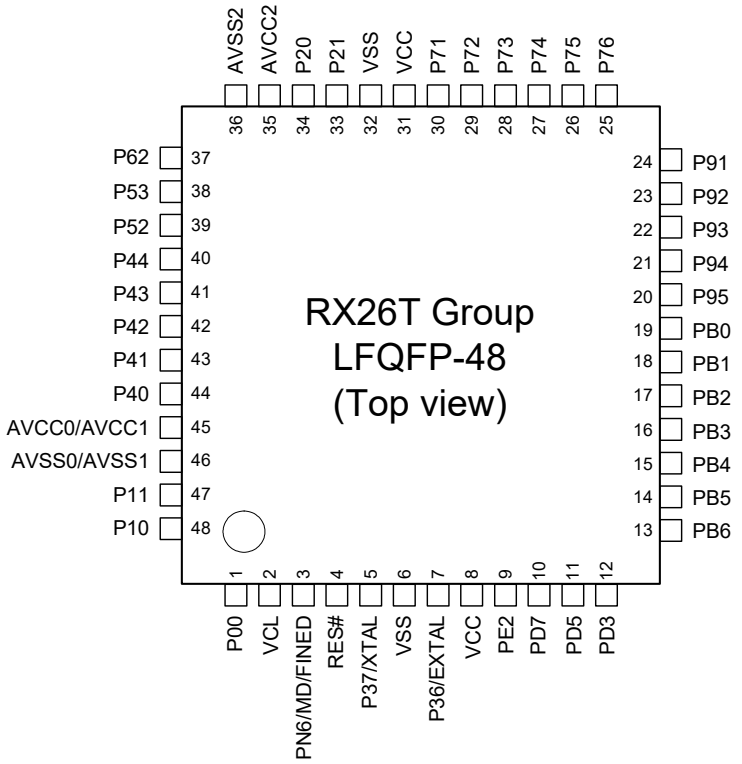
Figure 1.7 Pin Assignment (64-pin LFQFP) (Products with 48 Kbytes of RAM)



Note: We recommend connecting the exposed die pad to VSS.
Note: This figure indicates the power supply pins and I/O port pins. For the pin configuration, see Table 1.7, List of Pin and Pin Functions (64-Pin LFQFP, 64-Pin HWQFN) (Products with 64 Kbytes of RAM).

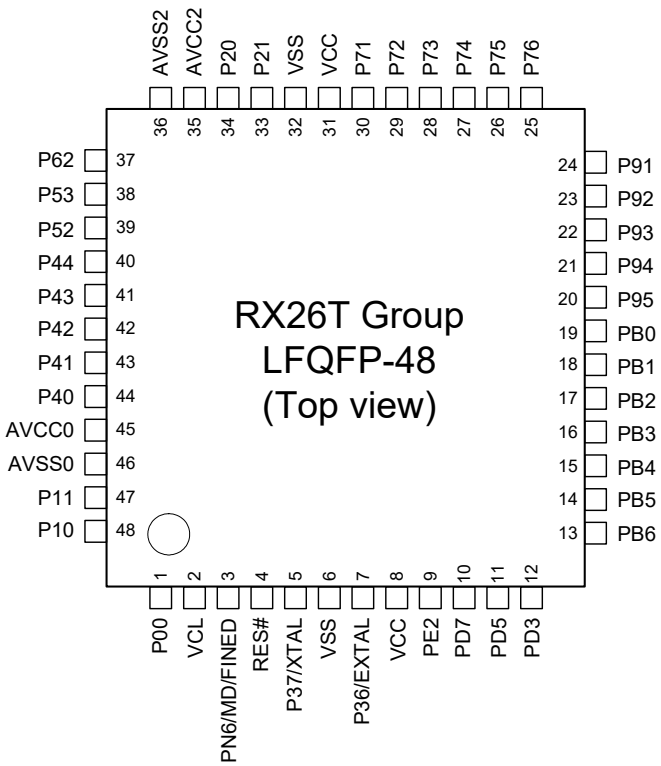
Figure 1.8 Pin Assignment (64-pin HWQFN)

1.5.4 48-Pin LFQFP and 48-Pin HWQFN



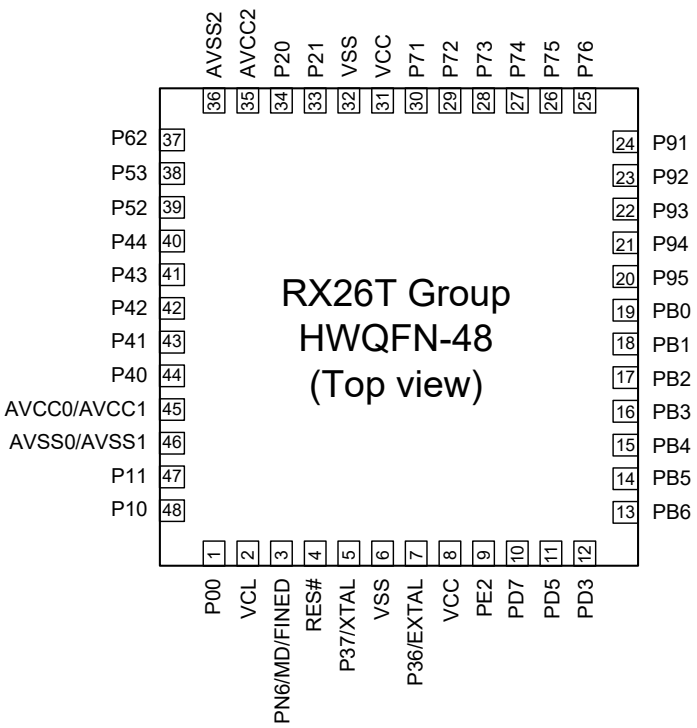
Note: This figure indicates the power supply pins and I/O port pins. For the pin configuration, see Table 1.9, List of Pin and Pin Functions (48-Pin LFQFP, 48-Pin HWQFN) (Products with 64 Kbytes of RAM).

Figure 1.9 Pin Assignment (48-pin LFQFP) (Products with 64 Kbytes of RAM)



Note: This figure indicates the power supply pins and I/O port pins. For the pin configuration, see Table 1.10, List of Pin and Pin Functions (48-Pin LFQFP) (Products with 48 Kbytes of RAM).

Figure 1.10 Pin Assignment (48-pin LFQFP) (Products with 48 Kbytes of RAM)



Note: We recommend connecting the exposed die pad to VSS.
Note: This figure indicates the power supply pins and I/O port pins. For the pin configuration, see Table 1.9, List of Pin and Pin Functions (48-Pin LFQFP, 48-Pin HWQFN) (Products with 64 Kbytes of RAM).

Figure 1.11 Pin Assignment (48-pin HWQFN)

1.6 List of Pin and Pin Functions

1.6.1 100-Pin LFQFP

Table 1.5 List of Pin and Pin Functions (100-Pin LFQFP) (1/6)

Pin Number 100-Pin LFQFP	Power Supply Clock System Control	I/O Port	Timer (MTU, GPTW, TMR, POE, POEG, CAC, CMTW)	Communications (SCI, RSCI, RSPI, RSPIA, RIIC, RI3C, CANFD)	Interrupt (IRQ, NMI)	Analog (A/D, D/A, CMPC)
1		PE5	MTIOC9D/MTIOC9D#/ GTIOC3A/GTETRGB/ GTIOC3A#/GTETRGD	SCK009/CTS009#/ RTS009#/SS009#/TXDB009	IRQ0	ADST0
2	EMLE	PN7	MTIOC9D/MTIOC9D#		IRQ5	ADST0
3	VSS					
4		P00	MTIOC9A/MTIOC9A#/ CACREF/GTIU/TIC3	RXD12/SMISO12/SSCL12/ RXDX12/RXD009/ SMISO009/SSCL009	IRQ2	ADST1/ COMP0
5	VCL					
6	MD/FINED	PN6				
7		P01	MTIOC9C/MTIOC9C#/ POE12#/GTETRGA/ GTETRGB/GTETRGC/ GTETRGD/GTIW	TXD12/SMOSI12/SSDA12/ TXDX12/SIOX12/TXD009/ TXDA009/SMOSI009/ SSDA009	IRQ4	ADST2/ COMP1
8		PE4	MTCLKC/MTCLKC#/ POE10#/GTETRGA/ GTETRGB/GTETRGC/ GTETRGD	SCK009/TXDB009	IRQ1	
9		PE3	MTCLKD/MTCLKD#/ POE11#/GTETRGA/ GTETRGB/GTETRGC/ GTETRGD	CTS009#/RTS009#/SS009#/ DE009	IRQ2	
10	RES#					
11	XTAL	P37		RXD5/SMISO5/SSCL5		
12	VSS					
13	EXTAL	P36		TXD5/SMOSI5/SSDA5		
14	VCC					
15		PE2	POE10#		NMI/IRQ0	
16		PE1	MTIOC9D/MTIOC9D#/TMO5	CTS5#/RTS5#/SS5#/ CTS12#/RTS12#/SS12#/ SSLA3/SSL03	IRQ15	
17		PE0	MTIOC9B/MTIOC9B#/ TMCI1/TMCI5/GTIV	RXD5/SMISO5/SSCL5/ SSLA2/SSL02/CRX0	IRQ7	
18	TRST#	PD7	MTIOC9A/MTIOC9A#/ TMRI1/TMRI5/GTIOC0A/ GTIOC3A/GTIOC0A#/ GTIOC3A#/GTIU	TXD5/SMOSI5/SSDA5/ SCK009/TXD008/TXDA008/ SMOSI008/SSDA008/ TXDB009/SSLA1/SSL01/ CTX0	IRQ8	
19	TMS	PD6	MTIOC9C/MTIOC9C#/ TMO1/GTIOC0B/GTIOC3B/ GTIOC0B#/GTIOC3B#/ GTIW	CTS1#/RTS1#/SS1#/ RXD12/SMISO12/SSCL12/ RXDX12/CTS011#/ RTS011#/SS011#/DE011/ SSLA0/SSL00	IRQ5	ADST0
20	TDI	PD5	TMRI0/TMRI6/GTIOC1A/ GTETRGA/GTIOC1A#/ GTIOC7A	RXD1/SMISO1/SSCL1/ RXD011/SMISO011/ SSCL011/SSL00	IRQ6	

Table 1.5 List of Pin and Pin Functions (100-Pin LFQFP) (2/6)

Pin Number 100-Pin LFQFP	Power Supply Clock System Control	I/O Port	Timer (MTU, GPTW, TMR, POE, POEG, CAC, CMTW)	Communications (SCI, RSCI, RSPI, RSPIA, RIIC, RI3C, CANFD)	Interrupt (IRQ, NMI)	Analog (A/D, D/A, CMPC)
21	TCK	PD4	TMCI0/TMCI6/GTIOC1B/ GTETRGB/GTIOC1B#	SCK1/TXD12/SMOSI12/ SSDA12/TXDX12/SIOX12/ SCK011/TXDB011/SSL02	IRQ2	
22	TDO	PD3	TMO0/GTIOC2A/GTETRGC/ GTIOC2A#/GTIOC7B	TXD1/SMOSI1/SSDA1/ TXD011/TXDA011/ SMOSI011/SSDA011/MOSI0		
23		PD2	TMCI1/TMO4/GTIOC2B/ GTIOC0A/GTIOC2B#/ GTIOC0A#	SCK5/SCK008/TXDB008/ MOSIA/MOSI0		
24		PD1	TMO2/GTIOC3A/GTIOC0B/ GTIOC3A#/GTIOC0B#	RXD008/SMISO008/ SSCL008/MISOA/MISO0		
25		PD0	TMO6/GTIOC3B/GTIOC1A/ GTIOC3B#/GTIOC1A#	TXD008/TXDA008/ SMOSI008/SSDA008/ RSPCKA/RSPCK0		
26		PB7	GTIOC1B/GTIOC1B#	SCK5/SCK12/SCK011/ TXDB011/SSL03		
27		PB6	GTIOC2A/GTIOC3A/ GTIOC2A#/GTIOC3A#/ TOC0	RXD5/SMISO5/SSCL5/ RXD12/SMISO12/SSCL12/ RXDX12/RXD011/ SMISO011/SSCL011/MISO0/ CRX0	IRQ2	
28		PB5	GTIOC2B/GTIOC3B/ GTIOC2B#/GTIOC3B#/TIC0	TXD5/SMOSI5/SSDA5/ TXD12/SMOSI12/SSDA12/ TXDX12/SIOX12/TXD011/ TXDA011/SMOSI011/ SSDA011/RSPCK0/CTX0		
29	VCC					
30		PB4	POE8#/GTETRGA/ GTETRGB/GTETRGC/ GTETRGD/GTCPPO0	CTS5#/RTS5#/SS5#/ RXD12/SMISO12/SSCL12/ RXDX12/CTS011#/ RTS011#/SS011#/SCK011/ TXDB011/MISOA/SSL01/ CRX0	IRQ3	
31	VSS					
32		PB3	MTIOC0A/MTIOC0A#/ CACREF/GTIU/TOC1	SCK6/TXD12/SMOSI12/ SSDA12/TXDX12/SIOX12/ CTS009#/RTS009#/SS009#/ DE009/RSPCKA/CTX0	IRQ9	
33		PB2	MTIOC0B/MTIOC0B#/ TMRI0/GTADSM0/ GTIOC7A/GTIOC7A#/GTIV/ TIC1	TXD6/SMOSI6/SSDA6/ SDA0/SDA00		ADSM0
34		PB1	MTIOC0C/MTIOC0C#/ TMCI0/GTADSM1/ GTIOC7B/GTIOC7B#/GTIW/ TOC2	RXD6/SMISO6/SSCL6/ SCL0/SCL00	IRQ4	ADSM1
35		PB0	MTIOC0D/MTIOC0D#/ TMO0/TIC2	TXD6/SMOSI6/SSDA6/ TXD008/TXDA008/ SMOSI008/SSDA008/ CTS011#/RTS011#/SS011#/ DE011/MOSIA/MOSI0	IRQ8	ADTRG2#
36		PA5	MTIOC1A/MTIOC1A#/ TMCI3	RXD6/SMISO6/SSCL6/ RXD008/SMISO008/ SSCL008/MISOA/MISO0	IRQ1	ADTRG1#

Table 1.5 List of Pin and Pin Functions (100-Pin LFQFP) (3/6)

Pin Number 100-Pin LFQFP	Power Supply Clock System Control	I/O Port	Timer (MTU, GPTW, TMR, POE, POEG, CAC, CMTW)	Communications (SCI, RSCI, RSPI, RSPIA, RIIC, RI3C, CANFD)	Interrupt (IRQ, NMI)	Analog (A/D, D/A, CMPC)
37		PA4	MTIOC1B/MTIOC1B#/ TMCI7	SCK6/TXD008/TXDA008/ SMOSI008/SSDA008/ RSPCKA/RSPCK0		ADTRG0#
38		PA3	MTIOC2A/MTIOC2A#/ TMRI7/GTADSM0	TXD009/TXDA009/ SMOSI009/SSDA009/ SCK008/TXDB008/SSLA0/ SSL00		
39		PA2	MTIOC2B/MTIOC2B#/ TMO7/GTADSM1	CTS6#/RTS6#/SS6#/ RXD009/SMISO009/ SSCL009/SSLA1/SSL01		
40		PA1	MTIOC6A/MTIOC6A#/ TMO4/GTCPPO4	TXD009/TXDA009/ SMOSI009/SSDA009/ RXD011/SMISO011/ SSCL011/SSLA2/SSL02/ CRX0	IRQ14	ADTRG0#
41		PA0	MTIOC6C/MTIOC6C#/TMO2	SCK009/TXD011/TXDA011/ SMOSI011/SSDA011/ TXDB009/SSLA3/SSL03/ CTX0		
42	VCC					
43		P96	POE4#/GTETRGA/ GTETRGB/GTETRGC/ GTETRGD/GTCPPO4	CTS008#/RTS008#/SS008#/ DE008/SSL03/RSPCK0	IRQ4	
44	VSS					
45		P95	MTIOC6B/MTIOC1A/ MTIOC6B#/MTIOC1A#/ TMCI3/GTIOC4A/GTIOC7A/ GTIOC4A#/GTIOC7A#/ GTOUUP	RXD6/SMISO6/SSCL6/ RXD008/SMISO008/ SSCL008/MISOA/SSL02/ MISO0	IRQ1	ADTRG1#
46		P94	MTIOC7A/MTIOC2A/ MTIOC7A#/MTIOC2A#/ TMRI7/GTIOC5A/ GTADSM0/GTIOC5A#/ GTOVUP	TXD009/TXDA009/ SMOSI009/SSDA009/ SCK008/TXDB008/SSLA0/ SSL00		
47		P93	MTIOC7B/MTIOC6A/ MTIOC7B#/MTIOC6A#/ TMO4/GTIOC6A/GTIOC6A#/ GTOWUP	TXD009/TXDA009/ SMOSI009/SSDA009/ RXD011/SMISO011/ SSCL011/SSLA2/SSL02/ MOSI0/CRX0	IRQ14	ADTRG0#
48		P92	MTIOC6D/MTIOC6C/ MTIOC6D#/MTIOC6C#/ TMO2/GTIOC4B/GTIOC7B/ GTIOC4B#/GTIOC7B#/ GTOULO	SCK009/TXD011/TXDA011/ SMOSI011/SSDA011/ TXDB009/SSLA3/SSL03/ MISO0/CTX0		
49		P91	MTIOC7C/MTIOC7C#/ GTIOC5B/GTIOC5B#/ GTOVLO	RXD5/SMISO5/SSCL5/ RSPCK0		
50		P90	MTIOC7D/MTIOC7D#/ GTIOC6B/GTIOC6B#/ GTOWLO	TXD5/SMOSI5/SSDA5/ SSL01		
51		P76	MTIOC4D/MTIOC4D#/ GTIOC2B/GTIOC6B/ GTIOC2B#/GTIOC6B#/ GTOWLO	SSL03		

Table 1.5 List of Pin and Pin Functions (100-Pin LFQFP) (4/6)

Pin Number 100-Pin LFQFP	Power Supply Clock System Control	I/O Port	Timer (MTU, GPTW, TMR, POE, POEG, CAC, CMTW)	Communications (SCI, RSCI, RSPI, RSPIA, RIIC, RI3C, CANFD)	Interrupt (IRQ, NMI)	Analog (A/D, D/A, CMPC)
52		P75	MTIOC4C/MTIOC4C#/ GTIOC1B/GTIOC5B/ GTIOC1B#/GTIOC5B#/ GTOVLO	SSL02		
53		P74	MTIOC3D/MTIOC3D#/ GTIOC0B/GTIOC4B/ GTIOC0B#/GTIOC4B#/ GTOULO	SSL01		
54		P73	MTIOC4B/MTIOC4B#/ GTIOC2A/GTIOC6A/ GTIOC2A#/GTIOC6A#/ GTOWUP	SSL00		
55		P72	MTIOC4A/MTIOC4A#/ GTIOC1A/GTIOC5A/ GTIOC1A#/GTIOC5A#/ GTOVUP	MOSI0		
56		P71	MTIOC3B/MTIOC3B#/ GTIOC0A/GTIOC4A/ GTIOC0A#/GTIOC4A#/ GTOUUP	MISO0		
57		P70	MTIOC0A/MTCLKC/ MTIOC0A#/MTCLKC#/ TMRI6/POE0#/GTETRGA/ GTETRGB/GTETRGC/ GTETRGD/GTCPPO0	SCK5/CTS009#/RTS009#/ SS009#/DE009/SSLA0/ RSPCK0	IRQ5	
58		P33	MTIOC3A/MTCLKA/ MTIOC3A#/MTCLKA#/ TMO0/GTIOC3B/GTIOC7B/ GTIOC3B#/GTIOC7B#/ GTCPPO0	SSLA3/SSL03	IRQ13	
59		P32	MTIOC3C/MTCLKB/ MTIOC3C#/MTCLKB#/ TMO6/GTIOC3A/GTIOC7A/ GTIOC3A#/GTIOC7A#	SSLA2/SSL02	IRQ12	
60	VCC					
61		P31	MTIOC0A/MTCLKC/ MTIOC0A#/MTCLKC#/ TMRI6/GTIU	SSLA1/SSL01	IRQ6	
62	VSS					
63		P30	MTIOC0B/MTCLKD/ MTIOC0B#/MTCLKD#/ TMCI6/GTIV	SCK008/CTS008#/ RTS008#/SS008#/DE008/ SSLA0/SSL00	IRQ7	COMP3
64		P27	MTIOC1A/MTIOC0C/ MTIOC1A#/MTIOC0C#/ TMO2/TMO6/POE9#	RSPCKA/RSPCK0	IRQ15	
65		P24	MTIC5U/MTIC5U#/TMCI2/ TMO6	CTS008#/RTS008#/SS008#/ SCK008/DE008/RSPCKA/ RSPCK0	IRQ4	COMP0
66		P23	MTIC5V/MTIC5V#/TMO2/ CACREF	TXD12/SMOSI12/SSDA12/ TXDX12/SIOX12/TXD008/ TXDA008/SMOSI008/ SSDA008/MOSIA/MOSI0/ CTX0	IRQ11	COMP1

Table 1.5 List of Pin and Pin Functions (100-Pin LFQFP) (5/6)

Pin Number 100-Pin LFQFP	Power Supply Clock System Control	I/O Port	Timer (MTU, GPTW, TMR, POE, POEG, CAC, CMTW)	Communications (SCI, RSCI, RSPI, RSPIA, RIIC, RI3C, CANFD)	Interrupt (IRQ, NMI)	Analog (A/D, D/A, CMPC)
67		P22	MTIC5W/MTCLKD/ MTIC5W#/MTCLKD#/ TMRI2/TMO4/MTIOC9B/ GTIV	RXD12/SMISO12/SSCL12/ RXDX12/RXD008/ SMISO008/SSCL008/ SCK008/TXDB008/MISOA/ MISO0/CRX0	IRQ10	ADTRG2#/ COMP2
68		P21	MTIOC9A/MTCLKA/ MTIOC9A#/MTCLKA#/ TMCI4/TMO6/GTIU	TXD12/SMOSI12/SSDA12/ TXDX12/SIOX12/TXD008/ TXDA008/SMOSI008/ SSDA008/MOSIA/MOSI0	IRQ6	AN217/ ADTRG1#/ COMP5
69		P20	MTIOC9C/MTCLKB/ MTIOC9C#/MTCLKB#/ TMRI4/TMO2/GTIW	CTS008#/RTS008#/SS008#/ RXD008/SMISO008/ SSCL008/DE008/RSPCKA/ RSPCK0	IRQ7	AN216/ ADTRG0#/ COMP4
70		P65			IRQ9	AN211/ CMPC53/DA1
71		P64			IRQ8	AN210/ CMPC33/DA0
72	AVCC2					
73	AVSS2					
74		P63			IRQ7	AN209/ CMPC23
75		P62			IRQ6	AN208/ CMPC43
76		P61			IRQ5	AN207/ CMPC13
77		P60			IRQ4	AN206/ CMPC03
78		P55			IRQ3	AN203/ CMPC32
79		P54			IRQ2	AN202/ CMPC22/ CVREFC1
80		P53			IRQ1	AN201/ CMPC12/ CVREFC0
81		P52			IRQ0	AN200/ CMPC02
82		P51				AN205/ CMPC52
83		P50				AN204/ CMPC42
84		P47				AN103
85		P46				AN102/ CMPC50/ CMPC51
86		P45				AN101/ CMPC40/ CMPC41
87		P44				AN100/ CMPC30/ CMPC31
88		P43				AN003

Table 1.5 List of Pin and Pin Functions (100-Pin LFQFP) (6/6)

Pin Number 100-Pin LFQFP	Power Supply Clock System Control	I/O Port	Timer (MTU, GPTW, TMR, POE, POEG, CAC, CMTW)	Communications (SCI, RSCI, RSPI, RSPIA, RIIC, RI3C, CANFD)	Interrupt (IRQ, NMI)	Analog (A/D, D/A, CMPC)
89		P42				AN002/ CMPC20/ CMPC21
90		P41				AN001/ CMPC10/ CMPC11
91		P40				AN000/ CMPC00/ CMPC01
92	AVCC1					
93	AVCC0					
94	AVSS0					
95	AVSS1					
96		P82	MTIC5U/MTIC5U#/TMO4	SCK6/SCK12	IRQ3	COMP5
97		P81	MTIC5V/MTIC5V#/TMCI4	TXD6/SMOSI6/SSDA6/ TXD12/SMOSI12/SSDA12/ TXDX12/SIOX12		COMP4
98		P80	MTIC5W/MTIC5W#/TMRI4	RXD6/SMISO6/SSCL6/ RXD12/SMISO12/SSCL12/ RXDX12	IRQ5	COMP3
99		P11	MTIOC3A/MTCLKC/ MTIOC3A#/MTCLKC#/ TMO3/POE9#/MTIOC9D/ GTIOC3B/GTETRGA/ GTIOC3B#/GTETRGC/ GTCPPO0/TOC3	SCK009/SCK008/TXDB009	IRQ1	
100		P10	MTIOC9B/MTCLKD/ MTIOC9B#/MTCLKD#/ TMRI3/POE12#/GTIOC3A/ GTETRGB/GTIOC3A#/ GTETRGD/GTIV/TIC3	CTS6#/RTS6#/SS6#/ TXD009/TXDA009/ SMOSI009/SSDA009	IRQ0	

1.6.2 80-Pin LFQFP

Table 1.6 List of Pin and Pin Functions (80-Pin LFQFP) (1/5)

Pin Number 80-Pin LFQFP	Power Supply Clock System Control	I/O Port	Timer (MTU, GPTW, TMR, POE, POEG, CAC, CMTW)	Communications (SCI, RSCI, RSPI, RSPIA, RIIC, RI3C, CANFD)	Interrupt (IRQ, NMI)	Analog (A/D, D/A, CMPC)
1	EMLE	PN7	MTIOC9D/MTIOC9D#		IRQ5	ADST0
2	VSS					
3		P00	MTIOC9A/MTIOC9A#/ CACREF/GTIU/TIC3	RXD12/SMISO12/SSCL12/ RXDX12/RXD009/ SMISO009/SSCL009	IRQ2	ADST1/ COMP0
4	VCL					
5	MD/FINED	PN6				
6		P01	MTIOC9C/MTIOC9C#/ POE12#/GTETRGA/ GTETRGB/GTETRGC/ GTETRGD/GTIW	TXD12/SMOSI12/SSDA12/ TXDX12/SIOX12/TXD009/ TXDA009/SMOSI009/ SSDA009	IRQ4	ADST2/ COMP1
7		PE4	MTCLKC/MTCLKC#/ POE10#/GTETRGA/ GTETRGB/GTETRGC/ GTETRGD	SCK009/TXDB009	IRQ1	
8		PE3	MTCLKD/MTCLKD#/ POE11#/GTETRGA/ GTETRGB/GTETRGC/ GTETRGD	CTS009#/RTS009#/SS009#/ DE009	IRQ2	
9	RES#					
10	XTAL	P37		RXD5/SMISO5/SSCL5		
11	VSS					
12	EXTAL	P36		TXD5/SMOSI5/SSDA5		
13	VCC					
14		PE2	POE10#		NMI/IRQ0	
15	TRST#	PD7	MTIOC9A/MTIOC9A#/ TMRI1/TMRI5/GTIOC0A/ GTIOC3A/GTIOC0A#/ GTIOC3A#/GTIU	TXD5/SMOSI5/SSDA5/ SCK009/TXD008/TXDA008/ SMOSI008/SSDA008/ TXDB009/SSLA1/SSL01/ CTX0	IRQ8	
16	TMS	PD6	MTIOC9C/MTIOC9C#/ TMO1/GTIOC0B/GTIOC3B/ GTIOC0B#/GTIOC3B#/ GTIW	CTS1#/RTS1#/SS1#/ RXD12/SMISO12/SSCL12/ RXDX12/CTS011#/ RTS011#/SS011#/DE011/ SSLA0/SSL00	IRQ5	ADST0
17	TDI	PD5	TMRI0/TMRI6/GTIOC1A/ GTETRGA/GTIOC1A#/ GTIOC7A	RXD1/SMISO1/SSCL1/ RXD011/SMISO011/ SSCL011/SSL00	IRQ6	
18	TCK	PD4	TMCI0/TMCI6/GTIOC1B/ GTETRGB/GTIOC1B#	SCK1/TXD12/SMOSI12/ SSDA12/TXDX12/SIOX12/ SCK011/TXDB011/SSL02	IRQ2	
19	TDO	PD3	TMO0/GTIOC2A/GTETRGC/ GTIOC2A#/GTIOC7B	TXD1/SMOSI1/SSDA1/ TXD011/TXDA011/ SMOSI011/SSDA011/MOSI0		
20		PD2	TMCI1/TMO4/GTIOC2B/ GTIOC0A/GTIOC2B#/ GTIOC0A#	SCK5/SCK008/TXDB008/ MOSIA/MOSI0		

Table 1.6 List of Pin and Pin Functions (80-Pin LFQFP) (2/5)

Pin Number 80-Pin LFQFP	Power Supply Clock System Control	I/O Port	Timer (MTU, GPTW, TMR, POE, POEG, CAC, CMTW)	Communications (SCI, RSCI, RSPI, RSPIA, RIIC, RI3C, CANFD)	Interrupt (IRQ, NMI)	Analog (A/D, D/A, CMPC)
21		PB6	GTIOC2A/GTIOC3A/ GTIOC2A#/GTIOC3A#/ TOC0	RXD5/SMISO5/SSCL5/ RXD12/SMISO12/SSCL12/ RXDX12/RXD011/ SMISO011/SSCL011/MISO0/ CRX0	IRQ2	
22		PB5	GTIOC2B/GTIOC3B/ GTIOC2B#/GTIOC3B#/TIC0	TXD5/SMOSI5/SSDA5/ TXD12/SMOSI12/SSDA12/ TXDX12/SIOX12/TXD011/ TXDA011/SMOSI011/ SSDA011/RSPCK0/CTX0		
23	VCC					
24		PB4	POE8#/GTETRGA/ GTETRGB/GTETRGC/ GTETRGD/GTCPPO0	CTS5#/RTS5#/SS5#/ RXD12/SMISO12/SSCL12/ RXDX12/CTS011#/ RTS011#/SS011#/SCK011/ TXDB011/MISOA/SSL01/ CRX0	IRQ3	
25	VSS					
26		PB3	MTIOC0A/MTIOC0A#/ CACREF/GTIU/TOC1	SCK6/TXD12/SMOSI12/ SSDA12/TXDX12/SIOX12/ CTS009#/RTS009#/SS009#/ DE009/RSPCKA/CTX0	IRQ9	
27		PB2	MTIOC0B/MTIOC0B#/ TMRI0/GTADSM0/ GTIOC7A/GTIOC7A#/GTIV/ TIC1	TXD6/SMOSI6/SSDA6/ SDA0/SDA00		ADSM0
28		PB1	MTIOC0C/MTIOC0C#/ TMCI0/GTADSM1/ GTIOC7B/GTIOC7B#/GTIW/ TOC2	RXD6/SMISO6/SSCL6/ SCL0/SCL00	IRQ4	ADSM1
29		PB0	MTIOC0D/MTIOC0D#/ TMO0/TIC2	TXD6/SMOSI6/SSDA6/ TXD008/TXDA008/ SMOSI008/SSDA008/ CTS011#/RTS011#/SS011#/ DE011/MOSIA/MOSI0	IRQ8	ADTRG2#
30		PA5	MTIOC1A/MTIOC1A#/ TMCI3	RXD6/SMISO6/SSCL6/ RXD008/SMISO008/ SSCL008/MISOA/MISO0	IRQ1	ADTRG1#
31		PA3	MTIOC2A/MTIOC2A#/ TMRI7/GTADSM0	TXD009/TXDA009/ SMOSI009/SSDA009/ SCK008/TXDB008/SSLA0/ SSL00		
32	VCC					
33		P96	POE4#/GTETRGA/ GTETRGB/GTETRGC/ GTETRGD/GTCPPO4	CTS008#/RTS008#/SS008#/ DE008/SSL03/RSPCK0	IRQ4	
34	VSS					
35		P95	MTIOC6B/MTIOC1A/ MTIOC6B#/MTIOC1A#/ TMCI3/GTIOC4A/GTIOC7A/ GTIOC4A#/GTIOC7A#/ GTOUUP	RXD6/SMISO6/SSCL6/ RXD008/SMISO008/ SSCL008/MISOA/SSL02/ MISO0	IRQ1	ADTRG1#

Table 1.6 List of Pin and Pin Functions (80-Pin LFQFP) (3/5)

Pin Number 80-Pin LFQFP	Power Supply Clock System Control	I/O Port	Timer (MTU, GPTW, TMR, POE, POEG, CAC, CMTW)	Communications (SCI, RSCI, RSPI, RSPIA, RIIC, RI3C, CANFD)	Interrupt (IRQ, NMI)	Analog (A/D, D/A, CMPC)
36		P94	MTIOC7A/MTIOC2A/ MTIOC7A#/MTIOC2A#/ TMRI7/GTIOC5A/ GTADSM0/GTIOC5A#/ GTOVUP	TXD009/TXDA009/ SMOSI009/SSDA009/ SCK008/TXDB008/SSLA0/ SSL00		
37		P93	MTIOC7B/MTIOC6A/ MTIOC7B#/MTIOC6A#/ TMO4/GTIOC6A/GTIOC6A#/ GTOWUP	TXD009/TXDA009/ SMOSI009/SSDA009/ RXD011/SMISO011/ SSCL011/SSLA2/SSL02/ MOSI0/CRX0	IRQ14	ADTRG0#
38		P92	MTIOC6D/MTIOC6C/ MTIOC6D#/MTIOC6C#/ TMO2/GTIOC4B/GTIOC7B/ GTIOC4B#/GTIOC7B#/ GTOULO	SCK009/TXD011/TXDA011/ SMOSI011/SSDA011/ TXDB009/SSLA3/SSL03/ MISO0/CTX0		
39		P91	MTIOC7C/MTIOC7C#/ GTIOC5B/GTIOC5B#/ GTOVLO	RXD5/SMISO5/SSCL5/ RSPCK0		
40		P90	MTIOC7D/MTIOC7D#/ GTIOC6B/GTIOC6B#/ GTOWLO	TXD5/SMOSI5/SSDA5/ SSL01		
41		P76	MTIOC4D/MTIOC4D#/ GTIOC2B/GTIOC6B/ GTIOC2B#/GTIOC6B#/ GTOWLO	SSL03		
42		P75	MTIOC4C/MTIOC4C#/ GTIOC1B/GTIOC5B/ GTIOC1B#/GTIOC5B#/ GTOVLO	SSL02		
43		P74	MTIOC3D/MTIOC3D#/ GTIOC0B/GTIOC4B/ GTIOC0B#/GTIOC4B#/ GTOULO	SSL01		
44		P73	MTIOC4B/MTIOC4B#/ GTIOC2A/GTIOC6A/ GTIOC2A#/GTIOC6A#/ GTOWUP	SSL00		
45		P72	MTIOC4A/MTIOC4A#/ GTIOC1A/GTIOC5A/ GTIOC1A#/GTIOC5A#/ GTOVUP	MOSI0		
46		P71	MTIOC3B/MTIOC3B#/ GTIOC0A/GTIOC4A/ GTIOC0A#/GTIOC4A#/ GTOUUP	MISO0		
47		P70	MTIOC0A/MTCLKC/ MTIOC0A#/MTCLKC#/ TMRI6/POE0#/GTETRGA/ GTETRGB/GTETRGC/ GTETRGD/GTCPPO0	SCK5/CTS009#/RTS009#/ SS009#/DE009/SSLA0/ RSPCK0	IRQ5	
48	VCC					
49		P31	MTIOC0A/MTCLKC/ MTIOC0A#/MTCLKC#/ TMRI6/GTIU	SSLA1/SSL01	IRQ6	
50	VSS					

Table 1.6 List of Pin and Pin Functions (80-Pin LFQFP) (4/5)

Pin Number 80-Pin LFQFP	Power Supply Clock System Control	I/O Port	Timer (MTU, GPTW, TMR, POE, POEG, CAC, CMTW)	Communications (SCI, RSCI, RSPI, RSPIA, RIIC, RI3C, CANFD)	Interrupt (IRQ, NMI)	Analog (A/D, D/A, CMPC)
51		P30	MTIOC0B/MTCLKD/ MTIOC0B#/MTCLKD#/ TMCI6/GTIV	SCK008/CTS008#/ RTS008#/SS008#/DE008/ SSLA0/SSL00	IRQ7	COMP3
52		P27	MTIOC1A/MTIOC0C/ MTIOC1A#/MTIOC0C#/ TMO2/TMO6/POE9#	RSPCKA/RSPCK0	IRQ15	
53		P22	MTIC5W/MTCLKD/ MTIC5W#/MTCLKD#/ TMRI2/TMO4/MTIOC9B/ GTIV	RXD12/SMISO12/SSCL12/ RXDX12/RXD008/ SMISO008/SSCL008/ SCK008/TXDB008/MISOA/ MISO0/CRX0	IRQ10	ADTRG2#/ COMP2
54		P21	MTIOC9A/MTCLKA/ MTIOC9A#/MTCLKA#/ TMCI4/TMO6/GTIU	TXD12/SMOSI12/SSDA12/ TXDX12/SIOX12/TXD008/ TXDA008/SMOSI008/ SSDA008/MOSIA/MOSI0	IRQ6	AN217/ ADTRG1#/ COMP5
55		P20	MTIOC9C/MTCLKB/ MTIOC9C#/MTCLKB#/ TMRI4/TMO2/GTIW	CTS008#/RTS008#/SS008#/ RXD008/SMISO008/ SSCL008/DE008/RSPCKA/ RSPCK0	IRQ7	AN216/ ADTRG0#/ COMP4
56		P65			IRQ9	AN211/ CMPC53/DA1
57		P64			IRQ8	AN210/ CMPC33/DA0
58	AVCC2					
59	AVSS2					
60		P60			IRQ4	AN206/ CMPC03
61		P55			IRQ3	AN203/ CMPC32
62		P54			IRQ2	AN202/ CMPC22/ CVREFC1
63		P53			IRQ1	AN201/ CMPC12/ CVREFC0
64		P52			IRQ0	AN200/ CMPC02
65		P51				AN205/ CMPC52
66		P50				AN204/ CMPC42
67		P47				AN103
68		P46				AN102/ CMPC50/ CMPC51
69		P45				AN101/ CMPC40/ CMPC41
70		P44				AN100/ CMPC30/ CMPC31
71		P43				AN003

Table 1.6 List of Pin and Pin Functions (80-Pin LFQFP) (5/5)

Pin Number 80-Pin LFQFP	Power Supply Clock System Control	I/O Port	Timer (MTU, GPTW, TMR, POE, POEG, CAC, CMTW)	Communications (SCI, RSCI, RSPI, RSPIA, RIIC, RI3C, CANFD)	Interrupt (IRQ, NMI)	Analog (A/D, D/A, CMPC)
72		P42				AN002/ CMPC20/ CMPC21
73		P41				AN001/ CMPC10/ CMPC11
74		P40				AN000/ CMPC00/ CMPC01
75	AVCC1					
76	AVCC0					
77	AVSS0					
78	AVSS1					
79		P11	MTIOC3A/MTCLKC/ MTIOC3A#/MTCLKC#/ TMO3/POE9#/MTIOC9D/ GTIOC3B/GTETRGA/ GTIOC3B#/GTETRGC/ GTCPPO0/TOC3	SCK009/SCK008/TXDB009	IRQ1	
80		P10	MTIOC9B/MTCLKD/ MTIOC9B#/MTCLKD#/ TMRI3/POE12#/GTIOC3A/ GTETRGB/GTIOC3A#/ GTETRGD/GTIV/TIC3	CTS6#/RTS6#/SS6#/ TXD009/TXDA009/ SMOSI009/SSDA009	IRQ0	

1.6.3 64-Pin LFQFP, 64-Pin HWQFN (Products with 64 Kbytes of RAM)

Table 1.7 List of Pin and Pin Functions (64-Pin LFQFP, 64-Pin HWQFN) (Products with 64 Kbytes of RAM) (1/4)

Pin Number 64-Pin LFQFP, HWQFN	Power Supply Clock System Control	I/O Port	Timer (MTU, GPTW, TMR, POE, POEG, CAC, CMTW)	Communications (SCI, RSCI, RSPI, RSPIA, RIIC, RI3C, CANFD)	Interrupt (IRQ, NMI)	Analog (A/D, D/A, CMPC)
1	EMLE	PN7	MTIOC9D/MTIOC9D#		IRQ5	ADST0
2		P00	MTIOC9A/MTIOC9A#/ CACREF/GTIU/TIC3	RXD12/SMISO12/SSCL12/ RXDX12/RXD009/ SMISO009/SSCL009	IRQ2	ADST1/ COMP0
3	VCL					
4	MD/FINED	PN6				
5		P01	MTIOC9C/MTIOC9C#/ POE12#/GTETRGA/ GTETRGB/GTETRGC/ GTETRGD/GTIW	TXD12/SMOSI12/SSDA12/ TXDX12/SIOX12/TXD009/ TXDA009/SMOSI009/ SSDA009	IRQ4	ADST2/ COMP1
6	RES#					
7	XTAL	P37		RXD5/SMISO5/SSCL5		
8	VSS					
9	EXTAL	P36		TXD5/SMOSI5/SSDA5		
10	VCC					
11		PE2	POE10#		NMI/IRQ0	
12	TRST#	PD7	MTIOC9A/MTIOC9A#/ TMRI1/TMRI5/GTIOC0A/ GTIOC3A/GTIOC0A#/ GTIOC3A#/GTIU	TXD5/SMOSI5/SSDA5/ SCK009/TXD008/TXDA008/ SMOSI008/SSDA008/ TXDB009/SSLA1/SSL01/ CTX0	IRQ8	
13	TMS	PD6	MTIOC9C/MTIOC9C#/ TMO1/GTIOC0B/GTIOC3B/ GTIOC0B#/GTIOC3B#/ GTIW	CTS1#/RTS1#/SS1#/ RXD12/SMISO12/SSCL12/ RXDX12/CTS011#/ RTS011#/SS011#/DE011/ SSLA0/SSL00	IRQ5	ADST0
14	TDI	PD5	TMRI0/TMRI6/GTIOC1A/ GTETRGA/GTIOC1A#/ GTIOC7A	RXD1/SMISO1/SSCL1/ RXD011/SMISO011/ SSCL011/SSL00	IRQ6	
15	TCK	PD4	TMCI0/TMCI6/GTIOC1B/ GTETRGB/GTIOC1B#	SCK1/TXD12/SMOSI12/ SSDA12/TXDX12/SIOX12/ SCK011/TXDB011/SSL02	IRQ2	
16	TDO	PD3	TMO0/GTIOC2A/GTETRGC/ GTIOC2A#/GTIOC7B	TXD1/SMOSI1/SSDA1/ TXD011/TXDA011/ SMOSI011/SSDA011/MOSI0		
17		PB6	GTIOC2A/GTIOC3A/ GTIOC2A#/GTIOC3A#/ TOC0	RXD5/SMISO5/SSCL5/ RXD12/SMISO12/SSCL12/ RXDX12/RXD011/ SMISO011/SSCL011/MISO0/ CRX0	IRQ2	
18		PB5	GTIOC2B/GTIOC3B/ GTIOC2B#/GTIOC3B#/TIC0	TXD5/SMOSI5/SSDA5/ TXD12/SMOSI12/SSDA12/ TXDX12/SIOX12/TXD011/ TXDA011/SMOSI011/ SSDA011/RSPCK0/CTX0		

Table 1.7 List of Pin and Pin Functions (64-Pin LFQFP, 64-Pin HWQFN) (Products with 64 Kbytes of RAM) (2/4)

Pin Number 64-Pin LFQFP, HWQFN	Power Supply Clock System Control	I/O Port	Timer (MTU, GPTW, TMR, POE, POEG, CAC, CMTW)	Communications (SCI, RSCI, RSPI, RSPIA, RIIC, RI3C, CANFD)	Interrupt (IRQ, NMI)	Analog (A/D, D/A, CMPC)
19		PB4	POE8#/GTETRGA/ GTETRGB/GTETRGC/ GTETRGD/GTCPPO0	CTS5#/RTS5#/SS5#/ RXD12/SMISO12/SSCL12/ RXDX12/CTS011#/ RTS011#/SS011#/SCK011/ TXDB011/MISOA/SSL01/ CRX0	IRQ3	
20		PB3	MTIOC0A/MTIOC0A#/ CACREF/GTIU/TOC1	SCK6/TXD12/SMOSI12/ SSDA12/TXDX12/SIOX12/ CTS009#/RTS009#/SS009#/ DE009/RSPCKA/CTX0	IRQ9	
21		PB2	MTIOC0B/MTIOC0B#/ TMRI0/GTADSM0/ GTIOC7A/GTIOC7A#/GTIV/ TIC1	TXD6/SMOSI6/SSDA6/ SDA0/SDA00		ADSM0
22		PB1	MTIOC0C/MTIOC0C#/ TMCI0/GTADSM1/ GTIOC7B/GTIOC7B#/GTIW/ TOC2	RXD6/SMISO6/SSCL6/ SCL0/SCL00	IRQ4	ADSM1
23		PB0	MTIOC0D/MTIOC0D#/ TMO0/TIC2	TXD6/SMOSI6/SSDA6/ TXD008/TXDA008/ SMOSI008/SSDA008/ CTS011#/RTS011#/SS011#/ DE011/MOSIA/MOSI0	IRQ8	ADTRG2#
24	VCC					
25		P96	POE4#/GTETRGA/ GTETRGB/GTETRGC/ GTETRGD/GTCPPO4	CTS008#/RTS008#/SS008#/ DE008/SSL03/RSPCK0	IRQ4	
26	VSS					
27		P95	MTIOC6B/MTIOC1A/ MTIOC6B#/MTIOC1A#/ TMCI3/GTIOC4A/GTIOC7A/ GTIOC4A#/GTIOC7A#/ GTOUUP	RXD6/SMISO6/SSCL6/ RXD008/SMISO008/ SSCL008/MISOA/SSL02/ MISO0	IRQ1	ADTRG1#
28		P94	MTIOC7A/MTIOC2A/ MTIOC7A#/MTIOC2A#/ TMRI7/GTIOC5A/ GTADSM0/GTIOC5A#/ GTOVUP	TXD009/TXDA009/ SMOSI009/SSDA009/ SCK008/TXDB008/SSLA0/ SSL00		
29		P93	MTIOC7B/MTIOC6A/ MTIOC7B#/MTIOC6A#/ TMO4/GTIOC6A/GTIOC6A#/ GTOWUP	TXD009/TXDA009/ SMOSI009/SSDA009/ RXD011/SMISO011/ SSCL011/SSLA2/SSL02/ MOSI0/CRX0	IRQ14	ADTRG0#
30		P92	MTIOC6D/MTIOC6C/ MTIOC6D#/MTIOC6C#/ TMO2/GTIOC4B/GTIOC7B/ GTIOC4B#/GTIOC7B#/ GTOULO	SCK009/TXD011/TXDA011/ SMOSI011/SSDA011/ TXDB009/SSLA3/SSL03/ MISO0/CTX0		
31		P91	MTIOC7C/MTIOC7C#/ GTIOC5B/GTIOC5B#/ GTOVLO	RXD5/SMISO5/SSCL5/ RSPCK0		
32		P90	MTIOC7D/MTIOC7D#/ GTIOC6B/GTIOC6B#/ GTOWLO	TXD5/SMOSI5/SSDA5/ SSL01		

Table 1.7 List of Pin and Pin Functions (64-Pin LFQFP, 64-Pin HWQFN) (Products with 64 Kbytes of RAM) (3/4)

Pin Number 64-Pin LFQFP, HWQFN	Power Supply Clock System Control	I/O Port	Timer (MTU, GPTW, TMR, POE, POEG, CAC, CMTW)	Communications (SCI, RSCI, RSPI, RSPIA, RIIC, RI3C, CANFD)	Interrupt (IRQ, NMI)	Analog (A/D, D/A, CMPC)
33		P76	MTIOC4D/MTIOC4D#/ GTIOC2B/GTIOC6B/ GTIOC2B#/GTIOC6B#/ GTOWLO	SSL03		
34		P75	MTIOC4C/MTIOC4C#/ GTIOC1B/GTIOC5B/ GTIOC1B#/GTIOC5B#/ GTOVLO	SSL02		
35		P74	MTIOC3D/MTIOC3D#/ GTIOC0B/GTIOC4B/ GTIOC0B#/GTIOC4B#/ GTOULO	SSL01		
36		P73	MTIOC4B/MTIOC4B#/ GTIOC2A/GTIOC6A/ GTIOC2A#/GTIOC6A#/ GTOWUP	SSL00		
37		P72	MTIOC4A/MTIOC4A#/ GTIOC1A/GTIOC5A/ GTIOC1A#/GTIOC5A#/ GTOVUP	MOSI0		
38		P71	MTIOC3B/MTIOC3B#/ GTIOC0A/GTIOC4A/ GTIOC0A#/GTIOC4A#/ GTOUUP	MISO0		
39		P70	MTIOC0A/MTCLKC/ MTIOC0A#/MTCLKC#/ TMRI6/POE0#/GTETRGA/ GTETRGB/GTETRGC/ GTETRGD/GTCPPO0	SCK5/CTS009#/RTS009#/ SS009#/DE009/SSLA0/ RSPCK0	IRQ5	
40	VCC					
41	VSS					
42		P22	MTIC5W/MTCLKD/ MTIC5W#/MTCLKD#/ TMRI2/TMO4/MTIOC9B/ GTIV	RXD12/SMISO12/SSCL12/ RXDX12/RXD008/ SMISO008/SSCL008/ SCK008/TXDB008/MISOA/ MISO0/CRX0	IRQ10	ADTRG2#/ COMP2
43		P21	MTIOC9A/MTCLKA/ MTIOC9A#/MTCLKA#/ TMCI4/TMO6/GTIU	TXD12/SMOSI12/SSDA12/ TXDX12/SIOX12/TXD008/ TXDA008/SMOSI008/ SSDA008/MOSIA/MOSI0	IRQ6	AN217/ ADTRG1#/ COMP5
44		P20	MTIOC9C/MTCLKB/ MTIOC9C#/MTCLKB#/ TMRI4/TMO2/GTIW	CTS008#/RTS008#/SS008#/ RXD008/SMISO008/ SSCL008/DE008/RSPCKA/ RSPCK0	IRQ7	AN216/ ADTRG0#/ COMP4
45		P65			IRQ9	AN211/ CMPC53/DA1
46		P64			IRQ8	AN210/ CMPC33/DA0
47	AVCC2					
48	AVSS2					
49		P54			IRQ2	AN202/ CMPC22/ CVREFC1

Table 1.7 List of Pin and Pin Functions (64-Pin LFQFP, 64-Pin HWQFN) (Products with 64 Kbytes of RAM) (4/4)

Pin Number 64-Pin LFQFP, HWQFN	Power Supply Clock System Control	I/O Port	Timer (MTU, GPTW, TMR, POE, POEG, CAC, CMTW)	Communications (SCI, RSCI, RSPI, RSPIA, RIIC, RI3C, CANFD)	Interrupt (IRQ, NMI)	Analog (A/D, D/A, CMPC)
50		P53			IRQ1	AN201/ CMPC12/ CVREFC0
51		P52			IRQ0	AN200/ CMPC02
52		P47				AN103
53		P46				AN102/ CMPC50/ CMPC51
54		P45				AN101/ CMPC40/ CMPC41
55		P44				AN100/ CMPC30/ CMPC31
56		P43				AN003
57		P42				AN002/ CMPC20/ CMPC21
58		P41				AN001/ CMPC10/ CMPC11
59		P40				AN000/ CMPC00/ CMPC01
60	AVCC1					
61	AVCC0					
62	AVSS0					
63	AVSS1					
64		P11	MTIOC3A/MTCLKC/ MTIOC3A#/MTCLKC#/ TMO3/POE9#/MTIOC9D/ GTIOC3B/GTETRGA/ GTIOC3B#/GTETRGC/ GTCPPO0/TOC3	SCK009/SCK008/TXDB009	IRQ1	

1.6.4 64-Pin LFQFP (Products with 48 Kbytes of RAM)

Table 1.8 List of Pin and Pin Functions (64-Pin LFQFP) (Products with 48 Kbytes of RAM) (1/4)

Pin Number 64-Pin LFQFP	Power Supply Clock System Control	I/O Port	Timer (MTU, GPTW, TMR, POE, POEG, CAC, CMTW)	Communications (SCI, RSPI, RIIC, CANFD)	Interrupt (IRQ, NMI)	Analog (A/D, D/A, CMPC)
1	EMLE	PN7	MTIOC9D/MTIOC9D#		IRQ5	ADST0
2		P00	MTIOC9A/MTIOC9A#/ CACREF/GTIU/TIC3	RXD12/SMISO12/SSCL12/ RXDX12	IRQ2	COMP0
3	VCL					
4	MD/FINED	PN6				
5		P01	MTIOC9C/MTIOC9C#/ POE12#/GTETRGA/ GTETRGB/GTETRGC/ GTETRGD/GTIW	TXD12/SMOSI12/SSDA12/ TXDX12/SIOX12	IRQ4	ADST2/ COMP1
6	RES#					
7	XTAL	P37		RXD5/SMISO5/SSCL5		
8	VSS					
9	EXTAL	P36		TXD5/SMOSI5/SSDA5		
10	VCC					
11		PE2	POE10#		NMI/IRQ0	
12	TRST#	PD7	MTIOC9A/MTIOC9A#/ TMRI1/TMRI5/GTIOC0A/ GTIOC3A/GTIOC0A#/ GTIOC3A#/GTIU	TXD5/SMOSI5/SSDA5/ SSLA1/CTX0	IRQ8	
13	TMS	PD6	MTIOC9C/MTIOC9C#/ TMO1/GTIOC0B/GTIOC3B/ GTIOC0B#/GTIOC3B#/ GTIW	CTS1#/RTS1#/SS1#/ RXD12/SMISO12/SSCL12/ RXDX12/SSLA0	IRQ5	ADST0
14	TDI	PD5	TMRI0/TMRI6/GTIOC1A/ GTETRGA/GTIOC1A#/ GTIOC7A	RXD1/SMISO1/SSCL1	IRQ6	
15	TCK	PD4	TMCI0/TMCI6/GTIOC1B/ GTETRGB/GTIOC1B#	SCK1/TXD12/SMOSI12/ SSDA12/TXDX12/SIOX12	IRQ2	
16	TDO	PD3	TMO0/GTIOC2A/GTETRGC/ GTIOC2A#/GTIOC7B	TXD1/SMOSI1/SSDA1		
17		PB6	GTIOC2A/GTIOC3A/ GTIOC2A#/GTIOC3A#/ TOC0	RXD5/SMISO5/SSCL5/ RXD12/SMISO12/SSCL12/ RXDX12/CRX0	IRQ2	
18		PB5	GTIOC2B/GTIOC3B/ GTIOC2B#/GTIOC3B#/TIC0	TXD5/SMOSI5/SSDA5/ TXD12/SMOSI12/SSDA12/ TXDX12/SIOX12/CTX0		
19		PB4	POE8#/GTETRGA/ GTETRGB/GTETRGC/ GTETRGD/GTCPPO0	CTS5#/RTS5#/SS5#/ RXD12/SMISO12/SSCL12/ RXDX12/MISOA/CRX0	IRQ3	
20		PB3	MTIOC0A/MTIOC0A#/ CACREF/GTIU/TOC1	SCK6/TXD12/SMOSI12/ SSDA12/TXDX12/SIOX12/ RSPCKA/CTX0	IRQ9	
21		PB2	MTIOC0B/MTIOC0B#/ TMRI0/GTADSM0/ GTIOC7A/GTIOC7A#/GTIV/ TIC1	TXD6/SMOSI6/SSDA6/ SDA0		ADSM0
22		PB1	MTIOC0C/MTIOC0C#/ TMCI0/GTADSM1/ GTIOC7B/GTIOC7B#/GTIW/ TOC2	RXD6/SMISO6/SSCL6/SCL0	IRQ4	ADSM1

Table 1.8 List of Pin and Pin Functions (64-Pin LFQFP) (Products with 48 Kbytes of RAM) (2/4)

Pin Number 64-Pin LFQFP	Power Supply Clock System Control	I/O Port	Timer (MTU, GPTW, TMR, POE, POEG, CAC, CMTW)	Communications (SCI, RSPI, RIIC, CANFD)	Interrupt (IRQ, NMI)	Analog (A/D, D/A, CMPC)
23		PB0	MTIOC0D/MTIOC0D#/ TMO0/TIC2	TXD6/SMOSI6/SSDA6/ MOSIA	IRQ8	ADTRG2#
24	VCC					
25		P96	POE4#/GTETRGA/ GTETRGB/GTETRGC/ GTETRGD/GTCPPO4		IRQ4	
26	VSS					
27		P95	MTIOC6B/MTIOC1A/ MTIOC6B#/MTIOC1A#/ TMCI3/GTIOC4A/GTIOC7A/ GTIOC4A#/GTIOC7A#/ GTOUUP	RXD6/SMISO6/SSCL6/ MISOA	IRQ1	
28		P94	MTIOC7A/MTIOC2A/ MTIOC7A#/MTIOC2A#/ TMRI7/GTIOC5A/ GTADSM0/GTIOC5A#/ GTOVUP	SSLA0		
29		P93	MTIOC7B/MTIOC6A/ MTIOC7B#/MTIOC6A#/ TMO4/GTIOC6A/GTIOC6A#/ GTOWUP	SSLA2/CRX0	IRQ14	ADTRG0#
30		P92	MTIOC6D/MTIOC6C/ MTIOC6D#/MTIOC6C#/ TMO2/GTIOC4B/GTIOC7B/ GTIOC4B#/GTIOC7B#/ GTOULO	SSLA3/CTX0		
31		P91	MTIOC7C/MTIOC7C#/ GTIOC5B/GTIOC5B#/ GTOVLO	RXD5/SMISO5/SSCL5		
32		P90	MTIOC7D/MTIOC7D#/ GTIOC6B/GTIOC6B#/ GTOWLO	TXD5/SMOSI5/SSDA5		
33		P76	MTIOC4D/MTIOC4D#/ GTIOC2B/GTIOC6B/ GTIOC2B#/GTIOC6B#/ GTOWLO			
34		P75	MTIOC4C/MTIOC4C#/ GTIOC1B/GTIOC5B/ GTIOC1B#/GTIOC5B#/ GTOVLO			
35		P74	MTIOC3D/MTIOC3D#/ GTIOC0B/GTIOC4B/ GTIOC0B#/GTIOC4B#/ GTOULO			
36		P73	MTIOC4B/MTIOC4B#/ GTIOC2A/GTIOC6A/ GTIOC2A#/GTIOC6A#/ GTOWUP			
37		P72	MTIOC4A/MTIOC4A#/ GTIOC1A/GTIOC5A/ GTIOC1A#/GTIOC5A#/ GTOVUP			
38		P71	MTIOC3B/MTIOC3B#/ GTIOC0A/GTIOC4A/ GTIOC0A#/GTIOC4A#/ GTOUUP			

Table 1.8 List of Pin and Pin Functions (64-Pin LFQFP) (Products with 48 Kbytes of RAM) (3/4)

Pin Number 64-Pin LFQFP	Power Supply Clock System Control	I/O Port	Timer (MTU, GPTW, TMR, POE, POEG, CAC, CMTW)	Communications (SCI, RSPI, RIIC, CANFD)	Interrupt (IRQ, NMI)	Analog (A/D, D/A, CMPC)
39		P70	MTIOC0A/MTCLKC/ MTIOC0A#/MTCLKC#/ TMRI6/POE0#/GTETRGA/ GTETRGB/GTETRGC/ GTETRGD/GTCPPO0	SCK5/SSLA0	IRQ5	
40	VCC					
41	VSS					
42		P22	MTIC5W/MTCLKD/ MTIC5W#/MTCLKD#/ TMRI2/TMO4/MTIOC9B/ GTIV	RXD12/SMISO12/SSCL12/ RXDX12/MISOA/CRX0	IRQ10	ADTRG2#/ COMP2
43		P21	MTIOC9A/MTCLKA/ MTIOC9A#/MTCLKA#/ TMCI4/TMO6/GTIU	TXD12/SMOSI12/SSDA12/ TXDX12/SIOX12/MOSIA	IRQ6	AN217/ COMP5
44		P20	MTIOC9C/MTCLKB/ MTIOC9C#/MTCLKB#/ TMRI4/TMO2/GTIW	RSPCKA	IRQ7	AN216/ ADTRG0#/ COMP4
45		P65			IRQ9	AN211/ CMPC53/DA1
46		P64			IRQ8	AN210/ CMPC52/DA0
47	AVCC2					
48	AVSS2					
49		P54			IRQ2	AN202/ CMPC22/ CVREFC1
50		P53			IRQ1	AN201/ CMPC12/ CVREFC0
51		P52			IRQ0	AN200/ CMPC02
52		P47				AN206/ CMPC03
53		P46				AN006/ CMPC21
54		P45				AN005/ CMPC11
55		P44				AN004/ CMPC01
56		P43				AN003/ CMPC23/ CMPC50
57		P42				AN002/ CMPC20
58		P41				AN001/ CMPC10
59		P40				AN000/ CMPC13/ CMPC00
60	NC					
61	AVCC0					
62	AVSS0					

Table 1.8 List of Pin and Pin Functions (64-Pin LFQFP) (Products with 48 Kbytes of RAM) (4/4)

Pin Number 64-Pin LFQFP	Power Supply Clock System Control	I/O Port	Timer (MTU, GPTW, TMR, POE, POEG, CAC, CMTW)	Communications (SCI, RSPI, RIIC, CANFD)	Interrupt (IRQ, NMI)	Analog (A/D, D/A, CMPC)
63	NC					
64		P11	MTIOC3A/MTCLKC/ MTIOC3A#/MTCLKC#/ TMO3/POE9#/MTIOC9D/ GTIOC3B/GTETRGA/ GTIOC3B#/GTETRGC/ GTCPPO0/TOC3		IRQ1	

1.6.5 48-Pin LFQFP, 48-Pin HWQFN (Products with 64 Kbytes of RAM)

Table 1.9 List of Pin and Pin Functions (48-Pin LFQFP, 48-Pin HWQFN) (Products with 64 Kbytes of RAM) (1/3)

Pin Number 48-Pin LFQFP, HWQFN	Power Supply Clock System Control	I/O Port	Timer (MTU, GPTW, TMR, POE, POEG, CAC, CMTW)	Communications (SCI, RSCI, RSPI, RSPIA, RIIC, RI3C, CANFD)	Interrupt (IRQ, NMI)	Analog (A/D, D/A, CMPC)
1		P00	MTIOC9A/MTIOC9A#/ CACREF/GTIU/TIC3	RXD12/SMISO12/SSCL12/ RXDX12/RXD009/ SMISO009/SSCL009	IRQ2	ADST1/ COMP0
2	VCL					
3	MD/FINED	PN6				
4	RES#					
5	XTAL	P37		RXD5/SMISO5/SSCL5		
6	VSS					
7	EXTAL	P36		TXD5/SMOSI5/SSDA5		
8	VCC					
9		PE2	POE10#		NMI/IRQ0	
10	TRST#	PD7	MTIOC9A/MTIOC9A#/ TMRI1/TMRI5/GTIOC0A/ GTIOC3A/GTIOC0A#/ GTIOC3A#/GTIU	TXD5/SMOSI5/SSDA5/ SCK009/TXD008/TXDA008/ SMOSI008/SSDA008/ TXDB009/SSLA1/SSL01/ CTX0	IRQ8	
11	TDI	PD5	TMRI0/TMRI6/GTIOC1A/ GTETRGA/GTIOC1A#/ GTIOC7A	RXD1/SMISO1/SSCL1/ RXD011/SMISO011/ SSCL011/SSL00	IRQ6	
12	TDO	PD3	TMO0/GTIOC2A/GTETRGC/ GTIOC2A#/GTIOC7B	TXD1/SMOSI1/SSDA1/ TXD011/TXDA011/ SMOSI011/SSDA011/MOSI0		
13		PB6	GTIOC2A/GTIOC3A/ GTIOC2A#/GTIOC3A#/ TOC0	RXD5/SMISO5/SSCL5/ RXD12/SMISO12/SSCL12/ RXDX12/RXD011/ SMISO011/SSCL011/MISO0/ CRX0	IRQ2	
14		PB5	GTIOC2B/GTIOC3B/ GTIOC2B#/GTIOC3B#/TIC0	TXD5/SMOSI5/SSDA5/ TXD12/SMOSI12/SSDA12/ TXDX12/SIOX12/TXD011/ TXDA011/SMOSI011/ SSDA011/RSPCK0/CTX0		
15		PB4	POE8#/GTETRGA/ GTETRGB/GTETRGC/ GTETRGD/GTCPPO0	CTS5#/RTS5#/SS5#/ RXD12/SMISO12/SSCL12/ RXDX12/CTS011#/ RTS011#/SS011#/SCK011/ TXDB011/MISOA/SSL01/ CRX0	IRQ3	
16		PB3	MTIOC0A/MTIOC0A#/ CACREF/GTIU/TOC1	SCK6/TXD12/SMOSI12/ SSDA12/TXDX12/SIOX12/ CTS009#/RTS009#/SS009#/ DE009/RSPCKA/CTX0	IRQ9	
17		PB2	MTIOC0B/MTIOC0B#/ TMRI0/GTADSM0/ GTIOC7A/GTIOC7A#/GTIV/ TIC1	TXD6/SMOSI6/SSDA6/ SDA0/SDA00		ADSM0
18		PB1	MTIOC0C/MTIOC0C#/ TMCI0/GTADSM1/ GTIOC7B/GTIOC7B#/GTIW/ TOC2	RXD6/SMISO6/SSCL6/ SCL0/SCL00	IRQ4	ADSM1

Table 1.9 List of Pin and Pin Functions (48-Pin LFQFP, 48-Pin HWQFN) (Products with 64 Kbytes of RAM) (2/3)

Pin Number 48-Pin LFQFP, HWQFN	Power Supply Clock System Control	I/O Port	Timer (MTU, GPTW, TMR, POE, POEG, CAC, CMTW)	Communications (SCI, RSCI, RSPI, RSPIA, RIIC, RI3C, CANFD)	Interrupt (IRQ, NMI)	Analog (A/D, D/A, CMPC)
19		PB0	MTIOC0D/MTIOC0D#/ TMO0/TIC2	TXD6/SMOSI6/SSDA6/ TXD008/TXDA008/ SMOSI008/SSDA008/ CTS011#/RTS011#/SS011#/ DE011/MOSIA/MOSI0	IRQ8	ADTRG2#
20		P95	MTIOC6B/MTIOC1A/ MTIOC6B#/MTIOC1A#/ TMCI3/GTIOC4A/GTIOC7A/ GTIOC4A#/GTIOC7A#/ GTOUUP	RXD6/SMISO6/SSCL6/ RXD008/SMISO008/ SSCL008/MISOA/SSL02/ MISO0	IRQ1	ADTRG1#
21		P94	MTIOC7A/MTIOC2A/ MTIOC7A#/MTIOC2A#/ TMRI7/GTIOC5A/ GTADSM0/GTIOC5A#/ GTOVUP	TXD009/TXDA009/ SMOSI009/SSDA009/ SCK008/TXDB008/SSLA0/ SSL00		
22		P93	MTIOC7B/MTIOC6A/ MTIOC7B#/MTIOC6A#/ TMO4/GTIOC6A/GTIOC6A#/ GTOWUP	TXD009/TXDA009/ SMOSI009/SSDA009/ RXD011/SMISO011/ SSCL011/SSLA2/SSL02/ MOSI0/CRX0	IRQ14	ADTRG0#
23		P92	MTIOC6D/MTIOC6C/ MTIOC6D#/MTIOC6C#/ TMO2/GTIOC4B/GTIOC7B/ GTIOC4B#/GTIOC7B#/ GTOULO	SCK009/TXD011/TXDA011/ SMOSI011/SSDA011/ TXDB009/SSLA3/SSL03/ MISO0/CTX0		
24		P91	MTIOC7C/MTIOC7C#/ GTIOC5B/GTIOC5B#/ GTOVLO	RXD5/SMISO5/SSCL5/ RSPCK0		
25		P76	MTIOC4D/MTIOC4D#/ GTIOC2B/GTIOC6B/ GTIOC2B#/GTIOC6B#/ GTOWLO	SSL03		
26		P75	MTIOC4C/MTIOC4C#/ GTIOC1B/GTIOC5B/ GTIOC1B#/GTIOC5B#/ GTOVLO	SSL02		
27		P74	MTIOC3D/MTIOC3D#/ GTIOC0B/GTIOC4B/ GTIOC0B#/GTIOC4B#/ GTOULO	SSL01		
28		P73	MTIOC4B/MTIOC4B#/ GTIOC2A/GTIOC6A/ GTIOC2A#/GTIOC6A#/ GTOWUP	SSL00		
29		P72	MTIOC4A/MTIOC4A#/ GTIOC1A/GTIOC5A/ GTIOC1A#/GTIOC5A#/ GTOVUP	MOSI0		
30		P71	MTIOC3B/MTIOC3B#/ GTIOC0A/GTIOC4A/ GTIOC0A#/GTIOC4A#/ GTOUUP	MISO0		
31	VCC					
32	VSS					

Table 1.9 List of Pin and Pin Functions (48-Pin LFQFP, 48-Pin HWQFN) (Products with 64 Kbytes of RAM) (3/3)

Pin Number 48-Pin LFQFP, HWQFN	Power Supply Clock System Control	I/O Port	Timer (MTU, GPTW, TMR, POE, POEG, CAC, CMTW)	Communications (SCI, RSCI, RSPI, RSPIA, RIIC, RI3C, CANFD)	Interrupt (IRQ, NMI)	Analog (A/D, D/A, CMPC)
33		P21	MTIOC9A/MTCLKA/ MTIOC9A#/MTCLKA#/ TMCI4/TMO6/GTIU	TXD12/SMOSI12/SSDA12/ TXDX12/SIOX12/TXD008/ TXDA008/SMOSI008/ SSDA008/MOSIA/MOSI0	IRQ6	AN217/ ADTRG1#/ COMP5
34		P20	MTIOC9C/MTCLKB/ MTIOC9C#/MTCLKB#/ TMRI4/TMO2/GTIW	CTS008#/RTS008#/SS008#/ RXD008/SMISO008/ SSCL008/DE008/RSPCKA/ RSPCK0	IRQ7	AN216/ ADTRG0#/ COMP4
35	AVCC2					
36	AVSS2					
37		P62			IRQ6	AN208/ CMPC43
38		P53			IRQ1	AN201/ CMPC12/ CVREFC0
39		P52			IRQ0	AN200/ CMPC02
40		P44				AN100/ CMPC30/ CMPC31
41		P43				AN003
42		P42				AN002/ CMPC20/ CMPC21
43		P41				AN001/ CMPC10/ CMPC11
44		P40				AN000/ CMPC00/ CMPC01
45	AVCC0/ AVCC1					
46	AVSS0/AVSS1					
47		P11	MTIOC3A/MTCLKC/ MTIOC3A#/MTCLKC#/ TMO3/POE9#/MTIOC9D/ GTIOC3B/GTETRGA/ GTIOC3B#/GTETRGC/ GTCPPO0/TOC3	SCK009/SCK008/TXDB009	IRQ1	
48		P10	MTIOC9B/MTCLKD/ MTIOC9B#/MTCLKD#/ TMRI3/POE12#/GTIOC3A/ GTETRGB/GTIOC3A#/ GTETRGD/GTIV/TIC3	CTS6#/RTS6#/SS6#/ TXD009/TXDA009/ SMOSI009/SSDA009	IRQ0	

1.6.6 48-Pin LFQFP (Products with 48 Kbytes of RAM)

Table 1.10 List of Pin and Pin Functions (48-Pin LFQFP) (Products with 48 Kbytes of RAM) (1/3)

Pin Number 48-Pin LFQFP	Power Supply Clock System Control	I/O Port	Timer (MTU, GPTW, TMR, POE, POEG, CAC, CMTW)	Communications (SCI, RSPI, RIIC, CANFD)	Interrupt (IRQ, NMI)	Analog (A/D, D/A, CMPC)
1		P00	MTIOC9A/MTIOC9A#/ CACREF/GTIU/TIC3	RXD12/SMISO12/SSCL12/ RXDX12	IRQ2	COMP0
2	VCL					
3	MD/FINED	PN6				
4	RES#					
5	XTAL	P37		RXD5/SMISO5/SSCL5		
6	VSS					
7	EXTAL	P36		TXD5/SMOSI5/SSDA5		
8	VCC					
9		PE2	POE10#		NMI/IRQ0	
10	TRST#	PD7	MTIOC9A/MTIOC9A#/ TMRI1/TMRI5/GTIOC0A/ GTIOC3A/GTIOC0A#/ GTIOC3A#/GTIU	TXD5/SMOSI5/SSDA5/ SSLA1/CTX0	IRQ8	
11	TDI	PD5	TMRI0/TMRI6/GTIOC1A/ GTETRGA/GTIOC1A#/ GTIOC7A	RXD1/SMISO1/SSCL1	IRQ6	
12	TDO	PD3	TMO0/GTIOC2A/GTETRGC/ GTIOC2A#/GTIOC7B	TXD1/SMOSI1/SSDA1		
13		PB6	GTIOC2A/GTIOC3A/ GTIOC2A#/GTIOC3A#/ TOC0	RXD5/SMISO5/SSCL5/ RXD12/SMISO12/SSCL12/ RXDX12/CRX0	IRQ2	
14		PB5	GTIOC2B/GTIOC3B/ GTIOC2B#/GTIOC3B#/TIC0	TXD5/SMOSI5/SSDA5/ TXD12/SMOSI12/SSDA12/ TXDX12/SIOX12/CTX0		
15		PB4	POE8#/GTETRGA/ GTETRGB/GTETRGC/ GTETRGD/GTCPPO0	CTS5#/RTS5#/SS5#/ RXD12/SMISO12/SSCL12/ RXDX12/MISOA/CRX0	IRQ3	
16		PB3	MTIOC0A/MTIOC0A#/ CACREF/GTIU/TOC1	SCK6/TXD12/SMOSI12/ SSDA12/TXDX12/SIOX12/ RSPCKA/CTX0	IRQ9	
17		PB2	MTIOC0B/MTIOC0B#/ TMRI0/GTADSM0/ GTIOC7A/GTIOC7A#/GTIV/ TIC1	TXD6/SMOSI6/SSDA6/ SDA0		ADSM0
18		PB1	MTIOC0C/MTIOC0C#/ TMCI0/GTADSM1/ GTIOC7B/GTIOC7B#/GTIW/ TOC2	RXD6/SMISO6/SSCL6/SCL0	IRQ4	ADSM1
19		PB0	MTIOC0D/MTIOC0D#/ TMO0/TIC2	TXD6/SMOSI6/SSDA6/ MOSIA	IRQ8	ADTRG2#
20		P95	MTIOC6B/MTIOC1A/ MTIOC6B#/MTIOC1A#/ TMCI3/GTIOC4A/GTIOC7A/ GTIOC4A#/GTIOC7A#/ GTOUUP	RXD6/SMISO6/SSCL6/ MISOA	IRQ1	

Table 1.10 List of Pin and Pin Functions (48-Pin LFQFP) (Products with 48 Kbytes of RAM) (2/3)

Pin Number 48-Pin LFQFP	Power Supply Clock System Control	I/O Port	Timer (MTU, GPTW, TMR, POE, POEG, CAC, CMTW)	Communications (SCI, RSPI, RIIC, CANFD)	Interrupt (IRQ, NMI)	Analog (A/D, D/A, CMPC)
21		P94	MTIOC7A/MTIOC2A/ MTIOC7A#/MTIOC2A#/ TMRI7/GTIOC5A/ GTADSM0/GTIOC5A#/ GTOVUP	SSLA0		
22		P93	MTIOC7B/MTIOC6A/ MTIOC7B#/MTIOC6A#/ TMO4/GTIOC6A/GTIOC6A#/ GTOWUP	SSLA2/CRX0	IRQ14	ADTRG0#
23		P92	MTIOC6D/MTIOC6C/ MTIOC6D#/MTIOC6C#/ TMO2/GTIOC4B/GTIOC7B/ GTIOC4B#/GTIOC7B#/ GTOULO	SSLA3/CTX0		
24		P91	MTIOC7C/MTIOC7C#/ GTIOC5B/GTIOC5B#/ GTOVLO	RXD5/SMISO5/SSCL5		
25		P76	MTIOC4D/MTIOC4D#/ GTIOC2B/GTIOC6B/ GTIOC2B#/GTIOC6B#/ GTOWLO			
26		P75	MTIOC4C/MTIOC4C#/ GTIOC1B/GTIOC5B/ GTIOC1B#/GTIOC5B#/ GTOVLO			
27		P74	MTIOC3D/MTIOC3D#/ GTIOC0B/GTIOC4B/ GTIOC0B#/GTIOC4B#/ GTOULO			
28		P73	MTIOC4B/MTIOC4B#/ GTIOC2A/GTIOC6A/ GTIOC2A#/GTIOC6A#/ GTOWUP			
29		P72	MTIOC4A/MTIOC4A#/ GTIOC1A/GTIOC5A/ GTIOC1A#/GTIOC5A#/ GTOVUP			
30		P71	MTIOC3B/MTIOC3B#/ GTIOC0A/GTIOC4A/ GTIOC0A#/GTIOC4A#/ GTOUUP			
31	VCC					
32	VSS					
33		P21	MTIOC9A/MTCLKA/ MTIOC9A#/MTCLKA#/ TMCI4/TMO6/GTIU	TXD12/SMOSI12/SSDA12/ TXDX12/SIOX12/MOSIA	IRQ6	AN217/ COMP5
34		P20	MTIOC9C/MTCLKB/ MTIOC9C#/MTCLKB#/ TMRI4/TMO2/GTIW	RSPCKA	IRQ7	AN216/ ADTRG0#/ COMP4
35	AVCC2					
36	AVSS2					
37		P62			IRQ6	AN208/ CMPC51

Table 1.10 List of Pin and Pin Functions (48-Pin LFQFP) (Products with 48 Kbytes of RAM) (3/3)

Pin Number 48-Pin LFQFP	Power Supply Clock System Control	I/O Port	Timer (MTU, GPTW, TMR, POE, POEG, CAC, CMTW)	Communications (SCI, RSPI, RIIC, CANFD)	Interrupt (IRQ, NMI)	Analog (A/D, D/A, CMPC)
38		P53			IRQ1	AN201/ CMPC12/ CVREFC0
39		P52			IRQ0	AN200/ CMPC02
40		P44				AN004/ CMPC01
41		P43				AN003/ CMPC23/ CMPC50
42		P42				AN002/ CMPC20
43		P41				AN001/ CMPC10
44		P40				AN000/ CMPC13/ CMPC00
45	AVCC0					
46	AVSS0					
47		P11	MTIOC3A/MTCLKC/ MTIOC3A#/MTCLKC#/ TMO3/POE9#/MTIOC9D/ GTIOC3B/GTETRGA/ GTIOC3B#/GTETRGC/ GTCPPO0/TOC3		IRQ1	
48		P10	MTIOC9B/MTCLKD/ MTIOC9B#/MTCLKD#/ TMRI3/POE12#/GTIOC3A/ GTETRGB/GTIOC3A#/ GTETRGD/GTIV/TIC3	CTS6#/RTS6#/SS6#	IRQ0	

2. CPU

The RXv3 CPU is based on the RXv3 instruction set architecture. Its instruction processing efficiency has been improved relative to that of the RXv2 CPU, so it delivers higher performance.
The RXv3 instruction set architecture (RXv3) provides upward compatibility from the RXv2 instruction set architecture (RXv2) and the RXv1 instruction set architecture (RXv1).

- Adoption of variable-length instruction format
  The CPU has short formats for frequently used instructions, facilitating the development of efficient programs that take up less memory.
- Powerful instruction set
  DSP instructions and floating-point operation instructions realize high-speed arithmetic processing.
- Versatile addressing modes
  The CPU has versatile addressing modes, with register-register operations, register-memory operations, and bitwise operations included. Data transfer between memory locations is also possible.

2.1 Features

- Minimum instruction execution rate: One clock cycle
- Address space: 4-Gbyte linear addresses
- Register set of the CPU
  General purpose: Sixteen 32-bit registers
  Control: Ten 32-bit registers
  Accumulator: Two 72-bit registers
- Variable-length instruction format (lengths from one to eight bytes)
- 113 instructions (products with 64 Kbytes of RAM), 111 instructions (products with 48 Kbytes of RAM)
  Standard provided instructions: 111
    Basic instructions: 77
    Single-precision floating point instructions: 11
    DSP instructions: 23
  Instructions for register bank save function: 2 (only supported by products with 64 Kbytes of RAM)
- Processor modes
  Supervisor mode and user mode
- Vector tables
  Exception vector table and interrupt vector table
- Memory protection unit
- Data arrangement
  Selectable as little endian or big endian

2.2 Register Set of the CPU

The CPU has sixteen general-purpose registers, ten control registers, and two accumulator used for DSP instructions.

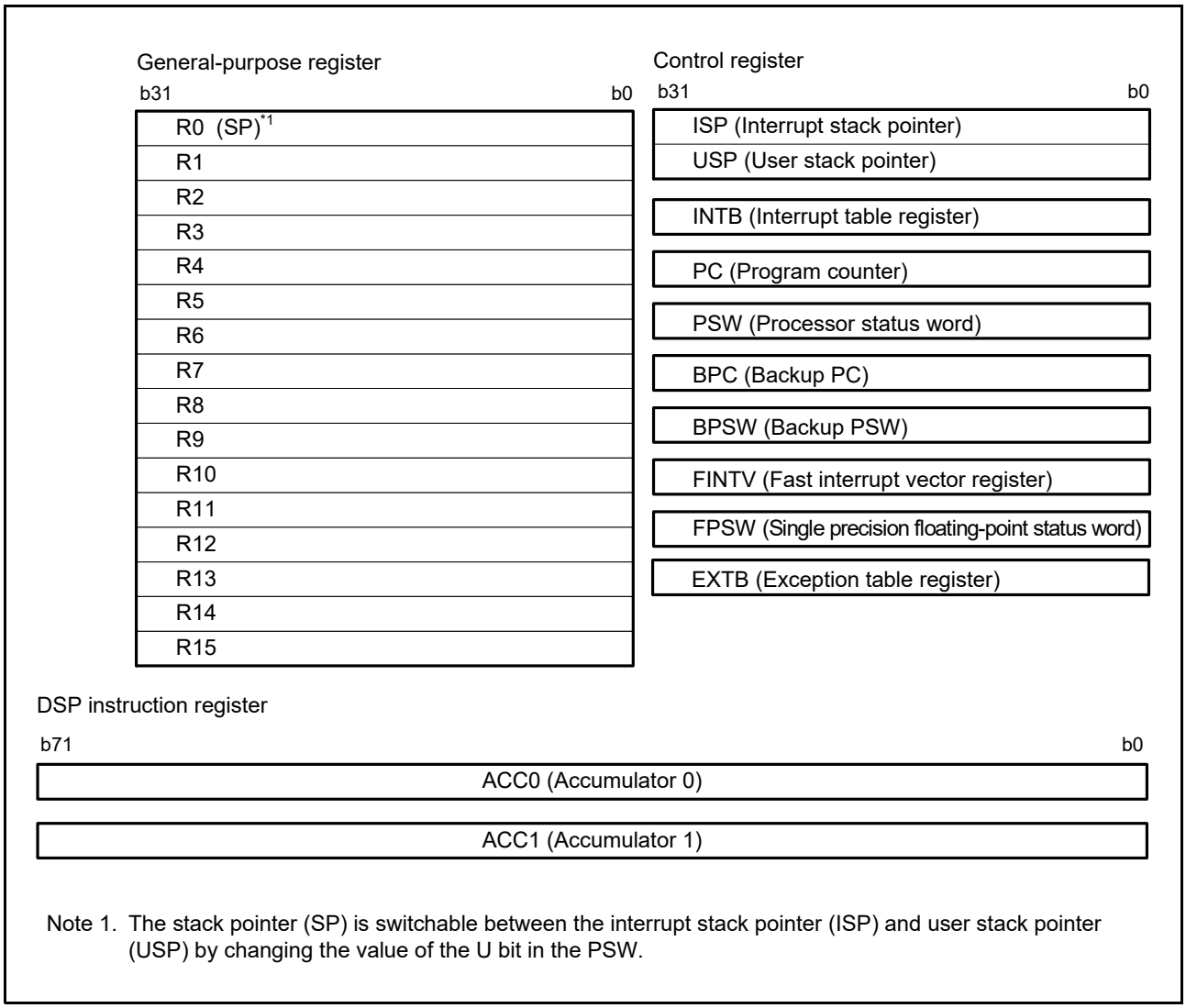


Figure 2.1 Register Set of the CPU

2.2.1 General-Purpose Registers (R0 to R15)

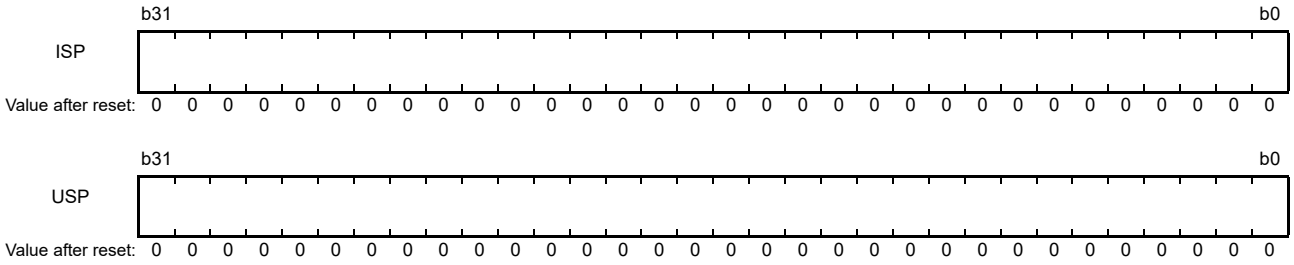
This CPU has sixteen 32-bit general-purpose registers (R0 to R15). R0 to R15 can be used as data registers or address registers.
R0, a general-purpose register, also functions as the stack pointer (SP).
The stack pointer is switched to operate as the interrupt stack pointer (ISP) or user stack pointer (USP) by the value of the stack pointer select bit (U) in the processor status word (PSW).

2.2.2 Control Registers

This CPU has the following ten control registers.

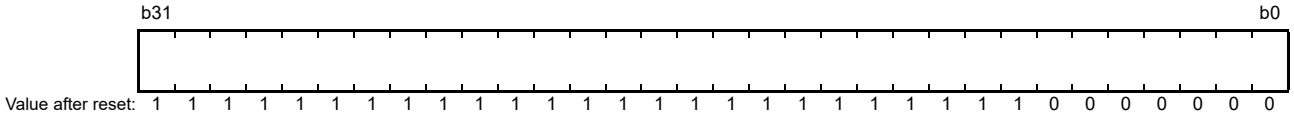
- Interrupt stack pointer (ISP)
- User stack pointer (USP)
- Exception table register (EXTB)
- Interrupt table register (INTB)
- Program counter (PC)
- Processor status word (PSW)
- Backup PC (BPC)
- Backup PSW (BPSW)
- Fast interrupt vector register (FINTV)
- Single-precision floating-point status word (FPSW)

2.2.2.1 Interrupt Stack Pointer (ISP)/User Stack Pointer (USP)



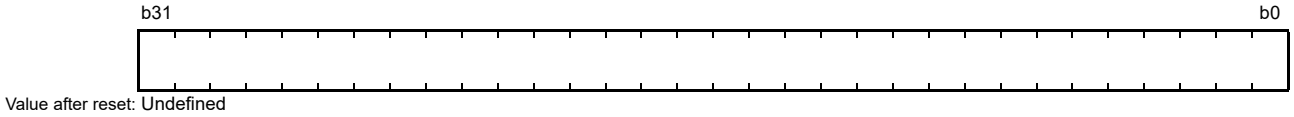
The stack pointer (SP) can be either of two types, the interrupt stack pointer (ISP) or the user stack pointer (USP). Whether the stack pointer operates as the ISP or USP depends on the value of the stack pointer select bit (U) in the processor status word (PSW).

2.2.2.2 Exception Table Register (EXTB)



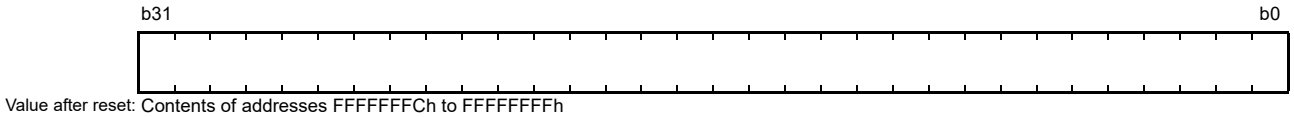
The exception table register (EXTB) specifies the address where the exception vector table starts.

2.2.2.3 Interrupt Table Register (INTB)



The interrupt table register (INTB) specifies the address where the interrupt vector table starts.

2.2.2.4 Program Counter (PC)



The program counter (PC) indicates the address of the instruction being executed.

2.2.2.5 Processor Status Word (PSW)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	IPL[3:0]				—	—	—	PM	—	—	U	I
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	O	S	Z	C
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	C	Carry Flag	0: No carry has occurred. 1: A carry has occurred.	R/W
b1	Z	Zero Flag	0: Result is non-zero. 1: Result is 0.	R/W
b2	S	Sign Flag	0: Result is a positive value or 0. 1: Result is a negative value.	R/W
b3	O	Overflow Flag	0: No overflow has occurred. 1: An overflow has occurred.	R/W
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	I*1	Interrupt Enable	0: Interrupt disabled. 1: Interrupt enabled.	R/W
b17	U*1	Stack Pointer Select	0: Interrupt stack pointer (ISP) is selected. 1: User stack pointer (USP) is selected.	R/W
b19, b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b20	PM*1, *2, *3	Processor Mode Select	0: Supervisor mode is selected. 1: User mode is selected.	R/W
b23 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b27 to b24	IPL[3:0]*1	Processor Interrupt Priority Level	b27 b24 0 0 0 0: Priority level 0 (lowest) 0 0 0 1: Priority level 1 0 0 1 0: Priority level 2 0 0 1 1: Priority level 3 0 1 0 0: Priority level 4 0 1 0 1: Priority level 5 0 1 1 0: Priority level 6 0 1 1 1: Priority level 7 1 0 0 0: Priority level 8 1 0 0 1: Priority level 9 1 0 1 0: Priority level 10 1 0 1 1: Priority level 11 1 1 0 0: Priority level 12 1 1 0 1: Priority level 13 1 1 1 0: Priority level 14 1 1 1 1: Priority level 15 (highest)	R/W
b31 to b28	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. In user mode, writing to the IPL[3:0], PM, U, and I bits by an MVTC or a POPC instruction is ignored. Writing to the IPL[3:0] bits by an MVTIPL instruction generates a privileged instruction exception.
Note 2. In supervisor mode, writing to the PM bit by an MVTC or a POPC instruction is ignored, but writing to the other bits is possible.
Note 3. Switching from supervisor mode to user mode requires execution of an RTE instruction after having set the PSW.PM bit saved on the stack to 1 or executing an RTFI instruction after having set the BPSW.PM bit to 1.

The processor status word (PSW) indicates the results of instruction execution or the state of the CPU.

C Flag (Carry Flag)

This flag retains the state of the bit after a carry, borrow, or shift-out has occurred.

Z Flag (Zero Flag)

This flag is set to 1 if the result of an operation is 0; otherwise its value is set to 0.

S Flag (Sign Flag)

This flag is set to 1 if the result of an operation is negative; otherwise its value is set to 0.

O Flag (Overflow Flag)

This flag is set to 1 if the result of an operation overflows; otherwise its value is set to 0.

I Bit (Interrupt Enable)

This bit enables interrupt requests. When a WAIT instruction is executed, the value of this bit becomes 1. It becomes 0 when an exception is accepted.

U Bit (Stack Pointer Select)

This bit specifies the stack pointer as either the ISP or USP. When an exception request is accepted, this bit is set to 0. When the processor mode is switched from supervisor mode to user mode, this bit is set to 1.

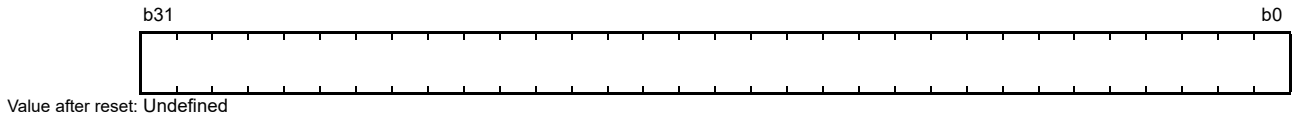
PM Bit (Processor Mode Select)

This bit specifies the processor mode. When an exception is accepted, the value of this bit becomes 0.

IPL[3:0] Bits (Processor Interrupt Priority Level)

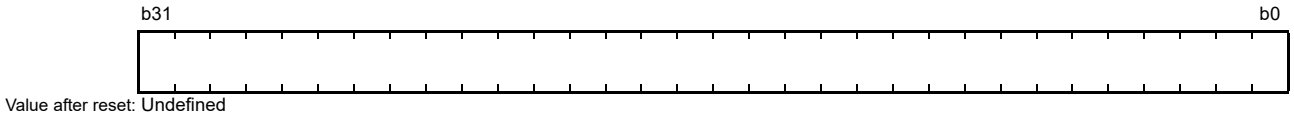
The IPL[3:0] bits specify the processor interrupt priority level as one of sixteen levels from zero to fifteen, wherein priority level zero is the lowest and priority level fifteen the highest. When the priority level of a requested interrupt is higher than the processor interrupt priority level, the interrupt is enabled. Setting the IPL[3:0] bits to level fifteen (Fh) disables all interrupt requests. The IPL[3:0] bits are set to level fifteen (Fh) when a non-maskable interrupt is generated. When interrupts in general are generated, the bits are set to the priority levels of accepted interrupts.

2.2.2.6 Backup PC (BPC)



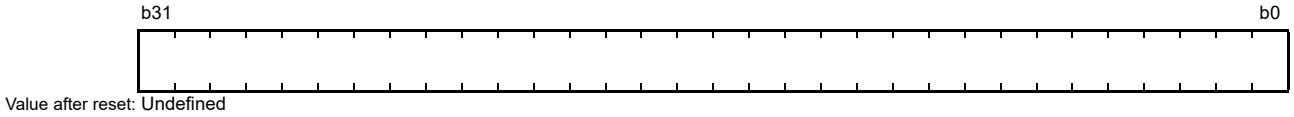
The backup PC (BPC) is provided to speed up response to interrupts.
After a fast interrupt has been generated, the contents of the program counter (PC) are saved in the BPC register.

2.2.2.7 Backup PSW (BPSW)



The backup PSW (BPSW) is provided to speed up response to interrupts.
After a fast interrupt has been generated, the contents of the processor status word (PSW) are saved in the BPSW. The allocation of bits in the BPSW corresponds to that in the PSW.

2.2.2.8 Fast Interrupt Vector Register (FINTV)



The fast interrupt vector register (FINTV) is provided to speed up response to interrupts.
The FINTV register specifies a branch destination address when a fast interrupt has been generated.

2.2.2.9 Single-Precision Floating-Point Status Word (FPSW)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	FS	FX	FU	FZ	FO	FV	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	EX	EU	EZ	EO	EV	—	DN	CE	CX	CU	CZ	CO	CV	RM[1:0]	
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	RM[1:0]	Single-Precision Floating-Point Rounding-Mode Setting	b1 b0 0 0: Rounding towards the nearest value 0 1: Rounding towards 0 1 0: Rounding towards +∞ 1 1: Rounding towards –∞	R/W
b2	CV	Invalid Operation Cause Flag	0: No invalid operation has been encountered. 1: Invalid operation has been encountered.	R/(W) *1
b3	CO	Overflow Cause Flag	0: No overflow has occurred. 1: Overflow has occurred.	R/(W) *1
b4	CZ	Division-by-Zero Cause Flag	0: No division-by-zero has occurred. 1: Division-by-zero has occurred.	R/(W) *1
b5	CU	Underflow Cause Flag	0: No underflow has occurred. 1: Underflow has occurred.	R/(W) *1
b6	CX	Inexact Cause Flag	0: No inexact exception has been generated. 1: Inexact exception has been generated.	R/(W) *1
b7	CE	Unimplemented Processing Cause Flag	0: No unimplemented processing has been encountered. 1: Unimplemented process has been encountered.	R/(W) *1
b8	DN	0 Flush Bit of Denormalized Number	0: A denormalized number is handled as a denormalized number. 1: A denormalized number is handled as 0.*2	R/W
b9	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10	EV	Invalid Operation Exception Enable	0: Invalid operation exception is masked. 1: Invalid operation exception is enabled.	R/W
b11	EO	Overflow Exception Enable	0: Overflow exception is masked. 1: Overflow exception is enabled.	R/W
b12	EZ	Division-by-Zero Exception Enable	0: Division-by-zero exception is masked. 1: Division-by-zero exception is enabled.	R/W
b13	EU	Underflow Exception Enable	0: Underflow exception is masked. 1: Underflow exception is enabled.	R/W
b14	EX	Inexact Exception Enable	0: Inexact exception is masked. 1: Inexact exception is enabled.	R/W
b25 to b15	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b26	FV*3	Invalid Operation Flag	0: No invalid operation has been encountered. 1: Invalid operation has been encountered.*8	R/W
b27	FO*4	Overflow Flag	0: No overflow has occurred. 1: Overflow has occurred.*8	R/W
b28	FZ*5	Division-by-Zero Flag	0: No division-by-zero has occurred. 1: Division-by-zero has occurred.*8	R/W
b29	FU*6	Underflow Flag	0: No underflow has occurred. 1: Underflow has occurred.*8	R/W
b30	FX*7	Inexact Flag	0: No inexact exception has been generated. 1: Inexact exception has been generated.*8	R/W
b31	FS	Single-Precision Floating-Point Error Summary Flag	This bit reflects the logical OR of the FU, FZ, FO, and FV flags.	R

Note 1. Writing 0 to the bit clears it. Writing 1 to the bit does not affect its value.
Note 2. Positive denormalized numbers are treated as +0, negative denormalized numbers as –0.
Note 3. When the EV bit is set to 0, the FV flag is enabled.
Note 4. When the EO bit is set to 0, the FO flag is enabled.
Note 5. When the EZ bit is set to 0, the FZ flag is enabled.
Note 6. When the EU bit is set to 0, the FU flag is enabled.
Note 7. When the EX bit is set to 0, the FX flag is enabled.
Note 8. Once the bit has been set to 1, this value is retained until it is set to 0 by software.

The single-precision floating-point status word (FPSW) indicates the results of single-precision floating-point arithmetic operations.
When the corresponding exception handling enable bits (Ej) are set to enable processing of the exceptions (Ej = 1), the Cj flags can be used by the exception handling routine to identify the source of that exception. If handling of an exception is masked (Ej = 0), the Fj flag can be used to check for the generation of the exception at the end of a sequence of processing. The Fj flags operate in an accumulative fashion (j = X, U, Z, O, or V).

RM[1:0] Bits (Single-Precision Floating-Point Rounding-Mode Setting)

These bits specify the single-precision floating-point rounding-mode.

Explanation of Single-Precision Floating-Point Rounding Modes

- Rounding towards the nearest value (the default behavior): An inexact result is rounded to the available value that is closest to the result of a hypothetical calculation with infinite precision. If two available values are equally close, rounding is to the even alternative.
- Rounding towards 0: An inexact result is rounded to the smallest available absolute value, i.e. in the direction of zero (simple truncation).
- Rounding towards +∞: An inexact result is rounded to the nearest available value in the direction of positive infinity.
- Rounding towards –∞: An inexact result is rounded to the nearest available value in the direction of negative infinity.

(1) Rounding to the nearest value is specified as the default mode and returns the most accurate value.
(2) Modes such as rounding towards 0, rounding towards +∞, and rounding towards –∞ are used to ensure precision when interval arithmetic is employed.

CV Flag (Invalid Operation Cause Flag), CO Flag (Overflow Cause Flag), CZ Flag (Division-by-Zero Cause Flag), CU Flag (Underflow Cause Flag), CX Flag (Inexact Cause Flag), and CE Flag (Unimplemented Processing Cause Flag)

Single-precision floating-point exceptions include the five specified in the IEEE754 standard, namely overflow, underflow, inexact, division-by-zero, and invalid operation. For a further single-precision floating-point exception that is generated upon detection of unimplemented processing, the corresponding flag (CE) is set to 1.

- If an exception or processing that is not implemented is not encountered in the execution of a single-precision floating-point arithmetic instruction, the corresponding flags become 0.
- When 0 is written to the bit by the MVTC and POPC instructions, the bit is set to 0; the bit retains the previous value when 1 is written by the instruction.

DN Bit (0 Flush Bit of Denormalized Number)

When this bit is set to 0, a denormalized number is handled as a denormalized number. When this bit is set to 1, a denormalized number is handled as 0.

EV Bit (Invalid Operation Exception Enable), EO Bit (Overflow Exception Enable), EZ Bit (Division-by-Zero Exception Enable), EU Bit (Underflow Exception Enable), and EX Bit (Inexact Exception Enable)

When any of five single-precision floating-point exceptions specified in the IEEE754 standard is generated by the single-precision floating-point operation instruction, the bit decides whether the CPU will start handling the exception. When the bit is set to 0, the exception handling is masked; when the bit is set to 1, the exception handling is enabled.

FV Flag (Invalid Operation Flag), FO Flag (Overflow Flag), FZ Flag (Division-by-Zero Flag), FU Flag (Underflow Flag), and FX Flag (Inexact Flag)

While the exception handling enable bit (Ej) is 0 (exception handling is masked), if any of five single-precision floating-point exceptions specified in the IEEE754 standard is generated, the corresponding bit is set to 1.

- When Ej is 1 (exception handling is enabled), the value of the flag remains.
- When the corresponding flag is set to 1, it remains 1 until it is set to 0 by software (accumulation flag).

FS Flag (Single-Precision Floating-Point Error Summary Flag)

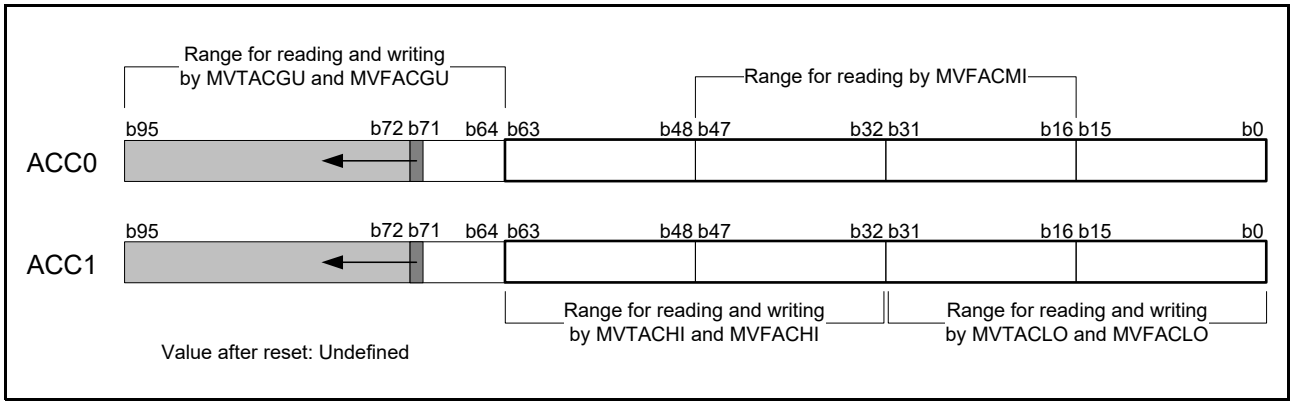
This bit reflects the logical OR of the FU, FZ, FO, and FV flags.

2.2.3 Accumulator

The accumulator (ACC0 or ACC1) is a 72-bit register used for DSP instructions. The accumulator is handled as a 96-bit register for reading and writing. At this time, when bits 95 to 72 of the accumulator are read, the value where the value of bit 71 is sign extended is read. Writing to bits 95 to 72 of the accumulator is ignored. ACC0 is also used for the multiply and multiply-and-accumulate instructions; EMUL, EMULU, FMUL, MUL, and RMPA, in which case the prior value in ACC0 is modified by execution of the instruction.

Use the MVTACGU, MVTACHI, and MVTACLO instructions for writing to the accumulator. The MVTACGU, MVTACHI, and MVTACLO instructions write data to bits 95 to 64, the higher-order 32 bits (bits 63 to 32), and the lower-order 32 bits (bits 31 to 0), respectively.

Use the MVFACGU, MVFACHI, MVFACMI, and MVFACLO instructions for reading data from the accumulator. The MVFACGU, MVFACHI, MVFACMI, and MVFACLO instructions read data from the guard bits (bits 95 to 64), higher-order 32 bits (bits 63 to 32), the middle 32 bits (bits 47 to 16), and the lower-order 32 bits (bits 31 to 0), respectively.



Note: The value of bit 71 is sign extended for bits 95 to 72 and the extended value is always read. Writing to this area is ignored.

2.3 Processor Mode

The CPU supports two processor modes, supervisor and user. These processor modes and the memory protection function enable the realization of a hierarchical CPU resource protection and memory protection mechanism. Each processor mode imposes a level on rights of access to memory and the instructions that can be executed. Supervisor mode carries greater rights than user mode. The initial state after a reset is supervisor mode.

2.3.1 Supervisor Mode

In supervisor mode, all CPU resources are accessible and all instructions are available. However, writing to the processor mode select bit (PM) in the processor status word (PSW) by executing an MVTC or a POPC instruction will be ignored. For details on how to write to the PM bit, refer to section 2.2.2.5, Processor Status Word (PSW).

2.3.2 User Mode

In user mode, write access to the CPU resources listed below is restricted. The restriction applies to any instruction capable of write access.

- Some bits (bits IPL[3:0], PM, U, and I) in the processor status word (PSW)
- Interrupt stack pointer (ISP)
- Exception table register (EXTB)
- Interrupt table register (INTB)
- Backup PSW (BPSW)
- Backup PC (BPC)
- Fast interrupt vector register (FINTV)

2.3.3 Privileged Instruction

Privileged instructions can only be executed in supervisor mode. Executing a privileged instruction in user mode produces a privileged instruction exception. Privileged instructions include the RTFI, MVTIPL, RTE, WAIT, SAVE, and RSTR instructions. Note that, however, products with 48 Kbytes of RAM do not have the SAVE or RSTR instruction.

2.3.4 Switching Between Processor Modes

Manipulating the processor mode select bit (PM) in the processor status word (PSW) switches the processor mode. However, rewriting to the PM bit by executing an MVTC or a POPC instruction is prohibited. Switch the processor mode by following the procedures described below.

(1) Switching from user mode to supervisor mode

After an exception has been generated, the PSW.PM bit is set to 0 and the CPU switches to supervisor mode. The hardware pre-processing is executed in supervisor mode. The state of the processor mode before the exception was generated is retained in the copy of PSW.PM bit is saved on the stack.

(2) Switching from supervisor mode to user mode

Executing an RTE instruction when the value of the copy of the PSW.PM bit that has been preserved on the stack is 1 or an RTFI instruction when the value of the copy of the PSW.PM bit that has been preserved in the backup PSW (BPSW) is 1 causes a transition to user mode. In the transition to user mode, the value of the stack pointer designation bit (the U bit in the PSW) becomes 1.

2.4 Data Types

The CPU can handle four types of data: integer, single-precision floating-point number, bit, and string.
For details, refer to RX Family RXv3 Instruction Set Architecture User's Manual: Software.

2.4.1 Integer

An integer can be signed or unsigned. For signed integers, negative values are represented by two's complements.

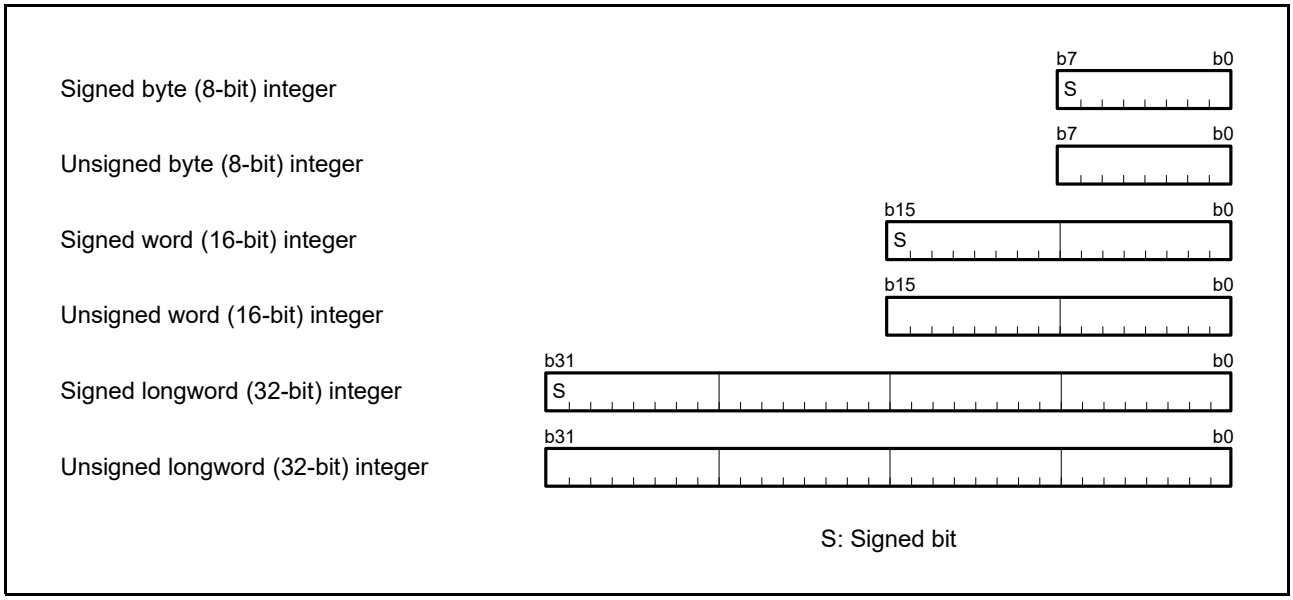


Figure 2.2 Integer

2.4.2 Single-Precision Floating-Point Numbers

The single-precision floating-point number is compliant with that specified in the IEEE754 standard; operands of this type can be used in eleven single-precision floating-point operation instructions: FADD, FCMP, FDIV, FMUL, FSQRT, FSUB, FTOI, FTOU, ITOF, ROUND, and UTOF.

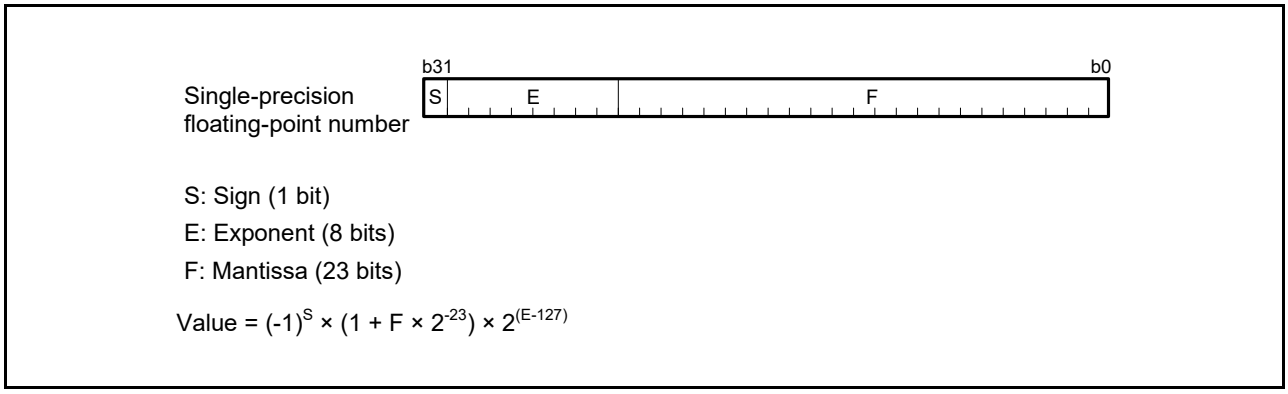


Figure 2.3 Single-Precision Floating-Point Number

The single-precision floating-point number can represent the values listed below.

- 0 < E < 255 (normal numbers)
- E = 0 and F = 0 (signed zero)
- E = 0 and F > 0 (denormalized numbers)*1
- E = 255 and F = 0 (infinity)
- E = 255 and F > 0 (NaN: Not-a-Number)

Note 1. The number is treated as 0 when the FPSW.DN bit is 1. When the DN bit is 0, an unimplemented processing exception is generated.

2.4.3 Bitwise Operations

Five bit-manipulation instructions are provided for bitwise operations: BCLR, BM*Cnd*, BNOT, BSET, and BTST.
A bit in a register is specified as the destination register and a bit number in the range from 31 to 0.
A bit in memory is specified as the destination address and a bit number from 7 to 0. The addressing modes available to specify addresses are register indirect and register relative.

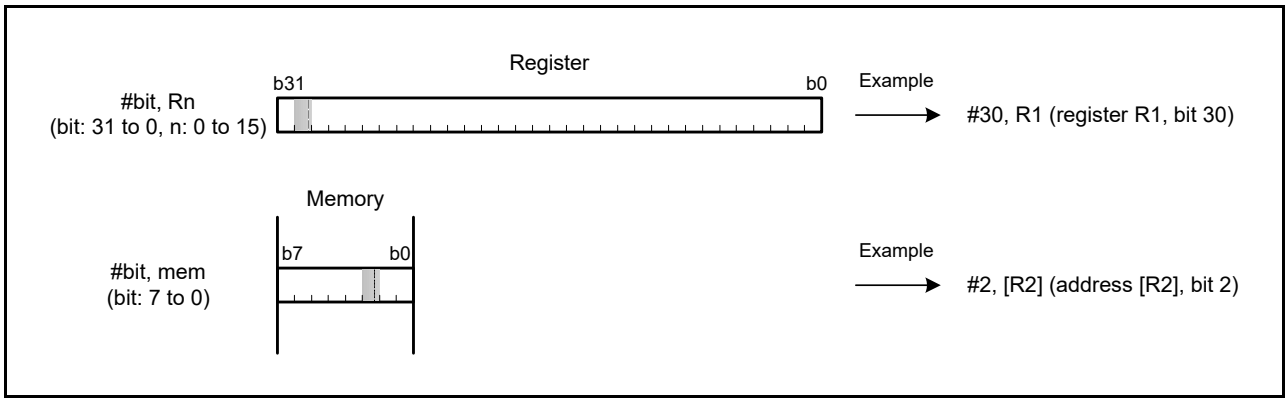


Figure 2.4 Bit

2.4.4 Strings

The string data type consists of an arbitrary number of consecutive byte (8-bit), word (16-bit), or longword (32-bit) units. Seven string manipulation instructions are provided for use with strings: SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, and SWHILE.

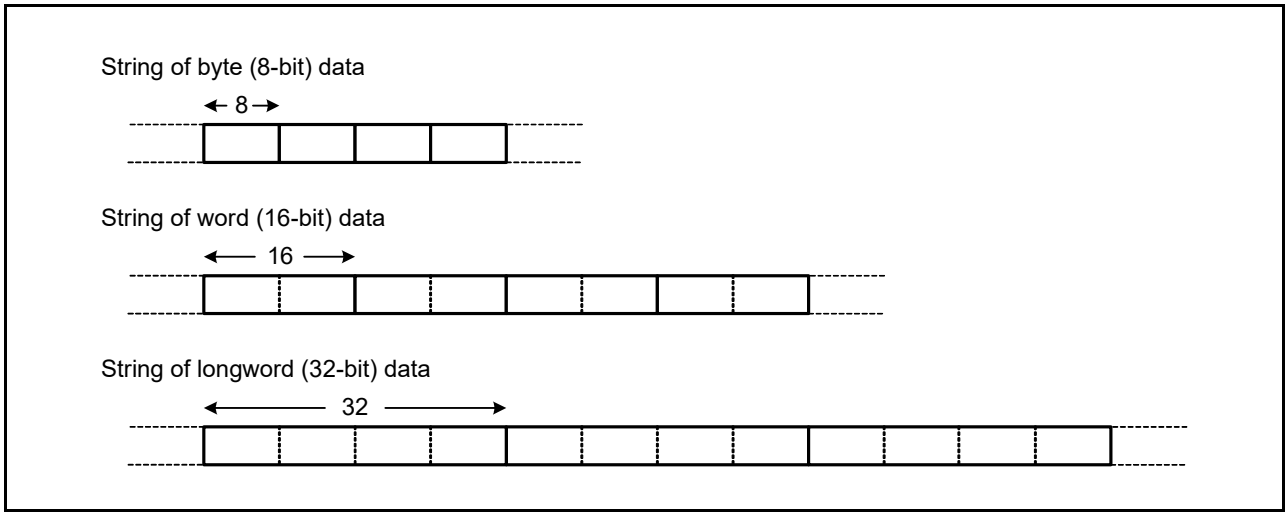


Figure 2.5 String

2.5 Endian

For the CPU, instructions are little endian, but the treatment of data is selectable as little or big endian.

2.5.1 Switching the Endian

As arrangements of bytes, this MCU supports both big endian, where the higher-order byte (MSB) is at location 0, and little endian, where the lower-order byte (LSB) is at location 0.
For details on the endian setting, see section 3, Operating Modes.
Operations for access differ according to the endian setting and, depending on the instruction, whether 8-, 16- or 32-bit access has been selected. Operations for access in the various possible cases are described in Table 2.1 to Table 2.12.
In the tables,

LL indicates bits D7 to D0 of the general-purpose register,
LH indicates bits D15 to D8 of the general-purpose register,
HL indicates bits D23 to D16 of the general-purpose register, and
HH indicates bits D31 to D24 of the general-purpose register.

	D31 to D24	D23 to D16	D15 to D8	D7 to D0
General purpose register: Rm	HH	HL	LH	LL

Table 2.1 32-Bit Read Operations when Little Endian has been Selected

Operation / Address of src	Reading a 32-bit unit from address 0	Reading a 32-bit unit from address 1	Reading a 32-bit unit from address 2	Reading a 32-bit unit from address 3	Reading a 32-bit unit from address 4
Address 0	Transfer to LL	—	—	—	—
Address 1	Transfer to LH	Transfer to LL	—	—	—
Address 2	Transfer to HL	Transfer to LH	Transfer to LL	—	—
Address 3	Transfer to HH	Transfer to HL	Transfer to LH	Transfer to LL	—
Address 4	—	Transfer to HH	Transfer to HL	Transfer to LH	Transfer to LL
Address 5	—	—	Transfer to HH	Transfer to HL	Transfer to LH
Address 6	—	—	—	Transfer to HH	Transfer to HL
Address 7	—	—	—	—	Transfer to HH

Table 2.2 32-Bit Read Operations when Big Endian has been Selected

Operation / Address of src	Reading a 32-bit unit from address 0	Reading a 32-bit unit from address 1	Reading a 32-bit unit from address 2	Reading a 32-bit unit from address 3	Reading a 32-bit unit from address 4
Address 0	Transfer to HH	—	—	—	—
Address 1	Transfer to HL	Transfer to HH	—	—	—
Address 2	Transfer to LH	Transfer to HL	Transfer to HH	—	—
Address 3	Transfer to LL	Transfer to LH	Transfer to HL	Transfer to HH	—
Address 4	—	Transfer to LL	Transfer to LH	Transfer to HL	Transfer to HH
Address 5	—	—	Transfer to LL	Transfer to LH	Transfer to HL
Address 6	—	—	—	Transfer to LL	Transfer to LH
Address 7	—	—	—	—	Transfer to LL

Table 2.3 32-Bit Write Operations when Little Endian has been Selected

Operation / Address of dest	Writing a 32-bit unit to address 0	Writing a 32-bit unit to address 1	Writing a 32-bit unit to address 2	Writing a 32-bit unit to address 3	Writing a 32-bit unit to address 4
Address 0	Transfer from LL	—	—	—	—
Address 1	Transfer from LH	Transfer from LL	—	—	—
Address 2	Transfer from HL	Transfer from LH	Transfer from LL	—	—
Address 3	Transfer from HH	Transfer from HL	Transfer from LH	Transfer from LL	—
Address 4	—	Transfer from HH	Transfer from HL	Transfer from LH	Transfer from LL
Address 5	—	—	Transfer from HH	Transfer from HL	Transfer from LH
Address 6	—	—	—	Transfer from HH	Transfer from HL
Address 7	—	—	—	—	Transfer from HH

Table 2.4 32-Bit Write Operations when Big Endian has been Selected

Operation / Address of dest	Writing a 32-bit unit to address 0	Writing a 32-bit unit to address 1	Writing a 32-bit unit to address 2	Writing a 32-bit unit to address 3	Writing a 32-bit unit to address 4
Address 0	Transfer from HH	—	—	—	—
Address 1	Transfer from HL	Transfer from HH	—	—	—
Address 2	Transfer from LH	Transfer from HL	Transfer from HH	—	—
Address 3	Transfer from LL	Transfer from LH	Transfer from HL	Transfer from HH	—
Address 4	—	Transfer from LL	Transfer from LH	Transfer from HL	Transfer from HH
Address 5	—	—	Transfer from LL	Transfer from LH	Transfer from HL
Address 6	—	—	—	Transfer from LL	Transfer from LH
Address 7	—	—	—	—	Transfer from LL

Table 2.5 16-Bit Read Operations when Little Endian has been Selected

Operation / Address of src	Reading a 16-bit unit from address 0	Reading a 16-bit unit from address 1	Reading a 16-bit unit from address 2	Reading a 16-bit unit from address 3	Reading a 16-bit unit from address 4	Reading a 16-bit unit from address 5	Reading a 16-bit unit from address 6
Address 0	Transfer to LL	—	—	—	—	—	—
Address 1	Transfer to LH	Transfer to LL	—	—	—	—	—
Address 2	—	Transfer to LH	Transfer to LL	—	—	—	—
Address 3	—	—	Transfer to LH	Transfer to LL	—	—	—
Address 4	—	—	—	Transfer to LH	Transfer to LL	—	—
Address 5	—	—	—	—	Transfer to LH	Transfer to LL	—
Address 6	—	—	—	—	—	Transfer to LH	Transfer to LL
Address 7	—	—	—	—	—	—	Transfer to LH

Table 2.6 16-Bit Read Operations when Big Endian has been Selected

Operation / Address of src	Reading a 16-bit unit from address 0	Reading a 16-bit unit from address 1	Reading a 16-bit unit from address 2	Reading a 16-bit unit from address 3	Reading a 16-bit unit from address 4	Reading a 16-bit unit from address 5	Reading a 16-bit unit from address 6
Address 0	Transfer to LH	—	—	—	—	—	—
Address 1	Transfer to LL	Transfer to LH	—	—	—	—	—
Address 2	—	Transfer to LL	Transfer to LH	—	—	—	—
Address 3	—	—	Transfer to LL	Transfer to LH	—	—	—
Address 4	—	—	—	Transfer to LL	Transfer to LH	—	—
Address 5	—	—	—	—	Transfer to LL	Transfer to LH	—
Address 6	—	—	—	—	—	Transfer to LL	Transfer to LH
Address 7	—	—	—	—	—	—	Transfer to LL

Table 2.7 16-Bit Write Operations when Little Endian has been Selected

Operation / Address of dest	Writing a 16-bit unit to address 0	Writing a 16-bit unit to address 1	Writing a 16-bit unit to address 2	Writing a 16-bit unit to address 3	Writing a 16-bit unit to address 4	Writing a 16-bit unit to address 5	Writing a 16-bit unit to address 6
Address 0	Transfer from LL	—	—	—	—	—	—
Address 1	Transfer from LH	Transfer from LL	—	—	—	—	—
Address 2	—	Transfer from LH	Transfer from LL	—	—	—	—
Address 3	—	—	Transfer from LH	Transfer from LL	—	—	—
Address 4	—	—	—	Transfer from LH	Transfer from LL	—	—
Address 5	—	—	—	—	Transfer from LH	Transfer from LL	—
Address 6	—	—	—	—	—	Transfer from LH	Transfer from LL
Address 7	—	—	—	—	—	—	Transfer from LH

Table 2.8 16-Bit Write Operations when Big Endian has been Selected

Operation / Address of dest	Writing a 16-bit unit to address 0	Writing a 16-bit unit to address 1	Writing a 16-bit unit to address 2	Writing a 16-bit unit to address 3	Writing a 16-bit unit to address 4	Writing a 16-bit unit to address 5	Writing a 16-bit unit to address 6
Address 0	Transfer from LH	—	—	—	—	—	—
Address 1	Transfer from LL	Transfer from LH	—	—	—	—	—
Address 2	—	Transfer from LL	Transfer from LH	—	—	—	—
Address 3	—	—	Transfer from LL	Transfer from LH	—	—	—
Address 4	—	—	—	Transfer from LL	Transfer from LH	—	—
Address 5	—	—	—	—	Transfer from LL	Transfer from LH	—
Address 6	—	—	—	—	—	Transfer from LL	Transfer from LH
Address 7	—	—	—	—	—	—	Transfer from LL

Table 2.9 8-Bit Read Operations when Little Endian has been Selected

Operation / Address of src	Reading an 8-bit unit from address 0	Reading an 8-bit unit from address 1	Reading an 8-bit unit from address 2	Reading an 8-bit unit from address 3
Address 0	Transfer to LL	—	—	—
Address 1	—	Transfer to LL	—	—
Address 2	—	—	Transfer to LL	—
Address 3	—	—	—	Transfer to LL

Table 2.10 8-Bit Read Operations when Big Endian has been Selected

Operation / Address of src	Reading an 8-bit unit from address 0	Reading an 8-bit unit from address 1	Reading an 8-bit unit from address 2	Reading an 8-bit unit from address 3
Address 0	Transfer to LL	—	—	—
Address 1	—	Transfer to LL	—	—
Address 2	—	—	Transfer to LL	—
Address 3	—	—	—	Transfer to LL

Table 2.11 8-Bit Write Operations when Little Endian has been Selected

Operation / Address of dest	Writing an 8-bit unit to address 0	Writing an 8-bit unit to address 1	Writing an 8-bit unit to address 2	Writing an 8-bit unit to address 3
Address 0	Transfer from LL	—	—	—
Address 1	—	Transfer from LL	—	—
Address 2	—	—	Transfer from LL	—
Address 3	—	—	—	Transfer from LL

Table 2.12 8-Bit Write Operations when Big Endian has been Selected

Operation / Address of dest	Writing an 8-bit unit to address 0	Writing an 8-bit unit to address 1	Writing an 8-bit unit to address 2	Writing an 8-bit unit to address 3
Address 0	Transfer from LL	—	—	—
Address 1	—	Transfer from LL	—	—
Address 2	—	—	Transfer from LL	—
Address 3	—	—	—	Transfer from LL

2.5.2 Access to I/O Registers

The addresses of I/O registers are fixed, and this is regardless of whether the setting is for little endian or big endian. Accordingly, changes to the endian do not affect access to I/O registers. For the arrangements of I/O registers, refer to the descriptions of registers in the relevant sections.

2.5.3 Notes on Access to I/O Registers

Ensure that access to I/O registers is in accord with the following rules.

- With I/O registers for which a bus width of eight bits is indicated, use instructions having operands of the same width (eight bits). That is, access these registers by using instructions with .B as the size specifier (.size), or with .B or .UB as the size-extension specifier (.memex).
- With I/O registers for which a bus width of 16 bits is indicated, use instructions having operands of the same width (16 bits). That is, access these registers by using instructions with .W as the size specifier (.size), or with .W or .UW as the size-extension specifier (.memex).
- With I/O registers for which a bus width of 32 bits is indicated, use instructions having operands of the same width (32 bits). That is, access these registers by using instructions with .L as the size specifier (.size), or with .L size-extension specifier (.memex).

2.5.4 Data Arrangement

2.5.4.1 Data Arrangement in Registers

Figure 2.6 shows the relation between the sizes of registers and bit numbers.

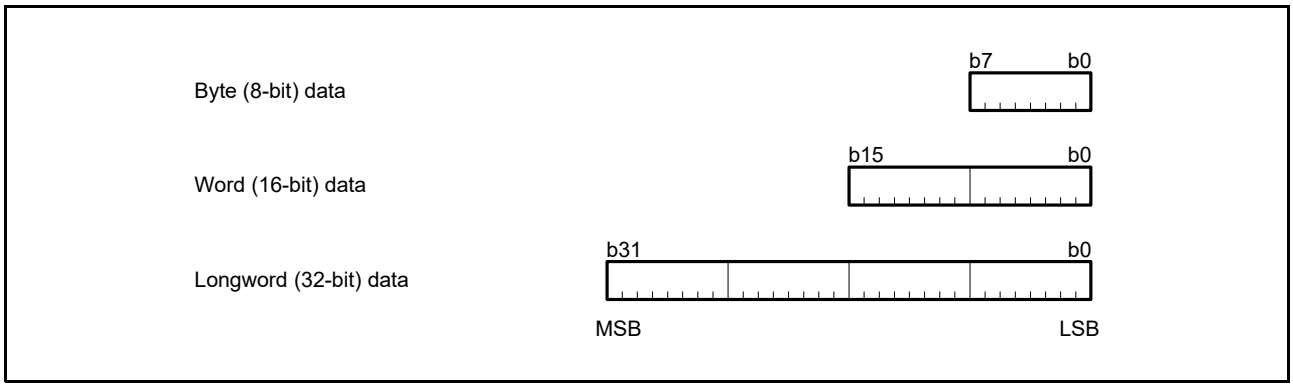


Figure 2.6 Data Arrangement in Registers

2.5.4.2 Data Arrangement in Memory

Data in memory have three sizes: byte (8-bit), word (16-bit), and longword (32-bit). The data arrangement is selectable as little endian or big endian. Figure 2.7 shows the arrangement of data in memory.

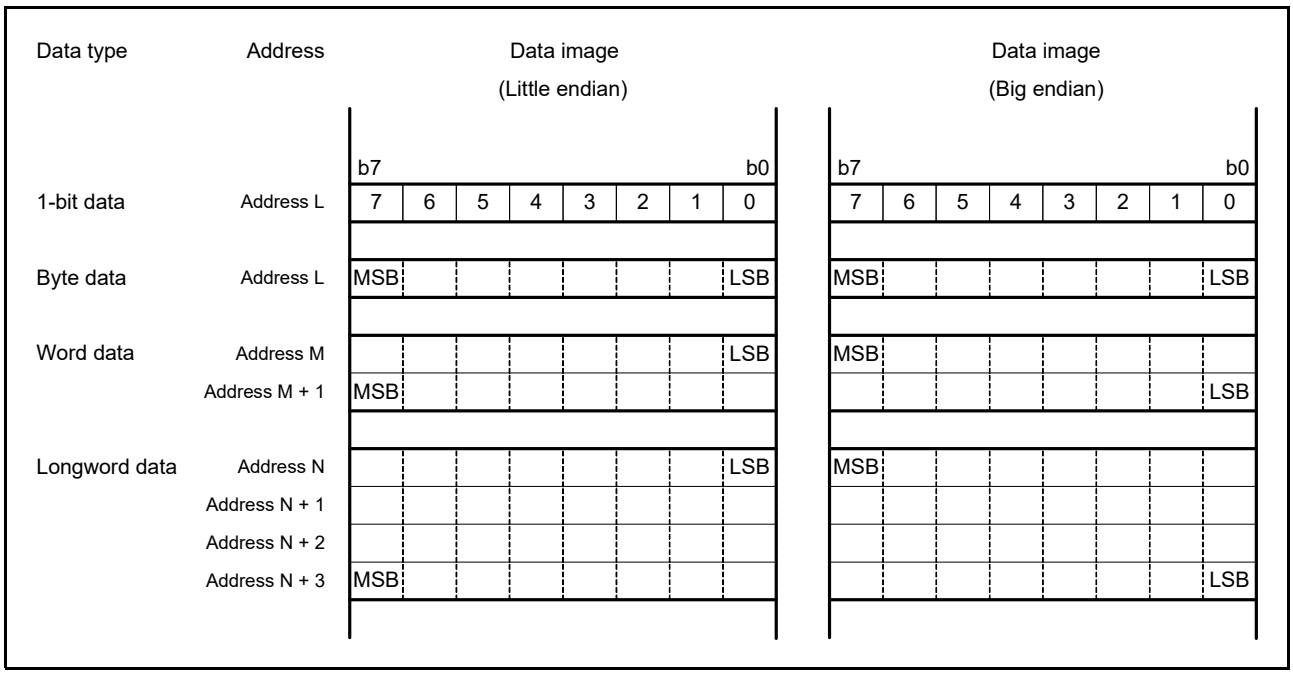


Figure 2.7 Data Arrangement in Memory

2.5.5 Notes on the Allocation of Instruction Codes

The allocation of instruction codes to an external space where the endian differs from that of the chip is prohibited. If the instruction codes are allocated to the external space, they must be allocated to areas where the endian setting is the same as that for the chip.

2.6 Vector Table

There are two types of vector table: exception and interrupt. Each vector in the vector table consists of four bytes and specifies the address where the corresponding exception handling routine starts.

2.6.1 Exception Vector Table

In the exception vector table, the individual vectors for the privileged instruction exception, access exception, undefined instruction exception, single-precision floating-point exception, and non-maskable interrupt are allocated to the 124-byte area where the value indicated by the exception table register (EXTB) is used as the starting address (ExtBase). The reset vector is always allocated to FFFFFFFCh, regardless of the value of the exception vector table.
Figure 2.8 shows the exception vector table.

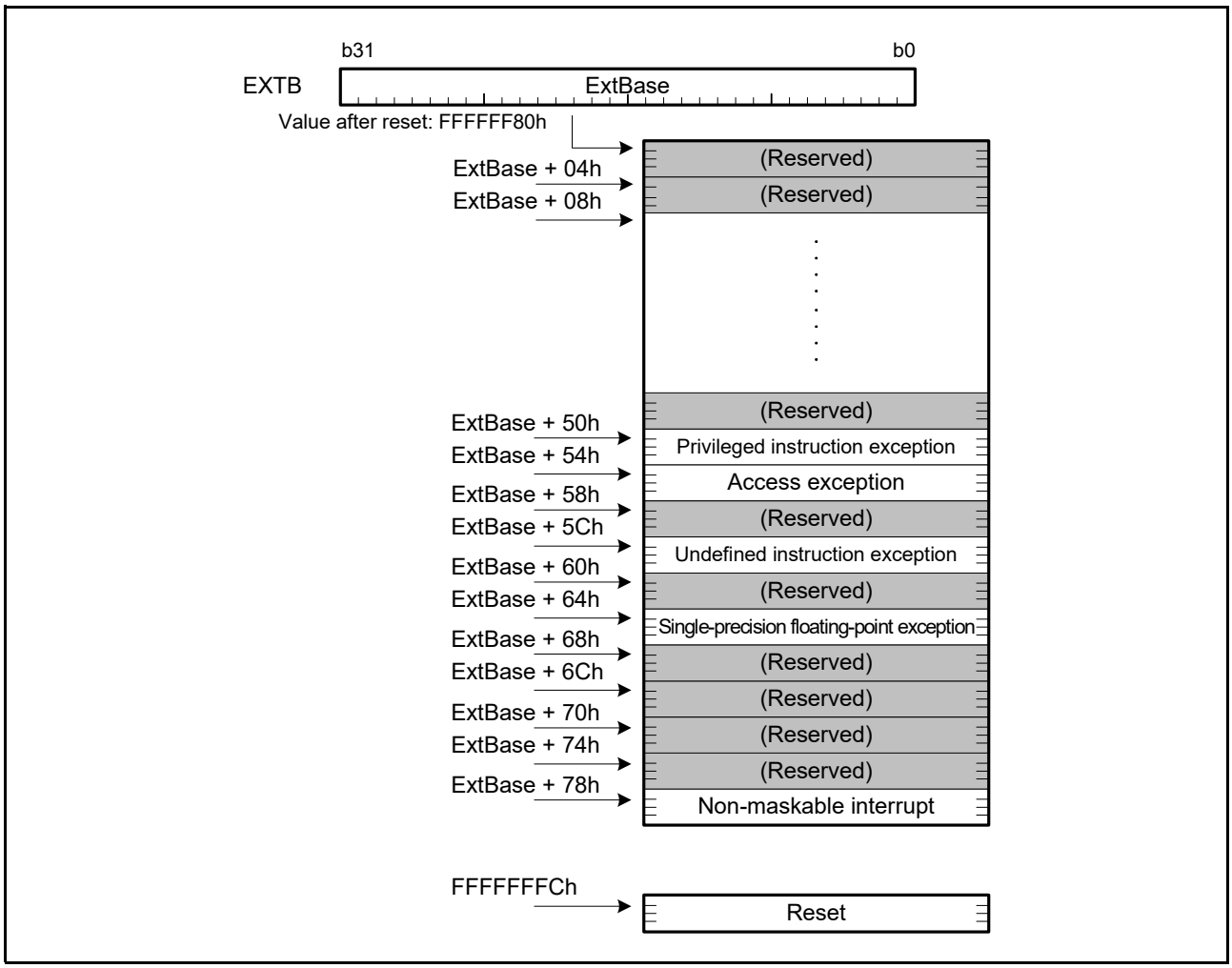


Figure 2.8 Exception Vector Table

2.6.2 Interrupt Vector Table

The address where the interrupt vector table is placed can be adjusted. The table is a 1,024-byte region that contains all vectors for unconditional traps and interrupts and starts at the address (IntBase) specified in the interrupt table register (INTB). Figure 2.9 shows the interrupt vector table.

Each vector in the interrupt vector table has a vector number from 0 to 255. Each of the INT instructions, which act as the sources of unconditional traps, is allocated to the vector that has the same number as is specified as the operand of the instruction itself (from 0 to 255). The BRK instruction is allocated to the vector with number 0. Furthermore, vector numbers (from 0 to 255) are allocated to interrupt requests in a fixed way for each product. For more on interrupt vector numbers, see section 14.3.1, Interrupt Vector Table.

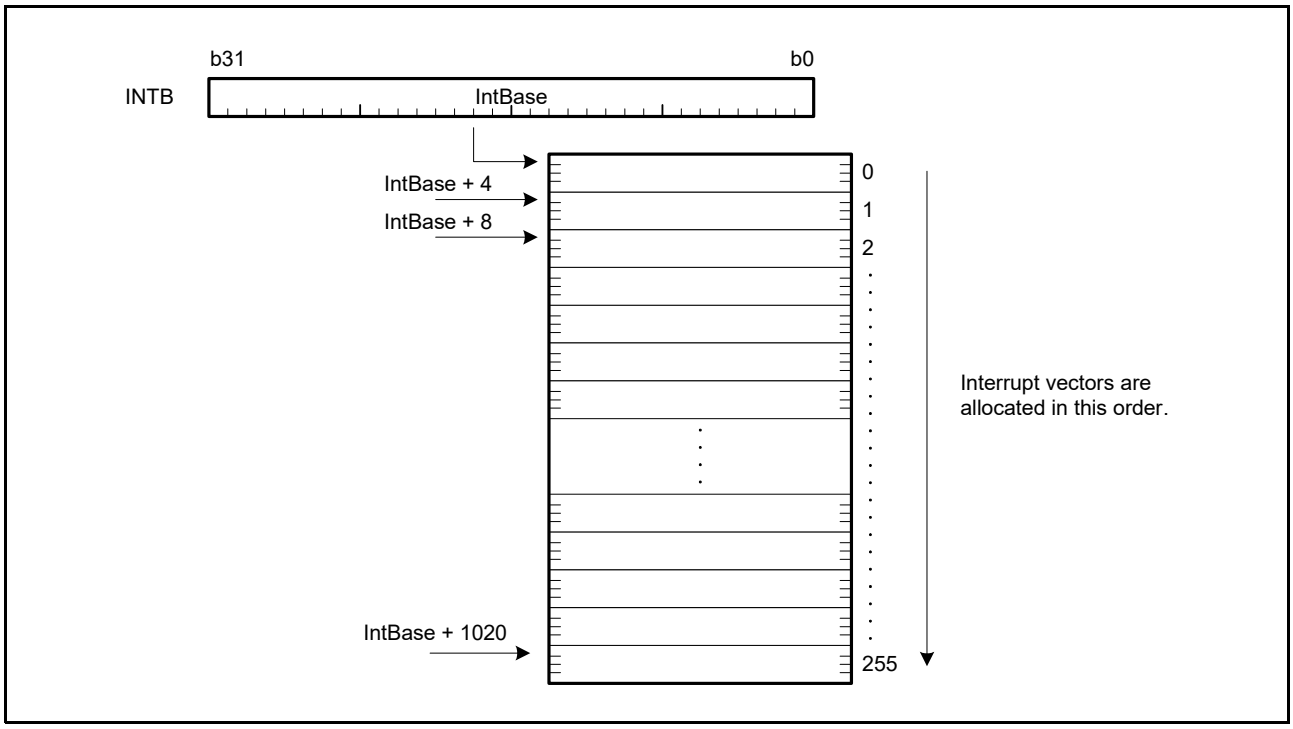


Figure 2.9 Interrupt Vector Table

2.7 Register Bank Save Function

The CPU has dedicated save register banks and functionality for using them for the fast saving and restoring of the values of CPU registers (see Figure 2.10). The save register banks enable the fast collective saving at the start of the exception handling routine and fast collective restoring of register values at the end of the exception handling routine.
The save register banks are only accessible by the SAVE and RSTR instructions, and are independent of the 4-Gbyte address space. Each of the multiple banks is used to save and restore the values of the following CPU registers: all general purpose registers except R0, the USP, FPSW, and accumulators (ACC0, ACC1). Values in the save register banks are undefined after a reset.
A unique number (bank number) is allocated to each save register bank.
The MCU has 16 save register banks, to which the bank numbers 0 to 15 are assigned.

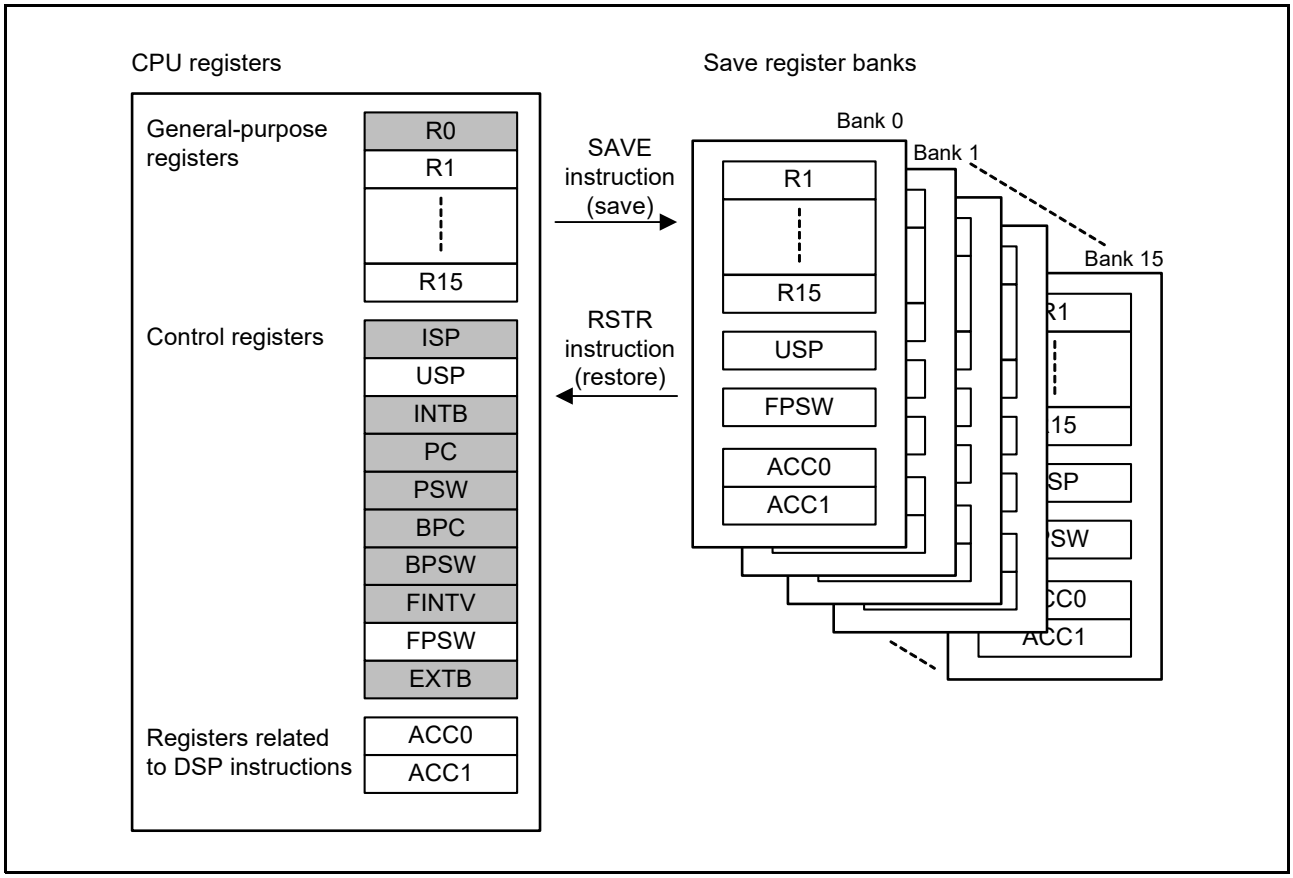


Figure 2.10 Save Register Banks

2.8 Operation of Instructions

2.8.1 Restrictions on RMPA and String-Manipulation Instructions

2.8.1.1 Transfer Size and Data Prefetching

The RMPA instruction and the string-manipulation instructions (SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, and SWHILE instructions) transfer data in longword units to speed up the reading of data from and writing of data to the memory. If the last of the data to be processed is less than a longword, data transfer proceeds with the sizes described below:

- RMPA, SSTR, SUNTIL, and SWHILE instructions: Size specified by the size specifier
- SCMPU, SMOVB, SMOVF, and SMOVU instructions: Byte

Additionally, in the above processing, the RMPA instruction and the string-manipulation instructions other than the SSTR instruction (that is, the SCMPU, SMOVB, SMOVF, SMOVU, SUNTIL, and SWHILE instructions) prefetch data when reading data from the memory. Data is prefetched from the prefetching start position with three bytes as the upper limit. The prefetching start positions of each operation are shown below.

- RMPA instruction: The multiplicand address specified by R1, and the multiplier address specified by R2
- SCMPU instruction: The source address specified by R1 for comparison, and the destination address specified by R2 for comparison
- SUNTIL and SWHILE instructions: The destination address specified by R1 for comparison
- SMOVB, SMOVF, and SMOVU instructions: The source address specified by R2 for transfer

2.8.1.2 Access to the External Space

Although the external space has a per-area ending-switching facility (only for data), the allocation of data to be handled by RMPA or string-manipulation instructions (SCMPU, SMOVB, SMOVF, SMOVU,SSTR, SUNTIL, and SWHILE instructions) to an area where the endian differs from that of the chip is prohibited, and operation is not guaranteed if this restriction is not observed. If data to be handled by RMPA or string-manipulation instructions are allocated to the external space, they must be allocated to areas where the endian setting is the same as that for the chip.

2.8.1.3 Access to I/O Registers

The allocation of data to be handled by RMPA or string-manipulation instructions (SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, and SWHILE instructions) to I/O registers is prohibited, and operation is not guaranteed if this restriction is not observed.

2.9 Numbers of Cycles

2.9.1 Instruction and Numbers of Cycles

Table 2.13 to Table 2.21 show the numbers of cycles in operation of each instruction. The listed numbers of cycles for access to memory are the numbers of cycles during no-wait access. The operands in the table below indicate the following meanings.

#IMM: Immediate
flag: bit, flag
Rs, Rs2, Rd, Rd2, Ri, Rb: General-purpose register
As, Ad: Accumulator
CR: Control register
dsp: displacement
pcdsp: displacement

Table 2.13 Numbers of Cycles for Arithmetic/logic Instructions

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Numbers of Cycles
Arithmetic/logic instructions (register-register, immediate-register)	• {ABS, NEG, NOT} “Rd”/“Rs, Rd” • {ADC, MAX, MIN, ROTL, ROTR} “#IMM, Rd”/“Rs, Rd” • ADD “#IMM, Rd”/“Rs, Rd”/“#IMM, Rs, Rd”/“Rs, Rs2, Rd” • {AND, MUL, OR, SUB, XOR} “#IMM, Rd”/“Rs, Rd”/“Rs, Rs2, Rd” • {CMP, TST} “#IMM, Rs”/“Rs, Rs2” • NOP • {ROLC, RORC, SAT} “Rd” • SBB “Rs, Rd” • {SHAR, SHLL, SHLR} “#IMM, Rd”/“Rs, Rd”/“#IMM, Rs, Rd”	1
	• DIV “#IMM, Rd”/“Rs, Rd”	3 to 20*1
	• DIVU “#IMM, Rd”/“Rs, Rd”	2 to 18*1
	• {EMUL, EMULU} “#IMM, Rd”/“Rs, Rd”	2
	• SATR	3
Arithmetic/logic instructions (memory source operand)	• {ADC, ADD, AND, MAX, MIN, MUL, OR, SBB, SUB, XOR} “[Rs], Rd”/“dsp[Rs], Rd” • {CMP, TST} “[Rs], Rs2”/“dsp[Rs], Rs2”	3
	• DIV “[Rs], Rd / dsp[Rs], Rd”	5 to 22*1
	• DIVU“[Rs], Rd / dsp[Rs], Rd”	4 to 20*1
	• {EMUL, EMULU} “[Rs], Rd”/“dsp[Rs], Rd”	4
	• RMPA.B	6+7×floor(n/4)+4×(n%4) n: Number of processing bytes*2
	• RMPA.W	6+5×floor(n/2)+4×(n%2) n: Number of processing words*2
	• RMPA.L	6+4n n: Number of processing longwords

Note 1. The numbers of cycles for the dividing instruction varies according to the divisor and dividend.
Note 2. floor (x): Max. integer that is smaller than x.

Table 2.14 Numbers of Cycles for Transfer Instructions

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Numbers of Cycles
Transfer instructions (register-register, immediate-register)	• MOV “#IMM, Rd”/“Rs, Rd” • {MOVU, REVL, REVW} “Rs, Rd” • SC*Cnd* “Rd” • {STNZ, STZ} “#IMM, Rd”/ “Rs, Rd”	1
	• XCHG “Rs, Rd”	2
Transfer instructions (load operation)	• {MOV, MOVU} “[Rs], Rd”/“dsp[Rs], Rd”/“[Rs+], Rd”/“[-Rs], Rd”/“[Ri, Rb], Rd” • MOVLI “[Rs], Rd” • POP “Rd”	Throughput: 1 Latency: 2*1
	• POPC “CR”	Throughput: 3 Latency: 4*1
	• POPM “Rd-Rd2”	Throughput: n Latency: n+1 n: Number of registers*1, *2
Transfer instructions (store operation)	• MOV “Rs, [Rd]”/“Rs, dsp[Rd]”/“Rs, [Rd+]”/“Rs, [-Rd]”/“Rs, [Ri, Rb]”/“#IMM, dsp[Rd]”/“#IMM, [Rd]” • PUSH “Rs” • PUSHC “CR” • SC*Cnd* “[Rd]”/“dsp[Rd]” • MOVCO “Rs, [Rd]”	1
	• PUSHM “Rs-Rs2”	n n: Number of registers*3
Transfer instructions (memory-register)	• XCHG “[Rs], Rd”/“dsp[Rs], Rd”	2
Transfer instructions (memory-memory)	• MOV “[Rs], [Rd]”/“dsp[Rs], [Rd]”/“[Rs], dsp[Rd]”/“dsp[Rs], dsp[Rd]” • PUSH “[Rs]”/“dsp[Rs]”	3
Transfer instructions (bit field)	• {BFMOV, BFMOVZ} “#IMM, #IMM, #IMM, R, R”	1

Note 1. When the load data is used by the subsequent instruction, the numbers of cycles described as “latency” is counted as the numbers of cycles for the memory load instruction. For the cycles other than the memory load instruction, the numbers of cycles described as “throughput” is counted.

Note 2. The POPM instruction is converted into multiple load operations. The processing is the same as the one for the load operations of the MOV instruction, where the operation is repeated for the number of specified registers.

Note 3. The PUSHM instruction is converted into multiple store operations. The processing is the same as the one for the store operations of the MOV instruction, where the operation is repeated for the number of specified registers.

Table 2.15 Numbers of Cycles for Bit Manipulation Instructions

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Numbers of Cycles
Bit manipulation instructions (register)	• {BCLR, BNOT, BSET} “#IMM, Rd”/“Rs, Rd” • BM*Cnd* “#IMM, Rd” • BTST “#IMM, Rs”/“Rs, Rs2”	1
Bit manipulation instructions (memory source operand)	• {BCLR, BNOT, BSET} “#IMM, [Rd]”/“#IMM, dsp[Rd]”/“Rs, [Rd]”/“Rs, dsp[Rd]” • BM*Cnd* “#IMM, [Rd]”/“#IMM, dsp[Rd]” • BTST “#IMM, [Rs]”/“#IMM, dsp[Rs]”/“Rs, [Rs2]”/“Rs, dsp[Rs2]”	3

Table 2.16 Numbers of Cycles for Branch Instructions

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Numbers of Cycles
Branch instructions	• B*Cnd* “pcdsp” • {BRA, BSR} “pcdsp”/“Rs” • {JMP, JSR} “Rs”	Branch taken: 3 Branch not taken: 1
	• RTE	6
	• RTFI	3
	• RTS	5
	• RTSD “#IMM”	5
	• RTSD “#IMM, Rd-Rd2”	Throughput: n<5?5:1+n Latency: n<4?5:2+n n: Number of registers*1

?: Conditional operator

Note 1. When the load data is used by the subsequent instruction, the numbers of cycles described as “latency” is counted as the numbers of cycles for the memory load instruction. For the cycles other than the memory load instruction, the numbers of cycles described as “throughput” is counted.

Table 2.17 Numbers of Cycles for Single-Precision Floating-Point Operation Instructions

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Numbers of Cycles
Single-precision floating-point operation instructions (register-register, immediate-register)	• {FADD, FSUB} “#IMM, Rd”/ “Rs, Rd”/ “Rs, Rs2, Rd”	2
	• FCMP “#IMM, Rs”/“Rs, Rs2”	1
	• FDIV “#IMM, Rd”/“Rs, Rd”	16
	• FMUL “#IMM, Rd”/ “Rs, Rd” / “Rs, Rs2, Rd”	2
	• FSQRT “Rs, Rd”	16
	• {FTOI, ROUND, ITOF} “Rs, Rd”	2
	• {FTOU, UTOF} “Rs, Rd”	2
Single-precision floating-point operation instructions (memory source operand)	• {FADD, FSUB} “[Rs], Rd”/“dsp[Rs], Rd”	4
	• FCMP “[Rs], Rs2”/“dsp[Rs], Rs2”	3
	• FDIV “[Rs], Rd”/“dsp[Rs], Rd”	18
	• FMUL “[Rs], Rd”/“dsp[Rs], Rd”	4
	• FSQRT “[Rs], Rd”/“dsp[Rs], Rd”	18
	• {FTOI, ROUND, ITOF} “[Rs], Rd”/“dsp[Rs], Rd”	4
	• {FTOU, UTOF} “[Rs], Rd”/“dsp[Rs], Rd”	4

Table 2.18 Numbers of Cycles for DSP Instructions

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Numbers of Cycles
DSP instructions	• {EMULA, EMACA, EMSBA, MULLH, MULHI, MULLO, MACLH, MACHI, MACLO, MSBLH, MSBHI, MSBLO}“Rs, Rs2, Ad” • {MVFACHI, MVFACMI, MVFACLO, MVFACGU} “#IMM, As, Rd” • {MVTACHI, MVTACLO, MVTACGU} “Rs, Ad” • {RDACW, RDACL, RACW, RACL} “#IMM, Ad”	1

Table 2.19 Numbers of Cycles for String Manipulation Instructions

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Numbers of Cycles
String manipulation instructions*1	• SCMPU	2+4×floor(n/4)+4×(n%4) n: Number of comparison bytes*2
	• SMOVB	n>3?6+3×floor(n/4)+3×(n%4):2+3n n: Number of transfer bytes*2
	• SMOVF, SMOVU	2+3×floor(n/4)+3×(n%4) n: Number of transfer bytes*2
	• SSTR.B	2+floor(n/4)+n%4 n: Number of transfer bytes*2
	• SSTR.W	2+floor(n/2)+n%2 n: Number of transfer words*2
	• SSTR.L	2+n n: Number of transfer longwords
	• SUNTIL.B, SWHILE.B	3+3×floor(n/4)+3×(n%4) n: Number of comparison bytes*2
	• SUNTIL.W, SWHILE.W	3+3×floor(n/2)+3×(n%2) n: Number of comparison words*2
	• SUNTIL.L, SWHILE.L	3+3×n n: Number of comparison longwords

?: Conditional operator

Note 1. Each of the SCMPU, SMOVU, SWHILE, and SUNTIL instructions ends the execution regardless of the specified cycles, if the end condition is satisfied during execution.
Note 2. floor (x): Max. integer that is smaller than x.

Table 2.20 Numbers of Cycles for System Manipulation Instructions

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Numbers of Cycles
System manipulation instructions	• {CLRPSW, SETPSW}“flag” • MVTC “#IMM, CR”/“Rs, CR” • MVFC “CR, Rd” • MVTIPL “#IMM”	1
	• RTE	6
	• RTFI	3

Table 2.21 Numbers of Cycles for Instructions for Register Bank Save Function

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Numbers of Cycles
Instructions for register bank save function	• SAVE “#IMM”/“R”	1
	• RSTR “#IMM”/“R”	3 to 6

2.9.2 Numbers of Cycles for Response to Interrupts

Table 2.22 lists numbers of cycles taken by processing for response to interrupts.

Table 2.22 Numbers of Cycles for Response to Interrupts

Type of Interrupt Request/Details of Processing	Fast Interrupt	Other Interrupts
ICU Judgment of priority order	2 cycles	
CPU Numbers of cycles from notification to acceptance of the interrupt request	N cycles (varies with the instruction being executed at the time the interrupt was received)	
CPU Pre-processing by hardware Saving the current PC and PSW values in RAM (or in control registers in the case of the fast interrupt) Reading of the vector Branching to the start of the exception handling routine	4 cycles	6 cycles

Times calculated from the values in Table 2.22 will be applicable when access to memory from the CPU is processed with no waiting. This MCU has a RAM and code flash memory that allow no-wait access. Numbers of cycles for response to interrupts can be minimized by placing program code (and vectors) in code flash memory and the stack in RAM. Furthermore, place the addresses where the exception handling routine start on 8-byte boundaries.
For information on the numbers of cycles from notification to acceptance of the interrupt request, indicated by N in the table above, see Table 2.13 to Table 2.21.
The timing of interrupt acceptance depends on the execution state of the instruction. For more information on this, see section 13.3.1, Acceptance Timing and Saved PC Value.

2.10 Usage Note

2.10.1 Notes on Self-Diagnosis of the RAM for the Save Register Banks

The save register banks in this MCU are configured of RAM. As the save register banks are equipped with a buffer, data may be read from the buffer rather than from the memory cells of the RAM when the same address is to be read by a RSTR instruction after a write operation by the SAVE instruction. When running self-diagnosis of the RAM in the save register banks, confirm that the data have actually been written to the memory by following the procedure below so that data will not be read from the buffer.

(1) Write data to the bank targeted for diagnosis with the SAVE instruction.
(2) Write data to a bank different from the bank in the procedure (1) with the SAVE instruction.
(3) Read the data from the bank in the procedure (1) with the RSTR instruction.

3. Operating Modes

3.1 Operating Mode Types and Selection

There are two types of operating-mode selection: one is be selected by the level of pins when a reset (RES# pin reset, power-on reset, or LVD0 reset) is released, and the other is selected by software after a reset is released.
Table 3.1 shows the relationship between levels on the mode-setting pin (MD) when a reset is released and the operating mode selected at that time. For details on each of the operating modes, see section 3.3, Details of Operating Modes. Operation starts with the on-chip ROM (code flash memory and data flash memory) enabled, regardless of the mode in which operation started.

Table 3.1 Selection of Operating Modes by the Mode-Setting Pins when the Reset is Released

Mode-Setting Pin	
MD*1, *2	**Operating Mode**
High	Single-chip mode
Low	Boot mode (SCI interface)
Low → High*3	Boot mode (FINE interface)

Note 1. Do not change the level input to the MD pin during the following periods when the MCU operation is in a transition state.
- The waiting time after release from the RES# pin reset
- Power-on reset time
- LVD0 reset time
For details on the periods above, see section 49, Electrical Characteristics.

Note 2. After the chip starts up in the single-chip mode, this pin can be used as general I/O port pin PN6. For details, see section 20.3.13, General-purpose Input/Output Pin Select Extension Register (GPSEXT).

Note 3. After release the reset state while the MD pin is at the low level, switch it to the high level within 20 to 100 msec.

The endian is selectable in single-chip mode. Endian is selected by the endian selection bits (MDE[2:0]) in the endian select register (MDE). Table 3.2 lists the correspondence between the setting and endian. For details on selection of endian, see section 7.2.5, Endian Select Register (MDE).

Table 3.2 Selection of Endian

Setting of the MDE[2:0] Bits	Selected Endian
000b	Big endian
111b	Little endian

3.2 Register Descriptions

3.2.1 Mode Monitor Register (MDMONR)

Address(es): 0008 0000h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MD
Value after reset:	0	0	0	0	0	0	0	x	0	0	0	0	0	0	0	0/1

Bit	Symbol	Bit Name	Description	R/W
b0	MD	MD Pin Status Flag	0: The MD pin was low. 1: The MD pin was high.	R
b7 to b1	—	Reserved	These bits are read as 0.	R
b8	—	Reserved	The read value is undefined.	R
b15 to b9	—	Reserved	These bits are read as 0.	R

Note: The value of the MD flag is undefined when the setting of the GPSEXT.GPSMD bit is 1 (selecting the general-purpose input/output pin function).

3.2.2 System Control Register 1 (SYSCR1)

Address(es): 0008 0008h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RAME
Value after reset:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	RAME	RAM Enable	0: The RAM is disabled. 1: The RAM is enabled.	R/W
b7 to b1	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

RAME Bit (RAM Enable)

The RAME bit enables or disables the RAM.

A 0 should not be written to this bit during access to the RAM. When accessing the RAM immediately after changing the RAME bit from 0 to 1, make sure that the RAME bit is 1 before the access.

Even when the RAME bit is set to 0, the RAM retains its value. However, the voltage must be maintained at no less than the specified RAM standby voltage (VRAM), which is stipulated in section 49, Electrical Characteristics.

3.2.3 Voltage Level Setting Register (VOLSR)

Address(es): 0008 C295h

	b7	b6	b5	b4	b3	b2	b1	b0
	RICVLS	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	RICVLS	RIIC Operating Voltage Setting	0: VCC ≥ 4.5 V 1: VCC < 4.5 V	R/W

The VOLSR register specifies the power-supply voltage when the RIIC is in use.

RICVLS Bit (RIIC Operating Voltage Setting)

The RICVLS bit controls the slew rate for the RIIC. Set the bit according to the VCC voltage. Set this bit before releasing the RIIC from the module-stop state.

3.3 Details of Operating Modes

3.3.1 Single-Chip Mode

In single-chip mode, all I/O port pins are available for use as input or output port pins, inputs or outputs for peripheral functions, or as interrupt inputs.
If the high level is on the MD pin and the reset is released, the MCU starts in single-chip mode.

3.3.2 Boot Mode (SCI Interface)

In this mode, the on-chip flash memory writing program (boot program) stored in a dedicated area of the MCU operates. The on-chip ROM (code flash memory and data flash memory) can be modified from outside the MCU by using a universal asynchronous receiver/transmitter (SCI1). For details, see section 48, Flash Memory (FLASH).
The MCU starts up in boot mode if the low level is on the MD pin, and the reset is released.

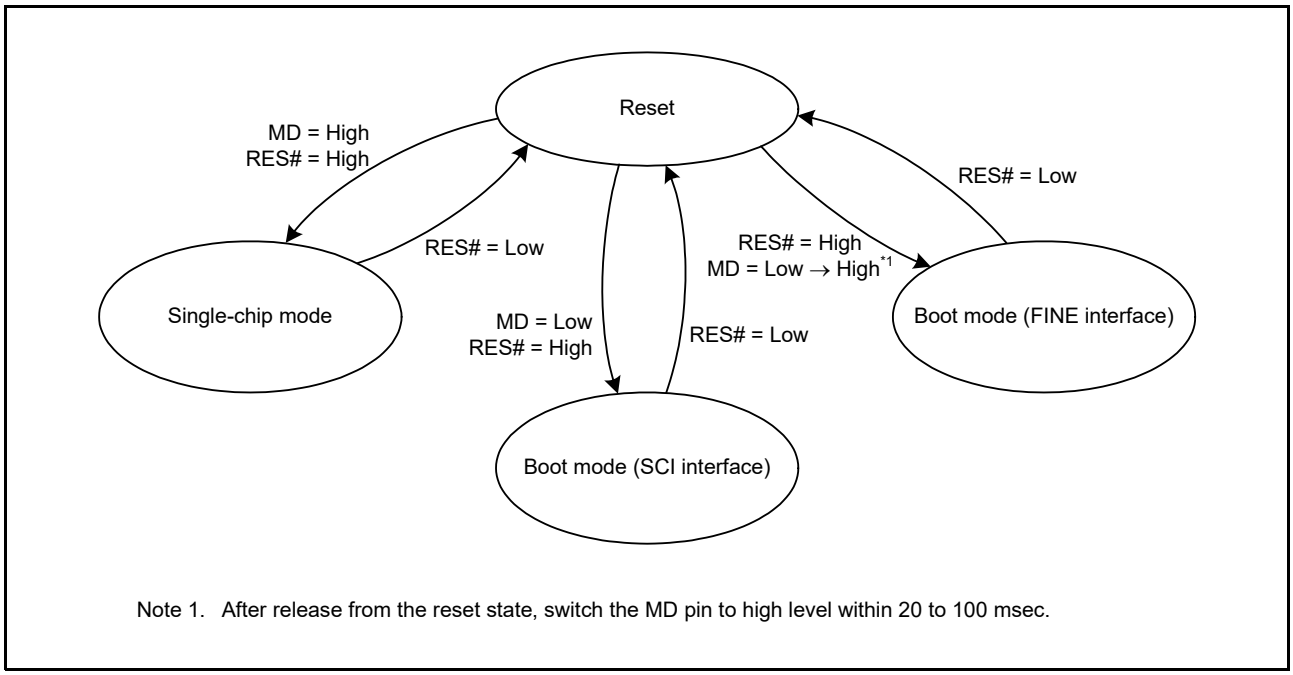
3.3.3 Boot Mode (FINE Interface)

In this mode, the on-chip flash memory writing program (boot program) stored in a dedicated area of the MCU operates. The on-chip ROM (code flash memory and data flash memory) can be modified from outside the MCU by using the FINE. For details, see section 48, Flash Memory (FLASH).
After the reset is released while the low level is on the MD pin, if the MD pin is set to High within 20 to 100 msec, the MCU will start up in boot mode (FINE interface).

3.4 Transitions of Operating Modes

3.4.1 Operating Mode Transitions Determined by the Mode-Setting Pins

Figure 3.1 shows operating mode transitions determined by the setting of the MD pin.



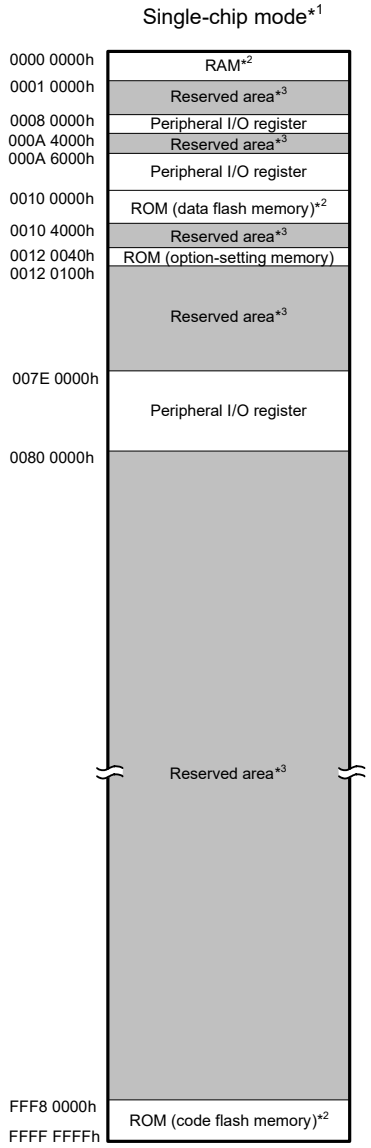
Note 1. After release from the reset state, switch the MD pin to high level within 20 to 100 msec.

Figure 3.1 Mode-Setting Pin Level and Operating Mode

4. Address Space

4.1 Address Space

This MCU has a 4-Gbyte address space, consisting of the range of addresses from 0000 0000h to FFFF FFFFh. Figure 4.1 shows the memory maps in the respective operating modes. Accessible areas will differ according to the operating mode and states of control bits.



Note 1. The memory map in boot mode is the same as that in single-chip mode.
Note 2. The capacity of ROM/RAM differs depending on the products.

Code Flash Memory			Data Flash Memory		RAM	
	Address					
Capacity	Linear mode	Dual mode (BANKSEL.BANKSWP[2:0] = 111b)	Capacity	Address	Capacity	Address
512 Kbytes	FFF8 0000h to FFFF FFFFh	bank 1: FFF8 0000h to FFFB FFFFh	16 Kbytes	0010 0000h to 0010 3FFFh	64 Kbytes	0000 0000h to 0000 FFFFh
		bank 0: FFFC 0000h to FFFF FFFFh				
256 Kbytes	FFFC 0000h to FFFF FFFFh	—				
128 Kbytes	FFFE 0000h to FFFF FFFFh	—				
256 Kbytes	FFFC 0000h to FFFF FFFFh	—			48 Kbytes	0000 0000h to 0000 BFFFh
128 Kbytes	FFFE 0000h to FFFF FFFFh	—				

Note 3. Reserved areas should not be accessed.

Figure 4.1 Memory Map in Each Operating Mode

5. I/O Registers

This section gives information on the on-chip I/O register addresses. The information is given as shown below. Notes on writing to registers are also given at the end.

(1) I/O register addresses (address order)

- Registers are listed from the lower allocation addresses.
- Registers are classified according to module symbols.
- The number of access cycles indicates the number of cycles based on the specified reference clock.
- Among the internal I/O register area, addresses not listed in the list of registers are reserved. Reserved addresses must not be accessed. Do not access these addresses; otherwise, the operation when accessing these bits and subsequent operations cannot be guaranteed.

(2) Notes on writing to I/O registers

When writing to an I/O register, the CPU starts executing the subsequent instruction before completing I/O register write. This may cause the subsequent instruction to be executed before the post-update I/O register value is reflected on the operation.
As described in the following examples, special care is required for the cases in which the subsequent instruction must be executed after the post-update I/O register value is actually reflected.

[Examples of cases requiring special care]

- The subsequent instruction must be executed while an interrupt request is disabled with the IENj bit in IERn of the ICU (interrupt request enable bit) set to 0.
- A WAIT instruction is executed immediately after the preprocessing for causing a transition to the low power consumption state.

In the above cases, after writing to an I/O register, wait until the write operation is completed using the following procedure and then execute the subsequent instruction.

(a) Write to an I/O register.
(b) Read the value from the I/O register to a general register.
(c) Execute the operation using the value read.
(d) Execute the subsequent instruction.

[Instruction examples]

- Byte-size I/O registers

```
MOV.L  #SFR_ADDR, R1
MOV.B  #SFR_DATA, [R1]
CMP    [R1].UB, R1
;; Next process
```

- Word-size I/O registers

```
MOV.L  #SFR_ADDR, R1
MOV.W  #SFR_DATA, [R1]
CMP    [R1].W, R1
;; Next process
```

- Longword-size I/O registers

```
MOV.L  #SFR_ADDR, R1
MOV.L  #SFR_DATA, [R1]
CMP    [R1].L, R1
;; Next process
```

If multiple registers are written to and a subsequent instruction should be executed after the write operations are entirely completed, only read the I/O register that was last written to and execute the operation using the value; it is not necessary to read or execute operation for all the registers that were written to.

(3) Number of Access Cycles to I/O Registers

For the number of I/O register access cycles, refer to Table 5.1, List of I/O Registers (Address Order).
The number of access cycles to I/O registers is obtained by following equation.*1

Number of access cycles to I/O registers = Number of bus cycles for internal main bus 1 +
Number of divided clock synchronization cycles +
Number of bus cycles for internal peripheral busses 1 to 6

The number of bus cycles of internal peripheral bus 1 to 6 differs according to the register to be accessed.
When peripheral functions connected to internal peripheral bus 2 to 6 are accessed, the number of divided clock synchronization cycles is added.
The number of divided clock synchronization cycles differs depending on the frequency ratio between ICLK and PCLK (or FCLK) or bus access timing.
In the peripheral function unit, when the frequency ratio of ICLK is equal to or greater than that of PCLK (or FCLK), the sum of the number of bus cycles for internal main bus 1 and the number of the divided clock synchronization cycles will be one cycle of PCLK (or FCLK) at a maximum. Therefore, one PCLK (or FCLK) has been added to the number of access states shown in Table 5.1.
When the frequency ratio of ICLK is lower than that of PCLK (or FCLK), the subsequent bus access is started from the ICLK cycle following the completion of the access to the peripheral functions. Therefore, the access cycles are described on an ICLK basis.

Note 1. This applies to the number of cycles when the access from the CPU does not conflict with the bus access from the different bus master (DMAC or DTC).

(4) Notes on Sleep Mode and Mode Transitions

During sleep mode or mode transitions, do not write to the registers related to system control (indicated by 'SYSTEM' in the Module Symbol column in Table 5.1, List of I/O Registers (Address Order)).

(5) Restrictions in Relation to RMPA and String-Manipulation Instructions

The allocation of data to be handled by RMPA or string-manipulation instructions to I/O registers is prohibited, and operation is not guaranteed if this restriction is not observed.

5.1 I/O Register Addresses (Address Order)

Table 5.1 List of I/O Registers (Address Order) (1 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 0000h	SYSTEM	Mode Monitor Register	MDMONR	16	16	3 ICLK		section 3
0008 0008h	SYSTEM	System Control Register 1	SYSCR1	16	16	3 ICLK		section 3
0008 000Ch	SYSTEM	Standby Control Register	SBYCR	16	16	3 ICLK		section 11
0008 0010h	SYSTEM	Module Stop Control Register A	MSTPCRA	32	32	3 ICLK		section 11
0008 0014h	SYSTEM	Module Stop Control Register B	MSTPCRB	32	32	3 ICLK		section 11
0008 0018h	SYSTEM	Module Stop Control Register C	MSTPCRC	32	32	3 ICLK		section 11
0008 001Ch	SYSTEM	Module Stop Control Register D	MSTPCRD	32	32	3 ICLK		section 11
0008 0020h	SYSTEM	System Clock Control Register	SCKCR	32	32	3 ICLK		section 9
0008 0024h	SYSTEM	System Clock Control Register 2	SCKCR2	16	16	3 ICLK		section 9
0008 0026h	SYSTEM	System Clock Control Register 3	SCKCR3	16	16	3 ICLK		section 9
0008 0028h	SYSTEM	PLL Control Register	PLLCR	16	16	3 ICLK		section 9
0008 002Ah	SYSTEM	PLL Control Register 2	PLLCR2	8	8	3 ICLK		section 9
0008 0032h	SYSTEM	Main Clock Oscillator Control Register	MOSCCR	8	8	3 ICLK		section 9
0008 0034h	SYSTEM	Low-Speed On-Chip Oscillator Control Register	LOCOCR	8	8	3 ICLK		section 9
0008 0035h	SYSTEM	IWDT-Dedicated On-Chip Oscillator Control Register	ILOCOCR	8	8	3 ICLK		section 9
0008 0036h	SYSTEM	High-Speed On-Chip Oscillator Control Register	HOCOCR	8	8	3 ICLK		section 9
0008 0037h	SYSTEM	High-Speed On-Chip Oscillator Control Register 2	HOCOCR2	8	8	3 ICLK		section 9
0008 003Ch	SYSTEM	Oscillation Stabilization Flag Register	OSCOVFSR	8	8	3 ICLK		section 9
0008 0040h	SYSTEM	Oscillation Stop Detection Control Register	OSTDCR	8	8	3 ICLK		section 9
0008 0041h	SYSTEM	Oscillation Stop Detection Status Register	OSTDSR	8	8	3 ICLK		section 9
0008 00A1h	SYSTEM	Sleep Mode Return Clock Source Switching Register	RSTCKCR	8	8	3 ICLK		section 11
0008 00A2h	SYSTEM	Main Clock Oscillator Wait Control Register	MOSCWTCR	8	8	3 ICLK		section 9
0008 00C0h	SYSTEM	Reset Status Register 2	RSTSR2	8	8	3 ICLK		section 6
0008 00C2h	SYSTEM	Software Reset Register	SWRR	16	16	3 ICLK		section 6
0008 00D0h	SYSTEM	Device Function Select Register 0	PRDFR0	32	32	3 ICLK		section 32, section 33
0008 00E0h	SYSTEM	Voltage Monitoring 1 Circuit Control Register 1	LVD1CR1	8	8	3 ICLK		section 8
0008 00E1h	SYSTEM	Voltage Monitoring 1 Circuit Status Register	LVD1SR	8	8	3 ICLK		section 8
0008 00E2h	SYSTEM	Voltage Monitoring 2 Circuit Control Register 1	LVD2CR1	8	8	3 ICLK		section 8
0008 00E3h	SYSTEM	Voltage Monitoring 2 Circuit Status Register	LVD2SR	8	8	3 ICLK		section 8
0008 03FEh	SYSTEM	Protect Register	PRCR	16	16	3 ICLK		section 12
0008 1200h	RAM	RAM Operating Mode Control Register	RAMMODE	8	8	2 ICLK		section 47
0008 1201h	RAM	RAM Error Status Register	RAMSTS	8	8	2 ICLK		section 47
0008 1204h	RAM	RAM Protection Register	RAMPRCR	8	8	2 ICLK		section 47
0008 1208h	RAM	RAM Error Address Capture Register	RAMECAD	32	32	2 ICLK		section 47
0008 1300h	BSC	Bus Error Status Clear Register	BERCLR	8	8	2 ICLK		section 15
0008 1304h	BSC	Bus Error Monitoring Enable Register	BEREN	8	8	2 ICLK		section 15
0008 1308h	BSC	Bus Error Status Register 1	BERSR1	8	8	2 ICLK		section 15
0008 130Ah	BSC	Bus Error Status Register 2	BERSR2	16	16	2 ICLK		section 15
0008 1310h	BSC	Bus Priority Control Register	BUSPRI	16	16	2 ICLK		section 15
0008 1404h	TFU	Fixed-Point Sincos Input and Output Setting Register	FXSCIOC	8	8	2 ICLK		section 40
0008 1405h	TFU	Fixed-Point Atanhypot_k Input and Output Setting Register	FXATIOC	8	8	2 ICLK		section 40
0008 1408h	TFU	Trigonometric Status Register	TRGSTS	8	8	2 ICLK		section 40
0008 1410h	TFU	Floating-Point Sincos Data Register 0	FPSCDT0	32	32	2 ICLK		section 40
0008 1414h	TFU	Floating-Point Sincos Data Register 1	FPSCDT1	32	32	2 ICLK		section 40
0008 1418h	TFU	Floating-Point Atanhypot_k Data Register 0	FPATDT0	32	32	2 ICLK		section 40
0008 141Ch	TFU	Floating-Point Atanhypot_k Data Register 1	FPATDT1	32	32	2 ICLK		section 40

Table 5.1 List of I/O Registers (Address Order) (2 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 1420h	TFU	Fixed-Point Sincos Data Register 0	FXSCDT0	32	32	2 ICLK		section 40
0008 1424h	TFU	Fixed-Point Sincos Data Register 1	FXSCDT1	32	32	2 ICLK		section 40
0008 1428h	TFU	Fixed-Point Atanhypot_k Data Register 0	FXATDT0	32	32	2 ICLK		section 40
0008 142Ch	TFU	Fixed-Point Atanhypot_k Data Register 1	FXATDT1	32	32	2 ICLK		section 40
0008 1430h	TFU	Data Saving and Restoring Register 0	DTSR0	32	32	2 ICLK		section 40
0008 1434h	TFU	Data Saving and Restoring Register 1	DTSR1	32	32	2 ICLK		section 40
0008 2000h	DMAC0	DMA Source Address Register	DMSAR	32	32	2 ICLK		section 17
0008 2004h	DMAC0	DMA Destination Address Register	DMDAR	32	32	2 ICLK		section 17
0008 2008h	DMAC0	DMA Transfer Count Register	DMCRA	32	32	2 ICLK		section 17
0008 200Ch	DMAC0	DMA Block Transfer Count Register	DMCRB	16	16	2 ICLK		section 17
0008 2010h	DMAC0	DMA Transfer Mode Register	DMTMD	16	16	2 ICLK		section 17
0008 2013h	DMAC0	DMA Interrupt Setting Register	DMINT	8	8	2 ICLK		section 17
0008 2014h	DMAC0	DMA Address Mode Register	DMAMD	16	16	2 ICLK		section 17
0008 2018h	DMAC0	DMA Offset Register	DMOFR	32	32	2 ICLK		section 17
0008 201Ch	DMAC0	DMA Transfer Enable Register	DMCNT	8	8	2 ICLK		section 17
0008 201Dh	DMAC0	DMA Software Start Register	DMREQ	8	8	2 ICLK		section 17
0008 201Eh	DMAC0	DMA Status Register	DMSTS	8	8	2 ICLK		section 17
0008 201Fh	DMAC0	DMA Request Source Flag Control Register	DMCSL	8	8	2 ICLK		section 17
0008 2040h	DMAC1	DMA Source Address Register	DMSAR	32	32	2 ICLK		section 17
0008 2044h	DMAC1	DMA Destination Address Register	DMDAR	32	32	2 ICLK		section 17
0008 2048h	DMAC1	DMA Transfer Count Register	DMCRA	32	32	2 ICLK		section 17
0008 204Ch	DMAC1	DMA Block Transfer Count Register	DMCRB	16	16	2 ICLK		section 17
0008 2050h	DMAC1	DMA Transfer Mode Register	DMTMD	16	16	2 ICLK		section 17
0008 2053h	DMAC1	DMA Interrupt Setting Register	DMINT	8	8	2 ICLK		section 17
0008 2054h	DMAC1	DMA Address Mode Register	DMAMD	16	16	2 ICLK		section 17
0008 205Ch	DMAC1	DMA Transfer Enable Register	DMCNT	8	8	2 ICLK		section 17
0008 205Dh	DMAC1	DMA Software Start Register	DMREQ	8	8	2 ICLK		section 17
0008 205Eh	DMAC1	DMA Status Register	DMSTS	8	8	2 ICLK		section 17
0008 205Fh	DMAC1	DMA Request Source Flag Control Register	DMCSL	8	8	2 ICLK		section 17
0008 2080h	DMAC2	DMA Source Address Register	DMSAR	32	32	2 ICLK		section 17
0008 2084h	DMAC2	DMA Destination Address Register	DMDAR	32	32	2 ICLK		section 17
0008 2088h	DMAC2	DMA Transfer Count Register	DMCRA	32	32	2 ICLK		section 17
0008 208Ch	DMAC2	DMA Block Transfer Count Register	DMCRB	16	16	2 ICLK		section 17
0008 2090h	DMAC2	DMA Transfer Mode Register	DMTMD	16	16	2 ICLK		section 17
0008 2093h	DMAC2	DMA Interrupt Setting Register	DMINT	8	8	2 ICLK		section 17
0008 2094h	DMAC2	DMA Address Mode Register	DMAMD	16	16	2 ICLK		section 17
0008 209Ch	DMAC2	DMA Transfer Enable Register	DMCNT	8	8	2 ICLK		section 17
0008 209Dh	DMAC2	DMA Software Start Register	DMREQ	8	8	2 ICLK		section 17
0008 209Eh	DMAC2	DMA Status Register	DMSTS	8	8	2 ICLK		section 17
0008 209Fh	DMAC2	DMA Request Source Flag Control Register	DMCSL	8	8	2 ICLK		section 17
0008 20C0h	DMAC3	DMA Source Address Register	DMSAR	32	32	2 ICLK		section 17
0008 20C4h	DMAC3	DMA Destination Address Register	DMDAR	32	32	2 ICLK		section 17
0008 20C8h	DMAC3	DMA Transfer Count Register	DMCRA	32	32	2 ICLK		section 17
0008 20CCh	DMAC3	DMA Block Transfer Count Register	DMCRB	16	16	2 ICLK		section 17
0008 20D0h	DMAC3	DMA Transfer Mode Register	DMTMD	16	16	2 ICLK		section 17
0008 20D3h	DMAC3	DMA Interrupt Setting Register	DMINT	8	8	2 ICLK		section 17
0008 20D4h	DMAC3	DMA Address Mode Register	DMAMD	16	16	2 ICLK		section 17
0008 20DCh	DMAC3	DMA Transfer Enable Register	DMCNT	8	8	2 ICLK		section 17
0008 20DDh	DMAC3	DMA Software Start Register	DMREQ	8	8	2 ICLK		section 17
0008 20DEh	DMAC3	DMA Status Register	DMSTS	8	8	2 ICLK		section 17

Table 5.1 List of I/O Registers (Address Order) (3 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 20DFh	DMAC3	DMA Request Source Flag Control Register	DMCSL	8	8	2 ICLK		section 17
0008 2100h	DMAC4	DMA Source Address Register	DMSAR	32	32	2 ICLK		section 17
0008 2104h	DMAC4	DMA Destination Address Register	DMDAR	32	32	2 ICLK		section 17
0008 2108h	DMAC4	DMA Transfer Count Register	DMCRA	32	32	2 ICLK		section 17
0008 210Ch	DMAC4	DMA Block Transfer Count Register	DMCRB	16	16	2 ICLK		section 17
0008 2110h	DMAC4	DMA Transfer Mode Register	DMTMD	16	16	2 ICLK		section 17
0008 2113h	DMAC4	DMA Interrupt Setting Register	DMINT	8	8	2 ICLK		section 17
0008 2114h	DMAC4	DMA Address Mode Register	DMAMD	16	16	2 ICLK		section 17
0008 211Ch	DMAC4	DMA Transfer Enable Register	DMCNT	8	8	2 ICLK		section 17
0008 211Dh	DMAC4	DMA Software Start Register	DMREQ	8	8	2 ICLK		section 17
0008 211Eh	DMAC4	DMA Status Register	DMSTS	8	8	2 ICLK		section 17
0008 211Fh	DMAC4	DMA Request Source Flag Control Register	DMCSL	8	8	2 ICLK		section 17
0008 2140h	DMAC5	DMA Source Address Register	DMSAR	32	32	2 ICLK		section 17
0008 2144h	DMAC5	DMA Destination Address Register	DMDAR	32	32	2 ICLK		section 17
0008 2148h	DMAC5	DMA Transfer Count Register	DMCRA	32	32	2 ICLK		section 17
0008 214Ch	DMAC5	DMA Block Transfer Count Register	DMCRB	16	16	2 ICLK		section 17
0008 2150h	DMAC5	DMA Transfer Mode Register	DMTMD	16	16	2 ICLK		section 17
0008 2153h	DMAC5	DMA Interrupt Setting Register	DMINT	8	8	2 ICLK		section 17
0008 2154h	DMAC5	DMA Address Mode Register	DMAMD	16	16	2 ICLK		section 17
0008 215Ch	DMAC5	DMA Transfer Enable Register	DMCNT	8	8	2 ICLK		section 17
0008 215Dh	DMAC5	DMA Software Start Register	DMREQ	8	8	2 ICLK		section 17
0008 215Eh	DMAC5	DMA Status Register	DMSTS	8	8	2 ICLK		section 17
0008 215Fh	DMAC5	DMA Request Source Flag Control Register	DMCSL	8	8	2 ICLK		section 17
0008 2180h	DMAC6	DMA Source Address Register	DMSAR	32	32	2 ICLK		section 17
0008 2184h	DMAC6	DMA Destination Address Register	DMDAR	32	32	2 ICLK		section 17
0008 2188h	DMAC6	DMA Transfer Count Register	DMCRA	32	32	2 ICLK		section 17
0008 218Ch	DMAC6	DMA Block Transfer Count Register	DMCRB	16	16	2 ICLK		section 17
0008 2190h	DMAC6	DMA Transfer Mode Register	DMTMD	16	16	2 ICLK		section 17
0008 2193h	DMAC6	DMA Interrupt Setting Register	DMINT	8	8	2 ICLK		section 17
0008 2194h	DMAC6	DMA Address Mode Register	DMAMD	16	16	2 ICLK		section 17
0008 219Ch	DMAC6	DMA Transfer Enable Register	DMCNT	8	8	2 ICLK		section 17
0008 219Dh	DMAC6	DMA Software Start Register	DMREQ	8	8	2 ICLK		section 17
0008 219Eh	DMAC6	DMA Status Register	DMSTS	8	8	2 ICLK		section 17
0008 219Fh	DMAC6	DMA Request Source Flag Control Register	DMCSL	8	8	2 ICLK		section 17
0008 21C0h	DMAC7	DMA Source Address Register	DMSAR	32	32	2 ICLK		section 17
0008 21C4h	DMAC7	DMA Destination Address Register	DMDAR	32	32	2 ICLK		section 17
0008 21C8h	DMAC7	DMA Transfer Count Register	DMCRA	32	32	2 ICLK		section 17
0008 21CCh	DMAC7	DMA Block Transfer Count Register	DMCRB	16	16	2 ICLK		section 17
0008 21D0h	DMAC7	DMA Transfer Mode Register	DMTMD	16	16	2 ICLK		section 17
0008 21D3h	DMAC7	DMA Interrupt Setting Register	DMINT	8	8	2 ICLK		section 17
0008 21D4h	DMAC7	DMA Address Mode Register	DMAMD	16	16	2 ICLK		section 17
0008 21DCh	DMAC7	DMA Transfer Enable Register	DMCNT	8	8	2 ICLK		section 17
0008 21DDh	DMAC7	DMA Software Start Register	DMREQ	8	8	2 ICLK		section 17
0008 21DEh	DMAC7	DMA Status Register	DMSTS	8	8	2 ICLK		section 17
0008 21DFh	DMAC7	DMA Request Source Flag Control Register	DMCSL	8	8	2 ICLK		section 17
0008 2200h	DMAC	DMAC Module Start Register	DMAST	8	8	2 ICLK		section 17
0008 2204h	DMAC	DMAC74 Interrupt Status Monitor Register	DMIST	8	8	2 ICLK		section 17
0008 2400h	DTC	DTC Control Register	DTCCR	8	8	2 ICLK		section 18
0008 2404h	DTC	DTC Vector Base Register	DTCVBR	32	32	2 ICLK		section 18
0008 2408h	DTC	DTC Address Mode Register	DTCADMOD	8	8	2 ICLK		section 18

Table 5.1 List of I/O Registers (Address Order) (4 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 240Ch	DTC	DTC Module Start Register	DTCST	8	8	2 ICLK		section 18
0008 240Eh	DTC	DTC Status Register	DTCSTS	16	16	2 ICLK		section 18
0008 2410h	DTC	DTC Index Table Base Register	DTCIBR	32	32	2 ICLK		section 18
0008 2414h	DTC	DTC Operation Register	DTCOR	8	8	2 ICLK		section 18
0008 2416h	DTC	DTC Sequence Transfer Enable Register	DTCSQE	16	16	2 ICLK		section 18
0008 2418h	DTC	DTC Address Displacement Register	DTCDISP	32	32	2 ICLK		section 18
0008 6400h	MPU	Region-0 Start Page Number Register	RSPAGE0	32	32	1 ICLK		section 16
0008 6404h	MPU	Region-0 End Page Number Register	REPAGE0	32	32	1 ICLK		section 16
0008 6408h	MPU	Region-1 Start Page Number Register	RSPAGE1	32	32	1 ICLK		section 16
0008 640Ch	MPU	Region-1 End Page Number Register	REPAGE1	32	32	1 ICLK		section 16
0008 6410h	MPU	Region-2 Start Page Number Register	RSPAGE2	32	32	1 ICLK		section 16
0008 6414h	MPU	Region-2 End Page Number Register	REPAGE2	32	32	1 ICLK		section 16
0008 6418h	MPU	Region-3 Start Page Number Register	RSPAGE3	32	32	1 ICLK		section 16
0008 641Ch	MPU	Region-3 End Page Number Register	REPAGE3	32	32	1 ICLK		section 16
0008 6420h	MPU	Region-4 Start Page Number Register	RSPAGE4	32	32	1 ICLK		section 16
0008 6424h	MPU	Region-4 End Page Number Register	REPAGE4	32	32	1 ICLK		section 16
0008 6428h	MPU	Region-5 Start Page Number Register	RSPAGE5	32	32	1 ICLK		section 16
0008 642Ch	MPU	Region-5 End Page Number Register	REPAGE5	32	32	1 ICLK		section 16
0008 6430h	MPU	Region-6 Start Page Number Register	RSPAGE6	32	32	1 ICLK		section 16
0008 6434h	MPU	Region-6 End Page Number Register	REPAGE6	32	32	1 ICLK		section 16
0008 6438h	MPU	Region-7 Start Page Number Register	RSPAGE7	32	32	1 ICLK		section 16
0008 643Ch	MPU	Region-7 End Page Number Register	REPAGE7	32	32	1 ICLK		section 16
0008 6500h	MPU	Memory-Protection Enable Register	MPEN	32	32	1 ICLK		section 16
0008 6504h	MPU	Background Access Control Register	MPBAC	32	32	1 ICLK		section 16
0008 6508h	MPU	Memory-Protection Error Status-Clearing Register	MPECLR	32	32	1 ICLK		section 16
0008 650Ch	MPU	Memory-Protection Error Status Register	MPESTS	32	32	1 ICLK		section 16
0008 6514h	MPU	Data Memory-Protection Error Address Register	MPDEA	32	32	1 ICLK		section 16
0008 6520h	MPU	Region Search Address Register	MPSA	32	32	1 ICLK		section 16
0008 6524h	MPU	Region Search Operation Register	MPOPS	16	16	1 ICLK		section 16
0008 6526h	MPU	Region Invalidation Operation Register	MPOPI	16	16	1 ICLK		section 16
0008 6528h	MPU	Instruction-Hit Region Register	MHITI	32	32	1 ICLK		section 16
0008 652Ch	MPU	Data-Hit Region Register	MHITD	32	32	1 ICLK		section 16
0008 7010h to 0008 70FFh	ICU	Interrupt Request Register 016 to Interrupt Request Register 255	IR016 to IR255	8	8	2 ICLK		section 14
0008 711Ah to 0008 71FFh	ICU	DTC Transfer Request Enable Register 026 to DTC Transfer Request Enable Register 255	DTCER026 to DTCER255	8	8	2 ICLK		section 14
0008 7202h to 0008 721Fh	ICU	Interrupt Request Enable Register 02 to Interrupt Request Enable Register 1F	IER02 to IER1F	8	8	2 ICLK		section 14
0008 72E0h	ICU	Software Interrupt Generation Register	SWINTR	8	8	2 ICLK		section 14
0008 72E1h	ICU	Software Interrupt 2 Generation Register	SWINT2R	8	8	2 ICLK		section 14
0008 72F0h	ICU	Fast Interrupt Set Register	FIR	16	16	2 ICLK		section 14
0008 7300h to 0008 73FFh	ICU	Interrupt Source Priority Register 000 to Interrupt Source Priority Register 255	IPR000 to IPR255	8	8	2 ICLK		section 14
0008 7400h	ICU	DMAC Trigger Select Register 0	DMRSR0	8	8	2 ICLK		section 14
0008 7404h	ICU	DMAC Trigger Select Register 1	DMRSR1	8	8	2 ICLK		section 14
0008 7408h	ICU	DMAC Trigger Select Register 2	DMRSR2	8	8	2 ICLK		section 14
0008 740Ch	ICU	DMAC Trigger Select Register 3	DMRSR3	8	8	2 ICLK		section 14
0008 7410h	ICU	DMAC Trigger Select Register 4	DMRSR4	8	8	2 ICLK		section 14
0008 7414h	ICU	DMAC Trigger Select Register 5	DMRSR5	8	8	2 ICLK		section 14
0008 7418h	ICU	DMAC Trigger Select Register 6	DMRSR6	8	8	2 ICLK		section 14
0008 741Ch	ICU	DMAC Trigger Select Register 7	DMRSR7	8	8	2 ICLK		section 14

Table 5.1 List of I/O Registers (Address Order) (5 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 7500h to 0008 750Fh	ICU	IRQ Control Register 0 to IRQ Control Register 15	IRQCR0 to IRQCR15	8	8	2 ICLK		section 14
0008 7520h	ICU	IRQ Pin Digital Filter Enable Register 0	IRQFLTE0	8	8	2 ICLK		section 14
0008 7521h	ICU	IRQ Pin Digital Filter Enable Register 1	IRQFLTE1	8	8	2 ICLK		section 14
0008 7528h	ICU	IRQ Pin Digital Filter Setting Register 0	IRQFLTC0	16	16	2 ICLK		section 14
0008 752Ah	ICU	IRQ Pin Digital Filter Setting Register 1	IRQFLTC1	16	16	2 ICLK		section 14
0008 7580h	ICU	Non-Maskable Interrupt Status Register	NMISR	8	8	2 ICLK		section 14
0008 7581h	ICU	Non-Maskable Interrupt Enable Register	NMIER	8	8	2 ICLK		section 14
0008 7582h	ICU	Non-Maskable Interrupt Status Clear Register	NMICLR	8	8	2 ICLK		section 14
0008 7583h	ICU	NMI Pin Interrupt Control Register	NMICR	8	8	2 ICLK		section 14
0008 7590h	ICU	NMI Pin Digital Filter Enable Register	NMIFLTE	8	8	2 ICLK		section 14
0008 7594h	ICU	NMI Pin Digital Filter Setting Register	NMIFLTC	8	8	2 ICLK		section 14
0008 7630h	ICU	Group BL0 Interrupt Request Register	GRPBL0	32	32	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 7634h	ICU	Group BL1 Interrupt Request Register	GRPBL1	32	32	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 7638h	ICU	Group BL2 Interrupt Request Register	GRPBL2	32	32	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 7670h	ICU	Group BL0 Interrupt Request Enable Register	GENBL0	32	32	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 7674h	ICU	Group BL1 Interrupt Request Enable Register	GENBL1	32	32	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 7678h	ICU	Group BL2 Interrupt Request Enable Register	GENBL2	32	32	2 ICLK to 1 PCLKB	2 ICLK	section 14
0008 7830h	ICU	Group AL0 Interrupt Request Register	GRPAL0	32	32	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 7834h	ICU	Group AL1 Interrupt Request Register	GRPAL1	32	32	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 7870h	ICU	Group AL0 Interrupt Request Enable Register	GENAL0	32	32	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 7874h	ICU	Group AL1 Interrupt Request Enable Register	GENAL1	32	32	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 7900h	ICU	Software Configurable Interrupt A Request Register 0	PIAR0	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 7901h	ICU	Software Configurable Interrupt A Request Register 1	PIAR1	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 7902h	ICU	Software Configurable Interrupt A Request Register 2	PIAR2	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 7903h	ICU	Software Configurable Interrupt A Request Register 3	PIAR3	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 7904h	ICU	Software Configurable Interrupt A Request Register 4	PIAR4	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 7905h	ICU	Software Configurable Interrupt A Request Register 5	PIAR5	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 7906h	ICU	Software Configurable Interrupt A Request Register 6	PIAR6	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 7907h	ICU	Software Configurable Interrupt A Request Register 7	PIAR7	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 7908h	ICU	Software Configurable Interrupt A Request Register 8	PIAR8	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 7909h	ICU	Software Configurable Interrupt A Request Register 9	PIAR9	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 790Ah	ICU	Software Configurable Interrupt A Request Register A	PIARA	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 790Bh	ICU	Software Configurable Interrupt A Request Register B	PIARB	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 790Ch	ICU	Software Configurable Interrupt A Request Register C	PIARC	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 790Dh	ICU	Software Configurable Interrupt A Request Register D	PIARD	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 790Eh	ICU	Software Configurable Interrupt A Request Register E	PIARE	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14

Table 5.1 List of I/O Registers (Address Order) (6 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 790Fh	ICU	Software Configurable Interrupt A Request Register F	PIARF	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 7912h	ICU	Software Configurable Interrupt A Request Register 12	PIAR12	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 7913h	ICU	Software Configurable Interrupt A Request Register 13	PIAR13	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 7914h	ICU	Software Configurable Interrupt A Request Register 14	PIAR14	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79D0h	ICU	Software Configurable Interrupt A Source Select Register 208	SLIAR208	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79D1h	ICU	Software Configurable Interrupt A Source Select Register 209	SLIAR209	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79D2h	ICU	Software Configurable Interrupt A Source Select Register 210	SLIAR210	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79D3h	ICU	Software Configurable Interrupt A Source Select Register 211	SLIAR211	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79D4h	ICU	Software Configurable Interrupt A Source Select Register 212	SLIAR212	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79D5h	ICU	Software Configurable Interrupt A Source Select Register 213	SLIAR213	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79D6h	ICU	Software Configurable Interrupt A Source Select Register 214	SLIAR214	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79D7h	ICU	Software Configurable Interrupt A Source Select Register 215	SLIAR215	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79D8h	ICU	Software Configurable Interrupt A Source Select Register 216	SLIAR216	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79D9h	ICU	Software Configurable Interrupt A Source Select Register 217	SLIAR217	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79DAh	ICU	Software Configurable Interrupt A Source Select Register 218	SLIAR218	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79DBh	ICU	Software Configurable Interrupt A Source Select Register 219	SLIAR219	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79DCh	ICU	Software Configurable Interrupt A Source Select Register 220	SLIAR220	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79DDh	ICU	Software Configurable Interrupt A Source Select Register 221	SLIAR221	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79DEh	ICU	Software Configurable Interrupt A Source Select Register 222	SLIAR222	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79DFh	ICU	Software Configurable Interrupt A Source Select Register 223	SLIAR223	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79E0h	ICU	Software Configurable Interrupt A Source Select Register 224	SLIAR224	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79E1h	ICU	Software Configurable Interrupt A Source Select Register 225	SLIAR225	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79E2h	ICU	Software Configurable Interrupt A Source Select Register 226	SLIAR226	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79E3h	ICU	Software Configurable Interrupt A Source Select Register 227	SLIAR227	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79E4h	ICU	Software Configurable Interrupt A Source Select Register 228	SLIAR228	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79E5h	ICU	Software Configurable Interrupt A Source Select Register 229	SLIAR229	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79E6h	ICU	Software Configurable Interrupt A Source Select Register 230	SLIAR230	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79E7h	ICU	Software Configurable Interrupt A Source Select Register 231	SLIAR231	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79E8h	ICU	Software Configurable Interrupt A Source Select Register 232	SLIAR232	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79E9h	ICU	Software Configurable Interrupt A Source Select Register 233	SLIAR233	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79EAh	ICU	Software Configurable Interrupt A Source Select Register 234	SLIAR234	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79EBh	ICU	Software Configurable Interrupt A Source Select Register 235	SLIAR235	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14

Table 5.1 List of I/O Registers (Address Order) (7 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 79ECh	ICU	Software Configurable Interrupt A Source Select Register 236	SLIAR236	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79EDh	ICU	Software Configurable Interrupt A Source Select Register 237	SLIAR237	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79EEh	ICU	Software Configurable Interrupt A Source Select Register 238	SLIAR238	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79EFh	ICU	Software Configurable Interrupt A Source Select Register 239	SLIAR239	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79F0h	ICU	Software Configurable Interrupt A Source Select Register 240	SLIAR240	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79F1h	ICU	Software Configurable Interrupt A Source Select Register 241	SLIAR241	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79F2h	ICU	Software Configurable Interrupt A Source Select Register 242	SLIAR242	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79F3h	ICU	Software Configurable Interrupt A Source Select Register 243	SLIAR243	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79F4h	ICU	Software Configurable Interrupt A Source Select Register 244	SLIAR244	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79F5h	ICU	Software Configurable Interrupt A Source Select Register 245	SLIAR245	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79F6h	ICU	Software Configurable Interrupt A Source Select Register 246	SLIAR246	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79F7h	ICU	Software Configurable Interrupt A Source Select Register 247	SLIAR247	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79F8h	ICU	Software Configurable Interrupt A Source Select Register 248	SLIAR248	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79F9h	ICU	Software Configurable Interrupt A Source Select Register 249	SLIAR249	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79FAh	ICU	Software Configurable Interrupt A Source Select Register 250	SLIAR250	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79FBh	ICU	Software Configurable Interrupt A Source Select Register 251	SLIAR251	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79FCh	ICU	Software Configurable Interrupt A Source Select Register 252	SLIAR252	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79FDh	ICU	Software Configurable Interrupt A Source Select Register 253	SLIAR253	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79FEh	ICU	Software Configurable Interrupt A Source Select Register 254	SLIAR254	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 79FFh	ICU	Software Configurable Interrupt A Source Select Register 255	SLIAR255	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 14
0008 7A00h	ICU	Software Configurable Interrupt Source Select Register Write Protect Register	SLIPRCR	8	8	2 ICLK to 1 PCLKA/B	2 ICLK	section 14
0008 8000h	CMT	Compare Match Timer Start Register 0	CMSTR0	16	16	2, 3 PCLKB	2 ICLK	section 28
0008 8002h	CMT0	Compare Match Timer Control Register	CMCR	16	16	2, 3 PCLKB	2 ICLK	section 28
0008 8004h	CMT0	Compare Match Counter	CMCNT	16	16	2, 3 PCLKB	2 ICLK	section 28
0008 8006h	CMT0	Compare Match Constant Register	CMCOR	16	16	2, 3 PCLKB	2 ICLK	section 28
0008 8008h	CMT1	Compare Match Timer Control Register	CMCR	16	16	2, 3 PCLKB	2 ICLK	section 28
0008 800Ah	CMT1	Compare Match Counter	CMCNT	16	16	2, 3 PCLKB	2 ICLK	section 28
0008 800Ch	CMT1	Compare Match Constant Register	CMCOR	16	16	2, 3 PCLKB	2 ICLK	section 28
0008 8010h	CMT	Compare Match Timer Start Register 1	CMSTR1	16	16	2, 3 PCLKB	2 ICLK	section 28
0008 8012h	CMT2	Compare Match Timer Control Register	CMCR	16	16	2, 3 PCLKB	2 ICLK	section 28
0008 8014h	CMT2	Compare Match Counter	CMCNT	16	16	2, 3 PCLKB	2 ICLK	section 28
0008 8016h	CMT2	Compare Match Constant Register	CMCOR	16	16	2, 3 PCLKB	2 ICLK	section 28
0008 8018h	CMT3	Compare Match Timer Control Register	CMCR	16	16	2, 3 PCLKB	2 ICLK	section 28
0008 801Ah	CMT3	Compare Match Counter	CMCNT	16	16	2, 3 PCLKB	2 ICLK	section 28
0008 801Ch	CMT3	Compare Match Constant Register	CMCOR	16	16	2, 3 PCLKB	2 ICLK	section 28
0008 8020h	WDT	WDT Refresh Register	WDTRR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 8022h	WDT	WDT Control Register	WDTCR	16	16	2, 3 PCLKB	2 ICLK	section 30
0008 8024h	WDT	WDT Status Register	WDTSR	16	16	2, 3 PCLKB	2 ICLK	section 30

Table 5.1 List of I/O Registers (Address Order) (8 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 8026h	WDT	WDT Reset Control Register	WDTRCR	8	8	2, 3 PCLKB	2 ICLK	section 30
0008 8030h	IWDT	IWDT Refresh Register	IWDTRR	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 8032h	IWDT	IWDT Control Register	IWDTCR	16	16	2, 3 PCLKB	2 ICLK	section 31
0008 8034h	IWDT	IWDT Status Register	IWDTSR	16	16	2, 3 PCLKB	2 ICLK	section 31
0008 8036h	IWDT	IWDT Reset Control Register	IWDTRCR	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 8038h	IWDT	IWDT Count Stop Control Register	IWDTCSTPR	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 8040h	DA	D/A Data Register 0	DADR0	16	16	2, 3 PCLKB	2 ICLK	section 43
0008 8042h	DA	D/A Data Register 1	DADR1	16	16	2, 3 PCLKB	2 ICLK	section 43
0008 8044h	DA	D/A Control Register	DACR	8	8	2, 3 PCLKB	2 ICLK	section 43
0008 8045h	DA	Data Register Format Select Register	DADPR	8	8	2, 3 PCLKB	2 ICLK	section 43
0008 8046h	DA	D/A A/D Synchronous Start Control Register	DAADSCR	8	8	2, 3 PCLKB	2 ICLK	section 43
0008 8049h	DA	D/A Destination Select Register	DADSELR	8	8	2, 3 PCLKB	2 ICLK	section 43
0008 8200h	TMR0	Timer Control Register	TCR	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 8201h	TMR1	Timer Control Register	TCR	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 8202h	TMR0	Timer Control/Status Register	TCSR	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 8203h	TMR1	Timer Control/Status Register	TCSR	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 8204h	TMR0	Time Constant Register A	TCORA	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 8204h	TMR01	Time Constant Register A	TCORA	16	16	2, 3 PCLKB	2 ICLK	section 27
0008 8205h	TMR1	Time Constant Register A	TCORA	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 8206h	TMR0	Time Constant Register B	TCORB	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 8206h	TMR01	Time Constant Register B	TCORB	16	16	2, 3 PCLKB	2 ICLK	section 27
0008 8207h	TMR1	Time Constant Register B	TCORB	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 8208h	TMR0	Timer Counter	TCNT	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 8208h	TMR01	Timer Counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	section 27
0008 8209h	TMR1	Timer Counter	TCNT	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 820Ah	TMR0	Timer Counter Control Register	TCCR	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 820Ah	TMR01	Timer Counter Control Register	TCCR	16	16	2, 3 PCLKB	2 ICLK	section 27
0008 820Bh	TMR1	Timer Counter Control Register	TCCR	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 820Ch	TMR0	Timer Counter Start Register	TCSTR	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 820Dh	TMR1	Timer Counter Start Register	TCSTR	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 8210h	TMR2	Timer Control Register	TCR	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 8211h	TMR3	Timer Control Register	TCR	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 8212h	TMR2	Timer Control/Status Register	TCSR	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 8213h	TMR3	Timer Control/Status Register	TCSR	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 8214h	TMR2	Time Constant Register A	TCORA	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 8214h	TMR23	Time Constant Register A	TCORA	16	16	2, 3 PCLKB	2 ICLK	section 27
0008 8215h	TMR3	Time Constant Register A	TCORA	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 8216h	TMR2	Time Constant Register B	TCORB	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 8216h	TMR23	Time Constant Register B	TCORB	16	16	2, 3 PCLKB	2 ICLK	section 27
0008 8217h	TMR3	Time Constant Register B	TCORB	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 8218h	TMR2	Timer Counter	TCNT	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 8218h	TMR23	Timer Counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	section 27
0008 8219h	TMR3	Timer Counter	TCNT	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 821Ah	TMR2	Timer Counter Control Register	TCCR	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 821Ah	TMR23	Timer Counter Control Register	TCCR	16	16	2, 3 PCLKB	2 ICLK	section 27
0008 821Bh	TMR3	Timer Counter Control Register	TCCR	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 821Ch	TMR2	Timer Counter Start Register	TCSTR	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 821Dh	TMR3	Timer Counter Start Register	TCSTR	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 8220h	TMR4	Timer Control Register	TCR	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 8221h	TMR5	Timer Control Register	TCR	8	8	2, 3 PCLKB	2 ICLK	section 27

Table 5.1 List of I/O Registers (Address Order) (9 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 8222h	TMR4	Timer Control/Status Register	TCSR	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 8223h	TMR5	Timer Control/Status Register	TCSR	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 8224h	TMR4	Time Constant Register A	TCORA	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 8224h	TMR45	Time Constant Register A	TCORA	16	16	2, 3 PCLKB	2 ICLK	section 27
0008 8225h	TMR5	Time Constant Register A	TCORA	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 8226h	TMR4	Time Constant Register B	TCORB	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 8226h	TMR45	Time Constant Register B	TCORB	16	16	2, 3 PCLKB	2 ICLK	section 27
0008 8227h	TMR5	Time Constant Register B	TCORB	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 8228h	TMR4	Timer Counter	TCNT	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 8228h	TMR45	Timer Counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	section 27
0008 8229h	TMR5	Timer Counter	TCNT	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 822Ah	TMR4	Timer Counter Control Register	TCCR	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 822Ah	TMR45	Timer Counter Control Register	TCCR	16	16	2, 3 PCLKB	2 ICLK	section 27
0008 822Bh	TMR5	Timer Counter Control Register	TCCR	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 8230h	TMR6	Timer Control Register	TCR	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 8231h	TMR7	Timer Control Register	TCR	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 8232h	TMR6	Timer Control/Status Register	TCSR	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 8233h	TMR7	Timer Control/Status Register	TCSR	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 8234h	TMR6	Time Constant Register A	TCORA	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 8234h	TMR67	Time Constant Register A	TCORA	16	16	2, 3 PCLKB	2 ICLK	section 27
0008 8235h	TMR7	Time Constant Register A	TCORA	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 8236h	TMR6	Time Constant Register B	TCORB	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 8236h	TMR67	Time Constant Register B	TCORB	16	16	2, 3 PCLKB	2 ICLK	section 27
0008 8237h	TMR7	Time Constant Register B	TCORB	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 8238h	TMR6	Timer Counter	TCNT	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 8238h	TMR67	Timer Counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	section 27
0008 8239h	TMR7	Timer Counter	TCNT	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 823Ah	TMR6	Timer Counter Control Register	TCCR	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 823Ah	TMR67	Timer Counter Control Register	TCCR	16	16	2, 3 PCLKB	2 ICLK	section 27
0008 823Bh	TMR7	Timer Counter Control Register	TCCR	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 8280h	CRC	CRC Control Register	CRCCR	8	8	2, 3 PCLKB	2 ICLK	section 39
0008 8284h	CRC	CRC Data Input Register	CRCDIR	32	8, 32	2, 3 PCLKB	2 ICLK	section 39
0008 8288h	CRC	CRC Data Output Register	CRCDOR	32	8, 16, 32	2, 3 PCLKB	2 ICLK	section 39
0008 8300h	RIIC0	I²C-bus Control Register 1	ICCR1	8	8	2, 3 PCLKB	2 ICLK	section 34
0008 8301h	RIIC0	I²C-bus Control Register 2	ICCR2	8	8	2, 3 PCLKB	2 ICLK	section 34
0008 8302h	RIIC0	I²C-bus Mode Register 1	ICMR1	8	8	2, 3 PCLKB	2 ICLK	section 34
0008 8303h	RIIC0	I²C-bus Mode Register 2	ICMR2	8	8	2, 3 PCLKB	2 ICLK	section 34
0008 8304h	RIIC0	I²C-bus Mode Register 3	ICMR3	8	8	2, 3 PCLKB	2 ICLK	section 34
0008 8305h	RIIC0	I²C-bus Function Enable Register	ICFER	8	8	2, 3 PCLKB	2 ICLK	section 34
0008 8306h	RIIC0	I²C-bus Status Enable Register	ICSER	8	8	2, 3 PCLKB	2 ICLK	section 34
0008 8307h	RIIC0	I²C-bus Interrupt Enable Register	ICIER	8	8	2, 3 PCLKB	2 ICLK	section 34
0008 8308h	RIIC0	I²C-bus Status Register 1	ICSR1	8	8	2, 3 PCLKB	2 ICLK	section 34
0008 8309h	RIIC0	I²C-bus Status Register 2	ICSR2	8	8	2, 3 PCLKB	2 ICLK	section 34
0008 830Ah	RIIC0	Slave Address Register L0	SARL0	8	8	2, 3 PCLKB	2 ICLK	section 34
0008 830Bh	RIIC0	Slave Address Register U0	SARU0	8	8	2, 3 PCLKB	2 ICLK	section 34
0008 830Ch	RIIC0	Slave Address Register L1	SARL1	8	8	2, 3 PCLKB	2 ICLK	section 34
0008 830Dh	RIIC0	Slave Address Register U1	SARU1	8	8	2, 3 PCLKB	2 ICLK	section 34
0008 830Eh	RIIC0	Slave Address Register L2	SARL2	8	8	2, 3 PCLKB	2 ICLK	section 34
0008 830Fh	RIIC0	Slave Address Register U2	SARU2	8	8	2, 3 PCLKB	2 ICLK	section 34
0008 8310h	RIIC0	I²C-bus Bit Rate Low-Level Register	ICBRL	8	8	2, 3 PCLKB	2 ICLK	section 34

Table 5.1 List of I/O Registers (Address Order) (10 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 8311h	RIIC0	I²C-bus Bit Rate High-Level Register	ICBRH	8	8	2, 3 PCLKB	2 ICLK	section 34
0008 8312h	RIIC0	I²C-bus Transmit Data Register	ICDRT	8	8	2, 3 PCLKB	2 ICLK	section 34
0008 8313h	RIIC0	I²C-bus Receive Data Register	ICDRR	8	8	2, 3 PCLKB	2 ICLK	section 34
0008 9000h	S12AD	A/D Control Register	ADCSR	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9004h	S12AD	A/D Channel Select Register A0	ADANSA0	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9008h	S12AD	A/D-Converted Value Addition/Average Function Channel Select Register 0	ADADS0	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 900Ch	S12AD	A/D-Converted Value Addition/Average Count Select Register	ADADC	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 900Eh	S12AD	A/D Control Extended Register	ADCER	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9010h	S12AD	A/D Conversion Start Trigger Select Register	ADSTRGR	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9014h	S12AD	A/D Channel Select Register B0	ADANSB0	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9018h	S12AD	A/D Data Duplication Register	ADDBLDR	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 901Eh	S12AD	A/D Self-Diagnosis Data Register	ADRD	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9020h	S12AD	A/D Data Register 0	ADDR0	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9022h	S12AD	A/D Data Register 1	ADDR1	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9024h	S12AD	A/D Data Register 2	ADDR2	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9026h	S12AD	A/D Data Register 3	ADDR3	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9028h	S12AD	A/D Data Register 4	ADDR4	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 902Ah	S12AD	A/D Data Register 5	ADDR5	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 902Ch	S12AD	A/D Data Register 6	ADDR6	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9066h	S12AD	A/D Sample-and-Hold Circuit Control Register	ADSHCR	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 907Ah	S12AD	A/D Disconnection Detection Control Register	ADDISCR	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 907Ch	S12AD	A/D Sample-and-Hold Operating Mode Select Register	ADSHMSR	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 907Dh	S12AD	A/D Event Link Control Register	ADELCCR	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 9080h	S12AD	A/D Group Scan Priority Control Register	ADGSPCR	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9084h	S12AD	A/D Data Duplication Register A	ADDBLDRA	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9086h	S12AD	A/D Data Duplication Register B	ADDBLDRB	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 908Ch	S12AD	A/D Comparison Function Window A/B Status Monitoring Register	ADWINMON	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 9090h	S12AD	A/D Comparison Function Control Register	ADCMPCR	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9094h	S12AD	A/D Comparison Function Window A Channel Select Register 0	ADCMPANSR 0	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9098h	S12AD	A/D Comparison Function Window A Comparison Condition Setting Register 0	ADCMPLR0	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 909Ch	S12AD	A/D Comparison Function Window A Lower Level Setting Register	ADCMPDR0	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 909Eh	S12AD	A/D Comparison Function Window A Upper Level Setting Register	ADCMPDR1	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 90A0h	S12AD	A/D Comparison Function Window A Channel Status Register 0	ADCMPSR0	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 90A6h	S12AD	A/D Comparison Function Window B Channel Select Register	ADCMPBNSR	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 90A8h	S12AD	A/D Comparison Function Window B Lower Level Setting Register	ADWINLLB	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 90AAh	S12AD	A/D Comparison Function Window B Upper Level Setting Register	ADWINULB	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 90ACh	S12AD	A/D Comparison Function Window B Channel Status Register	ADCMPBSR	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 90D4h	S12AD	A/D Channel Select Register C0	ADANSC0	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 90D9h	S12AD	A/D Group C Trigger Select Register	ADGCTRGR	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 90DCh	S12AD	A/D Group C Trigger Select Register 2	ADGCTRGR2	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 90E0h	S12AD	A/D Sampling State Register 0	ADSSTR0	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 90E1h	S12AD	A/D Sampling State Register 1	ADSSTR1	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 90E2h	S12AD	A/D Sampling State Register 2	ADSSTR2	8	8	2, 3 PCLKB	2 ICLK	section 42

Table 5.1 List of I/O Registers (Address Order) (11 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 90E3h	S12AD	A/D Sampling State Register 3	ADSSTR3	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 90E4h	S12AD	A/D Sampling State Register 4	ADSSTR4	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 90E5h	S12AD	A/D Sampling State Register 5	ADSSTR5	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 90E6h	S12AD	A/D Sampling State Register 6	ADSSTR6	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 91A0h	S12AD	A/D Programmable Gain Amplifier Control Register	ADPGACR	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 91A2h	S12AD	A/D Programmable Gain Amplifier Gain Setting Register 0	ADPGAGS0	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 91C0h	S12AD	A/D Channel Conversion Order Setting Register 0	ADSCS0	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 91C1h	S12AD	A/D Channel Conversion Order Setting Register 1	ADSCS1	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 91C2h	S12AD	A/D Channel Conversion Order Setting Register 2	ADSCS2	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 91C3h	S12AD	A/D Channel Conversion Order Setting Register 3	ADSCS3	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 91C4h	S12AD	A/D Channel Conversion Order Setting Register 4	ADSCS4	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 91C5h	S12AD	A/D Channel Conversion Order Setting Register 5	ADSCS5	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 91C6h	S12AD	A/D Channel Conversion Order Setting Register 6	ADSCS6	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 9200h	S12AD1	A/D Control Register	ADCSR	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9204h	S12AD1	A/D Channel Select Register A0	ADANSA0	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9208h	S12AD1	A/D-Converted Value Addition/Average Function Channel Select Register 0	ADADS0	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 920Ch	S12AD1	A/D-Converted Value Addition/Average Count Select Register	ADADC	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 920Eh	S12AD1	A/D Control Extended Register	ADCER	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9210h	S12AD1	A/D Conversion Start Trigger Select Register	ADSTRGR	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9214h	S12AD1	A/D Channel Select Register B0	ADANSB0	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9218h	S12AD1	A/D Data Duplication Register	ADDBLDR	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 921Eh	S12AD1	A/D Self-Diagnosis Data Register	ADRD	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9220h	S12AD1	A/D Data Register 0	ADDR0	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9222h	S12AD1	A/D Data Register 1	ADDR1	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9224h	S12AD1	A/D Data Register 2	ADDR2	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9226h	S12AD1	A/D Data Register 3	ADDR3	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9266h	S12AD1	A/D Sample-and-Hold Circuit Control Register	ADSHCR	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 927Ah	S12AD1	A/D Disconnection Detection Control Register	ADDISCR	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 927Ch	S12AD1	A/D Sample-and-Hold Operating Mode Select Register	ADSHMSR	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 927Dh	S12AD1	A/D Event Link Control Register	ADELCCR	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 9280h	S12AD1	A/D Group Scan Priority Control Register	ADGSPCR	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9284h	S12AD1	A/D Data Duplication Register A	ADDBLDRA	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9286h	S12AD1	A/D Data Duplication Register B	ADDBLDRB	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 928Ch	S12AD1	A/D Comparison Function Window A/B Status Monitoring Register	ADWINMON	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 9290h	S12AD1	A/D Comparison Function Control Register	ADCMPCR	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9294h	S12AD1	A/D Comparison Function Window A Channel Select Register 0	ADCMPANSR 0	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9298h	S12AD1	A/D Comparison Function Window A Comparison Condition Setting Register 0	ADCMPLR0	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 929Ch	S12AD1	A/D Comparison Function Window A Lower Level Setting Register	ADCMPDR0	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 929Eh	S12AD1	A/D Comparison Function Window A Upper Level Setting Register	ADCMPDR1	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 92A0h	S12AD1	A/D Comparison Function Window A Channel Status Register 0	ADCMPSR0	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 92A6h	S12AD1	A/D Comparison Function Window B Channel Select Register	ADCMPBNSR	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 92A8h	S12AD1	A/D Comparison Function Window B Lower Level Setting Register	ADWINLLB	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 92AAh	S12AD1	A/D Comparison Function Window B Upper Level Setting Register	ADWINULB	16	16	2, 3 PCLKB	2 ICLK	section 42

Table 5.1 List of I/O Registers (Address Order) (12 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 92ACh	S12AD1	A/D Comparison Function Window B Channel Status Register	ADCMPBSR	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 92D4h	S12AD1	A/D Channel Select Register C0	ADANSC0	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 92D9h	S12AD1	A/D Group C Trigger Select Register	ADGCTRGR	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 92DCh	S12AD1	A/D Group C Trigger Select Register 2	ADGCTRGR2	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 92E0h	S12AD1	A/D Sampling State Register 0	ADSSTR0	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 92E1h	S12AD1	A/D Sampling State Register 1	ADSSTR1	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 92E2h	S12AD1	A/D Sampling State Register 2	ADSSTR2	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 92E3h	S12AD1	A/D Sampling State Register 3	ADSSTR3	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 93A0h	S12AD1	A/D Programmable Gain Amplifier Control Register	ADPGACR	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 93A2h	S12AD1	A/D Programmable Gain Amplifier Gain Setting Register 0	ADPGAGS0	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 93C0h	S12AD1	A/D Channel Conversion Order Setting Register 0	ADSCS0	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 93C1h	S12AD1	A/D Channel Conversion Order Setting Register 1	ADSCS1	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 93C2h	S12AD1	A/D Channel Conversion Order Setting Register 2	ADSCS2	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 93C3h	S12AD1	A/D Channel Conversion Order Setting Register 3	ADSCS3	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 9400h	S12AD2	A/D Control Register	ADCSR	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9404h	S12AD2	A/D Channel Select Register A0	ADANSA0	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9406h	S12AD2	A/D Channel Select Register A1	ADANSA1	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9408h	S12AD2	A/D-Converted Value Addition/Average Function Channel Select Register 0	ADADS0	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 940Ah	S12AD2	A/D-Converted Value Addition/Average Function Channel Select Register 1	ADADS1	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 940Ch	S12AD2	A/D-Converted Value Addition/Average Count Select Register	ADADC	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 940Eh	S12AD2	A/D Control Extended Register	ADCER	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9410h	S12AD2	A/D Conversion Start Trigger Select Register	ADSTRGR	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9412h	S12AD2	A/D Conversion Extended Input Control Register	ADEXICR	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9414h	S12AD2	A/D Channel Select Register B0	ADANSB0	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9416h	S12AD2	A/D Channel Select Register B1	ADANSB1	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9418h	S12AD2	A/D Data Duplication Register	ADDBLDR	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 941Ah	S12AD2	A/D Temperature Sensor Data Register	ADTSDR	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 941Ch	S12AD2	A/D Internal Reference Voltage Data Register	ADOCDR	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 941Eh	S12AD2	A/D Self-Diagnosis Data Register	ADRD	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9420h	S12AD2	A/D Data Register 0	ADDR0	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9422h	S12AD2	A/D Data Register 1	ADDR1	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9424h	S12AD2	A/D Data Register 2	ADDR2	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9426h	S12AD2	A/D Data Register 3	ADDR3	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9428h	S12AD2	A/D Data Register 4	ADDR4	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 942Ah	S12AD2	A/D Data Register 5	ADDR5	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 942Ch	S12AD2	A/D Data Register 6	ADDR6	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 942Eh	S12AD2	A/D Data Register 7	ADDR7	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9430h	S12AD2	A/D Data Register 8	ADDR8	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9432h	S12AD2	A/D Data Register 9	ADDR9	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9434h	S12AD2	A/D Data Register 10	ADDR10	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9436h	S12AD2	A/D Data Register 11	ADDR11	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9440h	S12AD2	A/D Data Register 16	ADDR16	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9442h	S12AD2	A/D Data Register 17	ADDR17	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 947Ah	S12AD2	A/D Disconnection Detection Control Register	ADDISCR	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 947Dh	S12AD2	A/D Event Link Control Register	ADELCCR	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 9480h	S12AD2	A/D Group Scan Priority Control Register	ADGSPCR	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9484h	S12AD2	A/D Data Duplication Register A	ADDBLDRA	16	16	2, 3 PCLKB	2 ICLK	section 42

Table 5.1 List of I/O Registers (Address Order) (13 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 9486h	S12AD2	A/D Data Duplication Register B	ADDBLDRB	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 948Ch	S12AD2	A/D Comparison Function Window A/B Status Monitoring Register	ADWINMON	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 9490h	S12AD2	A/D Comparison Function Control Register	ADCMPCR	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9492h	S12AD2	A/D Comparison Function Window A Extended Input Select Register	ADCMPANSER	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 9493h	S12AD2	A/D Comparison Function Window A Extended Input Comparison Condition Setting Register	ADCMPLER	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 9494h	S12AD2	A/D Comparison Function Window A Channel Select Register 0	ADCMPANSR0	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9496h	S12AD2	A/D Comparison Function Window A Channel Select Register 1	ADCMPANSR1	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 9498h	S12AD2	A/D Comparison Function Window A Comparison Condition Setting Register 0	ADCMPLR0	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 949Ah	S12AD2	A/D Comparison Function Window A Comparison Condition Setting Register 1	ADCMPLR1	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 949Ch	S12AD2	A/D Comparison Function Window A Lower Level Setting Register	ADCMPDR0	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 949Eh	S12AD2	A/D Comparison Function Window A Upper Level Setting Register	ADCMPDR1	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 94A0h	S12AD2	A/D Comparison Function Window A Channel Status Register 0	ADCMPSR0	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 94A2h	S12AD2	A/D Comparison Function Window A Channel Status Register 1	ADCMPSR1	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 94A4h	S12AD2	A/D Comparison Function Window A Extended Input Channel Status Register	ADCMPSER	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 94A6h	S12AD2	A/D Comparison Function Window B Channel Select Register	ADCMPBNSR	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 94A8h	S12AD2	A/D Comparison Function Window B Lower Level Setting Register	ADWINLLB	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 94AAh	S12AD2	A/D Comparison Function Window B Upper Level Setting Register	ADWINULB	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 94ACh	S12AD2	A/D Comparison Function Window B Channel Status Register	ADCMPBSR	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 94D4h	S12AD2	A/D Channel Select Register C0	ADANSC0	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 94D6h	S12AD2	A/D Channel Select Register C1	ADANSC1	16	16	2, 3 PCLKB	2 ICLK	section 42
0008 94D8h	S12AD2	A/D Group C Extended Input Control Register	ADGCEXCR	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 94D9h	S12AD2	A/D Group C Trigger Select Register	ADGCTRGR	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 94DCh	S12AD2	A/D Group C Trigger Select Register 2	ADGCTRGR2	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 94DDh	S12AD2	A/D Sampling State Register L	ADSSTRL	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 94DEh	S12AD2	A/D Sampling State Register T	ADSSTRT	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 94DFh	S12AD2	A/D Sampling State Register O	ADSSTRO	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 94E0h	S12AD2	A/D Sampling State Register 0	ADSSTR0	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 94E1h	S12AD2	A/D Sampling State Register 1	ADSSTR1	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 94E2h	S12AD2	A/D Sampling State Register 2	ADSSTR2	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 94E3h	S12AD2	A/D Sampling State Register 3	ADSSTR3	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 94E4h	S12AD2	A/D Sampling State Register 4	ADSSTR4	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 94E5h	S12AD2	A/D Sampling State Register 5	ADSSTR5	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 94E6h	S12AD2	A/D Sampling State Register 6	ADSSTR6	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 94E7h	S12AD2	A/D Sampling State Register 7	ADSSTR7	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 94E8h	S12AD2	A/D Sampling State Register 8	ADSSTR8	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 94E9h	S12AD2	A/D Sampling State Register 9	ADSSTR9	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 94EAh	S12AD2	A/D Sampling State Register 10	ADSSTR10	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 94EBh	S12AD2	A/D Sampling State Register 11	ADSSTR11	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 95C0h	S12AD2	A/D Channel Conversion Order Setting Register 0	ADSCS0	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 95C1h	S12AD2	A/D Channel Conversion Order Setting Register 1	ADSCS1	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 95C2h	S12AD2	A/D Channel Conversion Order Setting Register 2	ADSCS2	8	8	2, 3 PCLKB	2 ICLK	section 42

Table 5.1 List of I/O Registers (Address Order) (14 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 95C3h	S12AD2	A/D Channel Conversion Order Setting Register 3	ADSCS3	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 95C4h	S12AD2	A/D Channel Conversion Order Setting Register 4	ADSCS4	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 95C5h	S12AD2	A/D Channel Conversion Order Setting Register 5	ADSCS5	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 95C6h	S12AD2	A/D Channel Conversion Order Setting Register 6	ADSCS6	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 95C7h	S12AD2	A/D Channel Conversion Order Setting Register 7	ADSCS7	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 95C8h	S12AD2	A/D Channel Conversion Order Setting Register 8	ADSCS8	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 95C9h	S12AD2	A/D Channel Conversion Order Setting Register 9	ADSCS9	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 95CAh	S12AD2	A/D Channel Conversion Order Setting Register 10	ADSCS10	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 95CBh	S12AD2	A/D Channel Conversion Order Setting Register 11	ADSCS11	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 95D0h	S12AD2	A/D Channel Conversion Order Setting Register 12	ADSCS12	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 95D1h	S12AD2	A/D Channel Conversion Order Setting Register 13	ADSCS13	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 95E2h	S12AD2	A/D Internal Reference Voltage Monitoring Circuit Enable Register	ADVMONCR	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 95E4h	S12AD2	A/D Internal Reference Voltage Monitoring Circuit Output Enable Register	ADVMONO	8	8	2, 3 PCLKB	2 ICLK	section 42
0008 A020h	SCI1	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A020h	SMCI1	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A021h	SCI1	Bit Rate Register	BRR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A022h	SCI1	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A022h	SMCI1	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A023h	SCI1	Transmit Data Register	TDR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A024h	SCI1	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A024h	SMCI1	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A025h	SCI1	Receive Data Register	RDR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A026h	SCI1	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A026h	SMCI1	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A027h	SCI1	Serial Extended Mode Register	SEMR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A028h	SCI1	Noise Filter Setting Register	SNFR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A029h	SCI1	I^2C Mode Register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A02Ah	SCI1	I^2C Mode Register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A02Bh	SCI1	I^2C Mode Register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A02Ch	SCI1	I^2C Status Register	SISR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A02Dh	SCI1	SPI Mode Register	SPMR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A02Eh	SCI1	Transmit Data Register H	TDRH	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A02Fh	SCI1	Transmit Data Register L	TDRL	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A02Eh	SCI1	Transmit Data Register HL	TDRHL	16	16	4, 5 PCLKB	2 ICLK	section 32
0008 A030h	SCI1	Receive Data Register H	RDRH	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A031h	SCI1	Receive Data Register L	RDRL	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A030h	SCI1	Receive Data Register HL	RDRHL	16	16	4, 5 PCLKB	2 ICLK	section 32
0008 A032h	SCI1	Modulation Duty Register	MDDR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A033h	SCI1	Data Comparison Control Register	DCCR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A03Ah	SCI1	Comparison Data Register H	CDR.H	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A03Bh	SCI1	Comparison Data Register L	CDR.L	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A03Ah	SCI1	Comparison Data Register	CDR	16	16	4, 5 PCLKB	2 ICLK	section 32
0008 A03Ch	SCI1	Serial Port Register	SPTR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A03Dh	SCI1	Transmit/Receive Timing Select Register	TMGR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0A0h	SCI5	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0A0h	SMCI5	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0A1h	SCI5	Bit Rate Register	BRR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0A2h	SCI5	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0A2h	SMCI5	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 32

Table 5.1 List of I/O Registers (Address Order) (15 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 A0A3h	SCI5	Transmit Data Register	TDR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0A4h	SCI5	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0A4h	SMCI5	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0A5h	SCI5	Receive Data Register	RDR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0A6h	SCI5	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0A6h	SMCI5	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0A7h	SCI5	Serial Extended Mode Register	SEMR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0A8h	SCI5	Noise Filter Setting Register	SNFR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0A9h	SCI5	I^2C Mode Register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0AAh	SCI5	I^2C Mode Register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0ABh	SCI5	I^2C Mode Register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0ACh	SCI5	I^2C Status Register	SISR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0ADh	SCI5	SPI Mode Register	SPMR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0AEh	SCI5	Transmit Data Register H	TDRH	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0AFh	SCI5	Transmit Data Register L	TDRL	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0AEh	SCI5	Transmit Data Register HL	TDRHL	16	16	4, 5 PCLKB	2 ICLK	section 32
0008 A0B0h	SCI5	Receive Data Register H	RDRH	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0B1h	SCI5	Receive Data Register L	RDRL	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0B0h	SCI5	Receive Data Register HL	RDRHL	16	16	4, 5 PCLKB	2 ICLK	section 32
0008 A0B2h	SCI5	Modulation Duty Register	MDDR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0B3h	SCI5	Data Comparison Control Register	DCCR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0BAh	SCI5	Comparison Data Register H	CDR.H	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0BBh	SCI5	Comparison Data Register L	CDR.L	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0BAh	SCI5	Comparison Data Register	CDR	16	16	4, 5 PCLKB	2 ICLK	section 32
0008 A0BCh	SCI5	Serial Port Register	SPTR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0BDh	SCI5	Transmit/Receive Timing Select Register	TMGR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0C0h	SCI6	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0C0h	SMCI6	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0C1h	SCI6	Bit Rate Register	BRR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0C2h	SCI6	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0C2h	SMCI6	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0C3h	SCI6	Transmit Data Register	TDR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0C4h	SCI6	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0C4h	SMCI6	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0C5h	SCI6	Receive Data Register	RDR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0C6h	SCI6	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0C6h	SMCI6	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0C7h	SCI6	Serial Extended Mode Register	SEMR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0C8h	SCI6	Noise Filter Setting Register	SNFR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0C9h	SCI6	I^2C Mode Register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0CAh	SCI6	I^2C Mode Register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0CBh	SCI6	I^2C Mode Register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0CCh	SCI6	I^2C Status Register	SISR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0CDh	SCI6	SPI Mode Register	SPMR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0CEh	SCI6	Transmit Data Register H	TDRH	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0CFh	SCI6	Transmit Data Register L	TDRL	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0CEh	SCI6	Transmit Data Register HL	TDRHL	16	16	4, 5 PCLKB	2 ICLK	section 32
0008 A0D0h	SCI6	Receive Data Register H	RDRH	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0D1h	SCI6	Receive Data Register L	RDRL	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0D0h	SCI6	Receive Data Register HL	RDRHL	16	16	4, 5 PCLKB	2 ICLK	section 32

Table 5.1 List of I/O Registers (Address Order) (16 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 A0D2h	SCI6	Modulation Duty Register	MDDR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0D3h	SCI6	Data Comparison Control Register	DCCR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0DAh	SCI6	Comparison Data Register H	CDR.H	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0DBh	SCI6	Comparison Data Register L	CDR.L	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0DAh	SCI6	Comparison Data Register	CDR	16	16	4, 5 PCLKB	2 ICLK	section 32
0008 A0DCh	SCI6	Serial Port Register	SPTR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 A0DDh	SCI6	Transmit/Receive Timing Select Register	TMGR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 B000h	CAC	CAC Control Register 0	CACR0	8	8	2, 3 PCLKB	2 ICLK	section 10
0008 B001h	CAC	CAC Control Register 1	CACR1	8	8	2, 3 PCLKB	2 ICLK	section 10
0008 B002h	CAC	CAC Control Register 2	CACR2	8	8	2, 3 PCLKB	2 ICLK	section 10
0008 B003h	CAC	CAC Interrupt Request Enable Register	CAICR	8	8	2, 3 PCLKB	2 ICLK	section 10
0008 B004h	CAC	CAC Status Register	CASTR	8	8	2, 3 PCLKB	2 ICLK	section 10
0008 B006h	CAC	CAC Upper-Limit Value Setting Register	CAULVR	16	16	2, 3 PCLKB	2 ICLK	section 10
0008 B008h	CAC	CAC Lower-Limit Value Setting Register	CALLVR	16	16	2, 3 PCLKB	2 ICLK	section 10
0008 B00Ah	CAC	CAC Counter Buffer Register	CACNTBR	16	16	2, 3 PCLKB	2 ICLK	section 10
0008 B100h	ELC	Event Link Control Register	ELCR	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B101h	ELC	Event Link Setting Register 0	ELSR0	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B104h	ELC	Event Link Setting Register 3	ELSR3	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B105h	ELC	Event Link Setting Register 4	ELSR4	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B108h	ELC	Event Link Setting Register 7	ELSR7	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B10Bh	ELC	Event Link Setting Register 10	ELSR10	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B10Ch	ELC	Event Link Setting Register 11	ELSR11	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B10Dh	ELC	Event Link Setting Register 12	ELSR12	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B10Eh	ELC	Event Link Setting Register 13	ELSR13	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B110h	ELC	Event Link Setting Register 15	ELSR15	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B111h	ELC	Event Link Setting Register 16	ELSR16	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B113h	ELC	Event Link Setting Register 18	ELSR18	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B114h	ELC	Event Link Setting Register 19	ELSR19	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B115h	ELC	Event Link Setting Register 20	ELSR20	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B116h	ELC	Event Link Setting Register 21	ELSR21	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B117h	ELC	Event Link Setting Register 22	ELSR22	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B118h	ELC	Event Link Setting Register 23	ELSR23	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B119h	ELC	Event Link Setting Register 24	ELSR24	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B11Ah	ELC	Event Link Setting Register 25	ELSR25	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B11Bh	ELC	Event Link Setting Register 26	ELSR26	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B11Ch	ELC	Event Link Setting Register 27	ELSR27	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B11Dh	ELC	Event Link Setting Register 28	ELSR28	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B11Fh	ELC	Event Link Option Setting Register A	ELOPA	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B120h	ELC	Event Link Option Setting Register B	ELOPB	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B121h	ELC	Event Link Option Setting Register C	ELOPC	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B122h	ELC	Event Link Option Setting Register D	ELOPD	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B123h	ELC	Port Group Setting Register 1	PGR1	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B124h	ELC	Port Group Setting Register 2	PGR2	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B125h	ELC	Port Group Control Register 1	PGC1	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B126h	ELC	Port Group Control Register 2	PGC2	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B127h	ELC	Port Buffer Register 1	PDBF1	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B128h	ELC	Port Buffer Register 2	PDBF2	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B129h	ELC	Event Link Port Setting Register 0	PEL0	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B12Ah	ELC	Event Link Port Setting Register 1	PEL1	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B12Bh	ELC	Event Link Port Setting Register 2	PEL2	8	8	2, 3 PCLKB	2 ICLK	section 19

Table 5.1 List of I/O Registers (Address Order) (17 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 B12Ch	ELC	Event Link Port Setting Register 3	PEL3	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B12Dh	ELC	Event Link Software Event Generation Register	ELSEGR	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B12Eh	ELC	Event Link Setting Register 30	ELSR30	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B12Fh	ELC	Event Link Setting Register 31	ELSR31	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B131h	ELC	Event Link Setting Register 33	ELSR33	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B13Dh	ELC	Event Link Setting Register 45	ELSR45	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B13Eh	ELC	Event Link Option Setting Register E	ELOPE	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B141h	ELC	Event Link Option Setting Register H	ELOPH	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B144h	ELC	Event Link Setting Register 46	ELSR46	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B145h	ELC	Event Link Setting Register 47	ELSR47	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B146h	ELC	Event Link Setting Register 48	ELSR48	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B147h	ELC	Event Link Setting Register 49	ELSR49	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B148h	ELC	Event Link Setting Register 50	ELSR50	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B149h	ELC	Event Link Setting Register 51	ELSR51	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B14Ah	ELC	Event Link Setting Register 52	ELSR52	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B14Bh	ELC	Event Link Setting Register 53	ELSR53	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B14Ch	ELC	Event Link Setting Register 54	ELSR54	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B14Dh	ELC	Event Link Setting Register 55	ELSR55	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B14Eh	ELC	Event Link Setting Register 56	ELSR56	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B14Fh	ELC	Event Link Setting Register 57	ELSR57	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B150h	ELC	Event Link Setting Register 58	ELSR58	8	8	2, 3 PCLKB	2 ICLK	section 19
0008 B300h	SCI12	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 B300h	SMCI12	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 B301h	SCI12	Bit Rate Register	BRR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 B302h	SCI12	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 B302h	SMCI12	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 B303h	SCI12	Transmit Data Register	TDR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 B304h	SCI12	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 B304h	SMCI12	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 B305h	SCI12	Receive Data Register	RDR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 B306h	SCI12	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 B306h	SMCI12	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 B307h	SCI12	Serial Extended Mode Register	SEMR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 B308h	SCI12	Noise Filter Setting Register	SNFR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 B309h	SCI12	I^2C Mode Register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 B30Ah	SCI12	I^2C Mode Register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 B30Bh	SCI12	I^2C Mode Register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 B30Ch	SCI12	I^2C Status Register	SISR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 B30Dh	SCI12	SPI Mode Register	SPMR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 B30Eh	SCI12	Transmit Data Register H	TDRH	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 B30Fh	SCI12	Transmit Data Register L	TDRL	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 B30Eh	SCI12	Transmit Data Register HL	TDRHL	16	16	4, 5 PCLKB	2 ICLK	section 32
0008 B310h	SCI12	Receive Data Register H	RDRH	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 B311h	SCI12	Receive Data Register L	RDRL	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 B310h	SCI12	Receive Data Register HL	RDRHL	16	16	4, 5 PCLKB	2 ICLK	section 32
0008 B312h	SCI12	Modulation Duty Register	MDDR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 B320h	SCI12	Extended Serial Module Enable Register	ESMER	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 B321h	SCI12	Control Register 0	CR0	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 B322h	SCI12	Control Register 1	CR1	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 B323h	SCI12	Control Register 2	CR2	8	8	2, 3 PCLKB	2 ICLK	section 32

Table 5.1 List of I/O Registers (Address Order) (18 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 B324h	SCI12	Control Register 3	CR3	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 B325h	SCI12	Port Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 B326h	SCI12	Interrupt Control Register	ICR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 B327h	SCI12	Status Register	STR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 B328h	SCI12	Status Clear Register	STCR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 B329h	SCI12	Control Field 0 Data Register	CF0DR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 B32Ah	SCI12	Control Field 0 Compare Enable Register	CF0CR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 B32Bh	SCI12	Control Field 0 Receive Data Register	CF0RR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 B32Ch	SCI12	Primary Control Field 1 Data Register	PCF1DR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 B32Dh	SCI12	Secondary Control Field 1 Data Register	SCF1DR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 B32Eh	SCI12	Control Field 1 Compare Enable Register	CF1CR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 B32Fh	SCI12	Control Field 1 Receive Data Register	CF1RR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 B330h	SCI12	Timer Control Register	TCR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 B331h	SCI12	Timer Mode Register	TMR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 B332h	SCI12	Timer Prescaler Register	TPRE	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 B333h	SCI12	Timer Count Register	TCNT	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 C000h	PORT0	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C001h	PORT1	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C002h	PORT2	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C003h	PORT3	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C004h	PORT4	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C005h	PORT5	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C006h	PORT6	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C007h	PORT7	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C008h	PORT8	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C009h	PORT9	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C00Ah	PORTA	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C00Bh	PORTB	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C00Dh	PORTD	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C00Eh	PORTE	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C016h	PORTN	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C020h	PORT0	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C021h	PORT1	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C022h	PORT2	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C023h	PORT3	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C024h	PORT4	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C025h	PORT5	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C026h	PORT6	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C027h	PORT7	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C028h	PORT8	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C029h	PORT9	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C02Ah	PORTA	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C02Bh	PORTB	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C02Dh	PORTD	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C02Eh	PORTE	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C036h	PORTN	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C040h	PORT0	Port Input Register	PIDR	8	8	4, 5 PCLKB	3 ICLK	section 20
0008 C041h	PORT1	Port Input Register	PIDR	8	8	4, 5 PCLKB	3 ICLK	section 20
0008 C042h	PORT2	Port Input Register	PIDR	8	8	4, 5 PCLKB	3 ICLK	section 20
0008 C043h	PORT3	Port Input Register	PIDR	8	8	4, 5 PCLKB	3 ICLK	section 20

Table 5.1 List of I/O Registers (Address Order) (19 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 C044h	PORT4	Port Input Register	PIDR	8	8	4, 5 PCLKB	3 ICLK	section 20
0008 C045h	PORT5	Port Input Register	PIDR	8	8	4, 5 PCLKB	3 ICLK	section 20
0008 C046h	PORT6	Port Input Register	PIDR	8	8	4, 5 PCLKB	3 ICLK	section 20
0008 C047h	PORT7	Port Input Register	PIDR	8	8	4, 5 PCLKB	3 ICLK	section 20
0008 C048h	PORT8	Port Input Register	PIDR	8	8	4, 5 PCLKB	3 ICLK	section 20
0008 C049h	PORT9	Port Input Register	PIDR	8	8	4, 5 PCLKB	3 ICLK	section 20
0008 C04Ah	PORTA	Port Input Register	PIDR	8	8	4, 5 PCLKB	3 ICLK	section 20
0008 C04Bh	PORTB	Port Input Register	PIDR	8	8	4, 5 PCLKB	3 ICLK	section 20
0008 C04Dh	PORTD	Port Input Register	PIDR	8	8	4, 5 PCLKB	3 ICLK	section 20
0008 C04Eh	PORTE	Port Input Register	PIDR	8	8	4, 5 PCLKB	3 ICLK	section 20
0008 C056h	PORTN	Port Input Register	PIDR	8	8	4, 5 PCLKB	3 ICLK	section 20
0008 C060h	PORT0	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C061h	PORT1	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C062h	PORT2	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C063h	PORT3	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C064h	PORT4	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C065h	PORT5	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C066h	PORT6	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C067h	PORT7	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C068h	PORT8	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C069h	PORT9	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C06Ah	PORTA	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C06Bh	PORTB	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C06Dh	PORTD	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C06Eh	PORTE	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C076h	PORTN	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C080h	PORT0	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C082h	PORT1	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C084h	PORT2	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C085h	PORT2	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C086h	PORT3	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C087h	PORT3	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C088h	PORT4	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C089h	PORT4	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C08Ah	PORT5	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C08Bh	PORT5	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C08Ch	PORT6	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C08Dh	PORT6	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C08Eh	PORT7	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C08Fh	PORT7	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C090h	PORT8	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C092h	PORT9	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C093h	PORT9	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C094h	PORTA	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C095h	PORTA	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C096h	PORTB	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C097h	PORTB	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C09Ah	PORTD	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C09Bh	PORTD	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C09Ch	PORTE	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 20

Table 5.1 List of I/O Registers (Address Order) (20 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 C09Dh	PORTE	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0ADh	PORTN	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0C0h	PORT0	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0C1h	PORT1	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0C2h	PORT2	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0C3h	PORT3	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0C4h	PORT4	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0C5h	PORT5	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0C6h	PORT6	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0C7h	PORT7	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0C8h	PORT8	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0C9h	PORT9	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0CAh	PORTA	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0CBh	PORTB	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0CDh	PORTD	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0CEh	PORTE	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0D6h	PORTN	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0E0h	PORT0	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0E1h	PORT1	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0E2h	PORT2	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0E3h	PORT3	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0E7h	PORT7	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0E8h	PORT8	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0E9h	PORT9	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0EAh	PORTA	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0EBh	PORTB	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0EDh	PORTD	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0EEh	PORTE	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C0F6h	PORTN	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C110h	PORT	Port Output Retention Setting Register 1	POHSR1	16	16	2, 3 PCLKB	2 ICLK	section 20
0008 C112h	PORT	Port Output Retention Setting Register 2	POHSR2	16	16	2, 3 PCLKB	2 ICLK	section 20
0008 C114h	PORT	Port Output Retention Control Register	POHCR	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C11Fh	MPC	Write-Protect Register	PWPR	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C122h	PORT	General-purpose Input/Output Pin Select Extension Register	GPSEXT	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C12Fh	PORT7	Drive Capacity Control Register 2	DSCR2	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C130h	PORT8	Drive Capacity Control Register 2	DSCR2	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C131h	PORT9	Drive Capacity Control Register 2	DSCR2	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C133h	PORTB	Drive Capacity Control Register 2	DSCR2	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C135h	PORTD	Drive Capacity Control Register 2	DSCR2	8	8	2, 3 PCLKB	2 ICLK	section 20
0008 C140h	MPC	P00 Pin Function Control Register	P00PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C141h	MPC	P01 Pin Function Control Register	P01PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C148h	MPC	P10 Pin Function Control Register	P10PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C149h	MPC	P11 Pin Function Control Register	P11PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C150h	MPC	P20 Pin Function Control Register	P20PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C151h	MPC	P21 Pin Function Control Register	P21PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C152h	MPC	P22 Pin Function Control Register	P22PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C153h	MPC	P23 Pin Function Control Register	P23PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C154h	MPC	P24 Pin Function Control Register	P24PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C157h	MPC	P27 Pin Function Control Register	P27PFS	8	8	2, 3 PCLKB	2 ICLK	section 21

Table 5.1 List of I/O Registers (Address Order) (21 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 C158h	MPC	P30 Pin Function Control Register	P30PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C159h	MPC	P31 Pin Function Control Register	P31PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C15Ah	MPC	P32 Pin Function Control Register	P32PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C15Bh	MPC	P33 Pin Function Control Register	P33PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C15Eh	MPC	P36 Pin Function Control Register	P36PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C15Fh	MPC	P37 Pin Function Control Register	P37PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C160h	MPC	P40 Pin Function Control Register	P40PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C161h	MPC	P41 Pin Function Control Register	P41PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C162h	MPC	P42 Pin Function Control Register	P42PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C163h	MPC	P43 Pin Function Control Register	P43PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C164h	MPC	P44 Pin Function Control Register	P44PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C165h	MPC	P45 Pin Function Control Register	P45PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C166h	MPC	P46 Pin Function Control Register	P46PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C167h	MPC	P47 Pin Function Control Register	P47PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C168h	MPC	P50 Pin Function Control Register	P50PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C169h	MPC	P51 Pin Function Control Register	P51PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C16Ah	MPC	P52 Pin Function Control Register	P52PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C16Bh	MPC	P53 Pin Function Control Register	P53PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C16Ch	MPC	P54 Pin Function Control Register	P54PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C16Dh	MPC	P55 Pin Function Control Register	P55PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C170h	MPC	P60 Pin Function Control Register	P60PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C171h	MPC	P61 Pin Function Control Register	P61PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C172h	MPC	P62 Pin Function Control Register	P62PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C173h	MPC	P63 Pin Function Control Register	P63PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C174h	MPC	P64 Pin Function Control Register	P64PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C175h	MPC	P65 Pin Function Control Register	P65PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C178h	MPC	P70 Pin Function Control Register	P70PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C179h	MPC	P71 Pin Function Control Register	P71PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C17Ah	MPC	P72 Pin Function Control Register	P72PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C17Bh	MPC	P73 Pin Function Control Register	P73PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C17Ch	MPC	P74 Pin Function Control Register	P74PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C17Dh	MPC	P75 Pin Function Control Register	P75PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C17Eh	MPC	P76 Pin Function Control Register	P76PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C180h	MPC	P80 Pin Function Control Register	P80PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C181h	MPC	P81 Pin Function Control Register	P81PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C182h	MPC	P82 Pin Function Control Register	P82PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C188h	MPC	P90 Pin Function Control Register	P90PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C189h	MPC	P91 Pin Function Control Register	P91PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C18Ah	MPC	P92 Pin Function Control Register	P92PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C18Bh	MPC	P93 Pin Function Control Register	P93PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C18Ch	MPC	P94 Pin Function Control Register	P94PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C18Dh	MPC	P95 Pin Function Control Register	P95PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C18Eh	MPC	P96 Pin Function Control Register	P96PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C190h	MPC	PA0 Pin Function Control Register	PA0PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C191h	MPC	PA1 Pin Function Control Register	PA1PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C192h	MPC	PA2 Pin Function Control Register	PA2PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C193h	MPC	PA3 Pin Function Control Register	PA3PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C194h	MPC	PA4 Pin Function Control Register	PA4PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C195h	MPC	PA5 Pin Function Control Register	PA5PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C198h	MPC	PB0 Pin Function Control Register	PB0PFS	8	8	2, 3 PCLKB	2 ICLK	section 21

Table 5.1 List of I/O Registers (Address Order) (22 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 C199h	MPC	PB1 Pin Function Control Register	PB1PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C19Ah	MPC	PB2 Pin Function Control Register	PB2PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C19Bh	MPC	PB3 Pin Function Control Register	PB3PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C19Ch	MPC	PB4 Pin Function Control Register	PB4PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C19Dh	MPC	PB5 Pin Function Control Register	PB5PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C19Eh	MPC	PB6 Pin Function Control Register	PB6PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C19Fh	MPC	PB7 Pin Function Control Register	PB7PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1A8h	MPC	PD0 Pin Function Control Register	PD0PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1A9h	MPC	PD1 Pin Function Control Register	PD1PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1AAh	MPC	PD2 Pin Function Control Register	PD2PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1ABh	MPC	PD3 Pin Function Control Register	PD3PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1ACh	MPC	PD4 Pin Function Control Register	PD4PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1ADh	MPC	PD5 Pin Function Control Register	PD5PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1AEh	MPC	PD6 Pin Function Control Register	PD6PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1AFh	MPC	PD7 Pin Function Control Register	PD7PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1B0h	MPC	PE0 Pin Function Control Register	PE0PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1B1h	MPC	PE1 Pin Function Control Register	PE1PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1B2h	MPC	PE2 Pin Function Control Register	PE2PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1B3h	MPC	PE3 Pin Function Control Register	PE3PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1B4h	MPC	PE4 Pin Function Control Register	PE4PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1B5h	MPC	PE5 Pin Function Control Register	PE5PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C1F7h	MPC	PN7 Pin Function Control Register	PN7PFS	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 C290h	SYSTEM	Reset Status Register 0	RSTSR0	8	8	4, 5 PCLKB	2, 3 ICLK	section 6
0008 C291h	SYSTEM	Reset Status Register 1	RSTSR1	8	8	4, 5 PCLKB	2, 3 ICLK	section 6
0008 C293h	SYSTEM	Main Clock Oscillator Function Control Register	MOFCR	8	8	4, 5 PCLKB	2, 3 ICLK	section 9
0008 C294h	SYSTEM	High-Speed On-Chip Oscillator Power Supply Control Register	HOCOPCR	8	8	4, 5 PCLKB	2, 3 ICLK	section 9
0008 C295h	SYSTEM	Voltage Level Setting Register	VOLSR	8	8	4, 5 PCLKB	2, 3 ICLK	section 3
0008 C296h	FLASH	Flash P/E Protect Register	FWEPROR	8	8	4, 5 PCLKB	2, 3 ICLK	section 48
0008 C297h	SYSTEM	Voltage Monitoring Circuit Control Register	LVCMPCR	8	8	4, 5 PCLKB	2, 3 ICLK	section 8
0008 C298h	SYSTEM	Voltage Detection Level Select Register	LVDLVLR	8	8	4, 5 PCLKB	2, 3 ICLK	section 8
0008 C29Ah	SYSTEM	Voltage Monitoring 1 Circuit Control Register 0	LVD1CR0	8	8	4, 5 PCLKB	2, 3 ICLK	section 8
0008 C29Bh	SYSTEM	Voltage Monitoring 2 Circuit Control Register 0	LVD2CR0	8	8	4, 5 PCLKB	2, 3 ICLK	section 8
0009 4200h	CMTW0	Timer Start Register	CMWSTR	16	16	2, 3 PCLKB	2 ICLK	section 29
0009 4204h	CMTW0	Timer Control Register	CMWCR	16	16	2, 3 PCLKB	2 ICLK	section 29
0009 4208h	CMTW0	Timer I/O Control Register	CMWIOR	16	16	2, 3 PCLKB	2 ICLK	section 29
0009 4210h	CMTW0	Timer Counter	CMWCNT	32	32	2, 3 PCLKB	2 ICLK	section 29
0009 4214h	CMTW0	Compare Match Constant Register	CMWCOR	32	32	2, 3 PCLKB	2 ICLK	section 29
0009 4218h	CMTW0	Input Capture Register 0	CMWICR0	32	32	2, 3 PCLKB	2 ICLK	section 29
0009 421Ch	CMTW0	Input Capture Register 1	CMWICR1	32	32	2, 3 PCLKB	2 ICLK	section 29
0009 4220h	CMTW0	Output Compare Register 0	CMWOCR0	32	32	2, 3 PCLKB	2 ICLK	section 29
0009 4224h	CMTW0	Output Compare Register 1	CMWOCR1	32	32	2, 3 PCLKB	2 ICLK	section 29
0009 4280h	CMTW1	Timer Start Register	CMWSTR	16	16	2, 3 PCLKB	2 ICLK	section 29
0009 4284h	CMTW1	Timer Control Register	CMWCR	16	16	2, 3 PCLKB	2 ICLK	section 29
0009 4288h	CMTW1	Timer I/O Control Register	CMWIOR	16	16	2, 3 PCLKB	2 ICLK	section 29
0009 4290h	CMTW1	Timer Counter	CMWCNT	32	32	2, 3 PCLKB	2 ICLK	section 29
0009 4294h	CMTW1	Compare Match Constant Register	CMWCOR	32	32	2, 3 PCLKB	2 ICLK	section 29
0009 4298h	CMTW1	Input Capture Register 0	CMWICR0	32	32	2, 3 PCLKB	2 ICLK	section 29
0009 429Ch	CMTW1	Input Capture Register 1	CMWICR1	32	32	2, 3 PCLKB	2 ICLK	section 29
0009 42A0h	CMTW1	Output Compare Register 0	CMWOCR0	32	32	2, 3 PCLKB	2 ICLK	section 29

Table 5.1 List of I/O Registers (Address Order) (23 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0009 42A4h	CMTW1	Output Compare Register 1	CMWOCR1	32	32	2, 3 PCLKB	2 ICLK	section 29
0009 E000h	POEG	POEG Group A Setting Register	POEGGA	32	32	2, 3 PCLKB	2 ICLK	section 26
0009 E004h	POEG	POEG Group A Input Control Register	POEGICRA	32	32	2, 3 PCLKB	2 ICLK	section 26
0009 E040h	POEG	GPTW Output Stopping Control Group A Write Protection Register	GTONCWPA	16	16	2, 3 PCLKB	2 ICLK	section 26
0009 E044h	POEG	GPTW Output Stopping Control Group A Controlling Register	GTONCCRA	16	16	2, 3 PCLKB	2 ICLK	section 26
0009 E100h	POEG	POEG Group B Setting Register	POEGGB	32	32	2, 3 PCLKB	2 ICLK	section 26
0009 E104h	POEG	POEG Group B Input Control Register	POEGICRB	32	32	2, 3 PCLKB	2 ICLK	section 26
0009 E140h	POEG	GPTW Output Stopping Control Group B Write Protection Register	GTONCWPB	16	16	2, 3 PCLKB	2 ICLK	section 26
0009 E144h	POEG	GPTW Output Stopping Control Group B Controlling Register	GTONCCRB	16	16	2, 3 PCLKB	2 ICLK	section 26
0009 E200h	POEG	POEG Group C Setting Register	POEGGC	32	32	2, 3 PCLKB	2 ICLK	section 26
0009 E204h	POEG	POEG Group C Input Control Register	POEGICRC	32	32	2, 3 PCLKB	2 ICLK	section 26
0009 E240h	POEG	GPTW Output Stopping Control Group C Write Protection Register	GTONCWPC	16	16	2, 3 PCLKB	2 ICLK	section 26
0009 E244h	POEG	GPTW Output Stopping Control Group C Controlling Register	GTONCCRC	16	16	2, 3 PCLKB	2 ICLK	section 26
0009 E300h	POEG	POEG Group D Setting Register	POEGGD	32	32	2, 3 PCLKB	2 ICLK	section 26
0009 E304h	POEG	POEG Group D Input Control Register	POEGICRD	32	32	2, 3 PCLKB	2 ICLK	section 26
0009 E340h	POEG	GPTW Output Stopping Control Group D Write Protection Register	GTONCWPD	16	16	2, 3 PCLKB	2 ICLK	section 26
0009 E344h	POEG	GPTW Output Stopping Control Group D Controlling Register	GTONCCRD	16	16	2, 3 PCLKB	2 ICLK	section 26
0009 E400h	POE	Input Level Control/Status Register 1	ICSR1	16	16	2, 3 PCLKB	2 ICLK	section 23
0009 E402h	POE	Output Level Control/Status Register 1	OCSR1	16	16	2, 3 PCLKB	2 ICLK	section 23
0009 E404h	POE	Input Level Control/Status Register 2	ICSR2	16	16	2, 3 PCLKB	2 ICLK	section 23
0009 E406h	POE	Output Level Control/Status Register 2	OCSR2	16	16	2, 3 PCLKB	2 ICLK	section 23
0009 E408h	POE	Input Level Control/Status Register 3	ICSR3	16	16	2, 3 PCLKB	2 ICLK	section 23
0009 E40Bh	POE	Port Output Enable Control Register 1	POECR1	8	8	2, 3 PCLKB	2 ICLK	section 23
0009 E40Ch	POE	Port Output Enable Control Register 2	POECR2	16	16	2, 3 PCLKB	2 ICLK	section 23
0009 E40Eh	POE	Port Output Enable Control Register 3	POECR3	16	16	2, 3 PCLKB	2 ICLK	section 23
0009 E410h	POE	Port Output Enable Control Register 4	POECR4	16	16	2, 3 PCLKB	2 ICLK	section 23
0009 E412h	POE	Port Output Enable Control Register 5	POECR5	16	16	2, 3 PCLKB	2 ICLK	section 23
0009 E414h	POE	Port Output Enable Control Register 6	POECR6	16	16	2, 3 PCLKB	2 ICLK	section 23
0009 E416h	POE	Input Level Control/Status Register 4	ICSR4	16	16	2, 3 PCLKB	2 ICLK	section 23
0009 E418h	POE	Input Level Control/Status Register 5	ICSR5	16	16	2, 3 PCLKB	2 ICLK	section 23
0009 E41Ah	POE	Active Level Setting Register 1	ALR1	16	16	2, 3 PCLKB	2 ICLK	section 23
0009 E41Ch	POE	Input Level Control/Status Register 6	ICSR6	16	16	2, 3 PCLKB	2 ICLK	section 23
0009 E41Eh	POE	Active Level Setting Register 2	ALR2	16	16	2, 3 PCLKB	2 ICLK	section 23
0009 E420h	POE	Input Level Control/Status Register 7	ICSR7	16	16	2, 3 PCLKB	2 ICLK	section 23
0009 E422h	POE	Port Output Enable Control Register 7	POECR7	16	16	2, 3 PCLKB	2 ICLK	section 23
0009 E424h	POE	Port Output Enable Control Register 8	POECR8	16	16	2, 3 PCLKB	2 ICLK	section 23
0009 E426h	POE	Port Output Enable Comparator Output Detection Flag Register	POECMPFR	16	16	2, 3 PCLKB	2 ICLK	section 23
0009 E428h	POE	Port Output Enable Comparator Request Select Register	POECMPSEL	16	16	2, 3 PCLKB	2 ICLK	section 23
0009 E42Ah	POE	Output Level Control/Status Register 3	OCSR3	16	16	2, 3 PCLKB	2 ICLK	section 23
0009 E42Ch	POE	Active Level Setting Register 3	ALR3	16	16	2, 3 PCLKB	2 ICLK	section 23
0009 E42Eh	POE	Software Port Output Enable Register	SPOER	16	16	2, 3 PCLKB	2 ICLK	section 23
0009 E430h	POE	Port Mode Mask Control Register 0	PMMCR0	16	16	2, 3 PCLKB	2 ICLK	section 23
0009 E432h	POE	Port Mode Mask Control Register 1	PMMCR1	16	16	2, 3 PCLKB	2 ICLK	section 23
0009 E434h	POE	Port Mode Mask Control Register 2	PMMCR2	16	16	2, 3 PCLKB	2 ICLK	section 23

Table 5.1 List of I/O Registers (Address Order) (24 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0009 E438h	POE	Port Output Enable Comparator Request Extended Selection Register 0	POECMPEX0	8	8	2, 3 PCLKB	2 ICLK	section 23
0009 E439h	POE	Port Output Enable Comparator Request Extended Selection Register 1	POECMPEX1	8	8	2, 3 PCLKB	2 ICLK	section 23
0009 E43Ah	POE	Port Output Enable Comparator Request Extended Selection Register 2	POECMPEX2	8	8	2, 3 PCLKB	2 ICLK	section 23
0009 E43Bh	POE	Port Output Enable Comparator Request Extended Selection Register 3	POECMPEX3	8	8	2, 3 PCLKB	2 ICLK	section 23
0009 E43Ch	POE	Port Output Enable Comparator Request Extended Selection Register 4	POECMPEX4	8	8	2, 3 PCLKB	2 ICLK	section 23
0009 E43Dh	POE	Port Output Enable Comparator Request Extended Selection Register 5	POECMPEX5	8	8	2, 3 PCLKB	2 ICLK	section 23
0009 E440h	POE	Input Level Control/Status Register 8	ICSR8	16	16	2, 3 PCLKB	2 ICLK	section 23
0009 E446h	POE	Output Level Control/Status Register 4	OCSR4	16	16	2, 3 PCLKB	2 ICLK	section 23
0009 E448h	POE	Output Level Control/Status Register 5	OCSR5	16	16	2, 3 PCLKB	2 ICLK	section 23
0009 E44Ah	POE	Active Level Setting Register 4	ALR4	16	16	2, 3 PCLKB	2 ICLK	section 23
0009 E44Ch	POE	Active Level Setting Register 5	ALR5	16	16	2, 3 PCLKB	2 ICLK	section 23
0009 E44Eh	POE	Port Output Enable Control Register 4B	POECR4B	16	16	2, 3 PCLKB	2 ICLK	section 23
0009 E450h	POE	Port Output Enable Control Register 6B	POECR6B	16	16	2, 3 PCLKB	2 ICLK	section 23
0009 E452h	POE	Port Output Enable Control Register 9	POECR9	16	16	2, 3 PCLKB	2 ICLK	section 23
0009 E454h	POE	Port Output Enable Control Register 10	POECR10	16	16	2, 3 PCLKB	2 ICLK	section 23
0009 E456h	POE	Port Output Enable Control Register 11	POECR11	16	16	2, 3 PCLKB	2 ICLK	section 23
0009 E458h	POE	Port Output Enable Comparator Request Extended Selection Register 6	POECMPEX6	8	8	2, 3 PCLKB	2 ICLK	section 23
0009 E459h	POE	Port Output Enable Comparator Request Extended Selection Register 7	POECMPEX7	8	8	2, 3 PCLKB	2 ICLK	section 23
0009 E45Ah	POE	Port Output Enable Comparator Request Extended Selection Register 8	POECMPEX8	8	8	2, 3 PCLKB	2 ICLK	section 23
0009 E45Ch	POE	Input Signal Mask Control Register 0	IMCR0	8	8	2, 3 PCLKB	2 ICLK	section 23
0009 E45Dh	POE	Input Signal Mask Control Register 1	IMCR1	8	8	2, 3 PCLKB	2 ICLK	section 23
0009 E45Eh	POE	Input Signal Mask Control Register 2	IMCR2	8	8	2, 3 PCLKB	2 ICLK	section 23
0009 E45Fh	POE	Input Signal Mask Control Register 3	IMCR3	8	8	2, 3 PCLKB	2 ICLK	section 23
0009 E460h	POE	MTU0 Pin Select Register 1	M0SELR1	8	8	2, 3 PCLKB	2 ICLK	section 23
0009 E461h	POE	MTU0 Pin Select Register 2	M0SELR2	8	8	2, 3 PCLKB	2 ICLK	section 23
0009 E462h	POE	MTU3 Pin Select Register	M3SELR	8	8	2, 3 PCLKB	2 ICLK	section 23
0009 E463h	POE	MTU4 Pin Select Register 1	M4SELR1	8	8	2, 3 PCLKB	2 ICLK	section 23
0009 E464h	POE	MTU4 Pin Select Register 2	M4SELR2	8	8	2, 3 PCLKB	2 ICLK	section 23
0009 E465h	POE	MTU6 Pin Select Register	M6SELR	8	8	2, 3 PCLKB	2 ICLK	section 23
0009 E466h	POE	MTU7 Pin Select Register 1	M7SELR1	8	8	2, 3 PCLKB	2 ICLK	section 23
0009 E467h	POE	MTU7 Pin Select Register 2	M7SELR2	8	8	2, 3 PCLKB	2 ICLK	section 23
0009 E468h	POE	MTU9 Pin Select Register 1	M9SELR1	8	8	2, 3 PCLKB	2 ICLK	section 23
0009 E469h	POE	MTU9 Pin Select Register 2	M9SELR2	8	8	2, 3 PCLKB	2 ICLK	section 23
0009 E46Ah	POE	GPTW0 Pin Select Register	G0SELR	8	8	2, 3 PCLKB	2 ICLK	section 23
0009 E46Bh	POE	GPTW1 Pin Select Register	G1SELR	8	8	2, 3 PCLKB	2 ICLK	section 23
0009 E46Ch	POE	GPTW2 Pin Select Register	G2SELR	8	8	2, 3 PCLKB	2 ICLK	section 23
0009 E46Dh	POE	GPTW3 Pin Select Register	G3SELR	8	8	2, 3 PCLKB	2 ICLK	section 23
0009 E46Eh	POE	GPTW4 Pin Select Register	G4SELR	8	8	2, 3 PCLKB	2 ICLK	section 23
0009 E46Fh	POE	GPTW5 Pin Select Register	G5SELR	8	8	2, 3 PCLKB	2 ICLK	section 23
0009 E470h	POE	GPTW6 Pin Select Register	G6SELR	8	8	2, 3 PCLKB	2 ICLK	section 23
0009 E471h	POE	GPTW7 Pin Select Register	G7SELR	8	8	2, 3 PCLKB	2 ICLK	section 23
0009 E474h	POE	Input Signal Mask Control Register 4	IMCR4	8	8	2, 3 PCLKB	2 ICLK	section 23
0009 E475h	POE	Input Signal Mask Control Register 5	IMCR5	8	8	2, 3 PCLKB	2 ICLK	section 23
0009 E476h	POE	Input Signal Mask Control Register 6	IMCR6	8	8	2, 3 PCLKB	2 ICLK	section 23
0009 E479h	POE	Input Signal Mask Control Register 9	IMCR9	8	8	2, 3 PCLKB	2 ICLK	section 23

Table 5.1 List of I/O Registers (Address Order) (25 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0009 E47Ah	POE	Input Signal Mask Control Register 10	IMCR10	8	8	2, 3 PCLKB	2 ICLK	section 23
0009 E47Bh	POE	Input Signal Mask Control Register 11	IMCR11	8	8	2, 3 PCLKB	2 ICLK	section 23
0009 E47Ch	POE	Input Signal Mask Control Register 12	IMCR12	8	8	2, 3 PCLKB	2 ICLK	section 23
0009 E47Dh	POE	Input Signal Mask Control Register 13	IMCR13	8	8	2, 3 PCLKB	2 ICLK	section 23
0009 E47Eh	POE	Input Signal Mask Control Register 14	IMCR14	8	8	2, 3 PCLKB	2 ICLK	section 23
000A 0580h	DOC	DOC Control Register	DOCR	8	8	2, 3 PCLKB	2 ICLK	section 46
000A 0584h	DOC	DOC Status Register	DOSR	8	8	2, 3 PCLKB	2 ICLK	section 46
000A 0588h	DOC	DOC Status Clear Register	DOSCR	8	8	2, 3 PCLKB	2 ICLK	section 46
000A 058Ch	DOC	DOC Data Input Register	DODIR	32	16, 32	2, 3 PCLKB	2 ICLK	section 46
000A 0590h	DOC	DOC Data Setting Register 0	DODSR0	32	16, 32	2, 3 PCLKB	2 ICLK	section 46
000A 0594h	DOC	DOC Data Setting Register 1	DODSR1	32	16, 32	2, 3 PCLKB	2 ICLK	section 46
000A 0C80h	CMPC0	Comparator Control Register	CMPCTL	8	8	1, 2 PCLKB	1, 2 ICLK	section 45
000A 0C84h	CMPC0	Comparator Input Select Register	CMPSEL0	8	8	1, 2 PCLKB	1, 2 ICLK	section 45
000A 0C88h	CMPC0	Comparator Reference Voltage Select Register	CMPSEL1	8	8	1, 2 PCLKB	1, 2 ICLK	section 45
000A 0C8Ch	CMPC0	Comparator Output Monitor Register	CMPMON	8	8	1, 2 PCLKB	1, 2 ICLK	section 45
000A 0C90h	CMPC0	Comparator External Output Enable Register	CMPIOC	8	8	1, 2 PCLKB	1, 2 ICLK	section 45
000A 0C98h	CMPC0	Comparator Control Register 2	CMPCTL2	8	8	1, 2 PCLKB	1, 2 ICLK	section 45
000A 0CA0h	CMPC1	Comparator Control Register	CMPCTL	8	8	1, 2 PCLKB	1, 2 ICLK	section 45
000A 0CA4h	CMPC1	Comparator Input Select Register	CMPSEL0	8	8	1, 2 PCLKB	1, 2 ICLK	section 45
000A 0CA8h	CMPC1	Comparator Reference Voltage Select Register	CMPSEL1	8	8	1, 2 PCLKB	1, 2 ICLK	section 45
000A 0CACh	CMPC1	Comparator Output Monitor Register	CMPMON	8	8	1, 2 PCLKB	1, 2 ICLK	section 45
000A 0CB0h	CMPC1	Comparator External Output Enable Register	CMPIOC	8	8	1, 2 PCLKB	1, 2 ICLK	section 45
000A 0CB8h	CMPC1	Comparator Control Register 2	CMPCTL2	8	8	1, 2 PCLKB	1, 2 ICLK	section 45
000A 0CC0h	CMPC2	Comparator Control Register	CMPCTL	8	8	1, 2 PCLKB	1, 2 ICLK	section 45
000A 0CC4h	CMPC2	Comparator Input Select Register	CMPSEL0	8	8	1, 2 PCLKB	1, 2 ICLK	section 45
000A 0CC8h	CMPC2	Comparator Reference Voltage Select Register	CMPSEL1	8	8	1, 2 PCLKB	1, 2 ICLK	section 45
000A 0CCCh	CMPC2	Comparator Output Monitor Register	CMPMON	8	8	1, 2 PCLKB	1, 2 ICLK	section 45
000A 0CD0h	CMPC2	Comparator External Output Enable Register	CMPIOC	8	8	1, 2 PCLKB	1, 2 ICLK	section 45
000A 0CD8h	CMPC2	Comparator Control Register 2	CMPCTL2	8	8	1, 2 PCLKB	1, 2 ICLK	section 45
000A 0CE0h	CMPC3	Comparator Control Register	CMPCTL	8	8	1, 2 PCLKB	1, 2 ICLK	section 45
000A 0CE4h	CMPC3	Comparator Input Select Register	CMPSEL0	8	8	1, 2 PCLKB	1, 2 ICLK	section 45
000A 0CE8h	CMPC3	Comparator Reference Voltage Select Register	CMPSEL1	8	8	1, 2 PCLKB	1, 2 ICLK	section 45
000A 0CECh	CMPC3	Comparator Output Monitor Register	CMPMON	8	8	1, 2 PCLKB	1, 2 ICLK	section 45
000A 0CF0h	CMPC3	Comparator External Output Enable Register	CMPIOC	8	8	1, 2 PCLKB	1, 2 ICLK	section 45
000A 0CF8h	CMPC3	Comparator Control Register 2	CMPCTL2	8	8	1, 2 PCLKB	1, 2 ICLK	section 45
000A 0D00h	CMPC4	Comparator Control Register	CMPCTL	8	8	1, 2 PCLKB	1, 2 ICLK	section 45
000A 0D04h	CMPC4	Comparator Input Select Register	CMPSEL0	8	8	1, 2 PCLKB	1, 2 ICLK	section 45
000A 0D08h	CMPC4	Comparator Reference Voltage Select Register	CMPSEL1	8	8	1, 2 PCLKB	1, 2 ICLK	section 45
000A 0D0Ch	CMPC4	Comparator Output Monitor Register	CMPMON	8	8	1, 2 PCLKB	1, 2 ICLK	section 45
000A 0D10h	CMPC4	Comparator External Output Enable Register	CMPIOC	8	8	1, 2 PCLKB	1, 2 ICLK	section 45
000A 0D18h	CMPC4	Comparator Control Register 2	CMPCTL2	8	8	1, 2 PCLKB	1, 2 ICLK	section 45
000A 0D20h	CMPC5	Comparator Control Register	CMPCTL	8	8	1, 2 PCLKB	1, 2 ICLK	section 45
000A 0D24h	CMPC5	Comparator Input Select Register	CMPSEL0	8	8	1, 2 PCLKB	1, 2 ICLK	section 45
000A 0D28h	CMPC5	Comparator Reference Voltage Select Register	CMPSEL1	8	8	1, 2 PCLKB	1, 2 ICLK	section 45
000A 0D2Ch	CMPC5	Comparator Output Monitor Register	CMPMON	8	8	1, 2 PCLKB	1, 2 ICLK	section 45
000A 0D30h	CMPC5	Comparator External Output Enable Register	CMPIOC	8	8	1, 2 PCLKB	1, 2 ICLK	section 45
000A 0D38h	CMPC5	Comparator Control Register 2	CMPCTL2	8	8	1, 2 PCLKB	1, 2 ICLK	section 45
000A 1400h	RSCI8	Receive Data Register	RDR	32	8, 16, 32	2, 3 PCLKB	2 ICLK	section 33
000A 1404h	RSCI8	Transmit Data Register	TDR	32	8, 16, 32	2, 3 PCLKB	2 ICLK	section 33
000A 1408h	RSCI8	Control Register 0	SCR0	32	32	2, 3 PCLKB	2 ICLK	section 33

Table 5.1 List of I/O Registers (Address Order) (26 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000A 140Ch	RSCI8	Control Register 1	SCR1	32	32	2, 3 PCLKB	2 ICLK	section 33
000A 1410h	RSCI8	Control Register 2	SCR2	32	32	2, 3 PCLKB	2 ICLK	section 33
000A 1414h	RSCI8	Control Register 3	SCR3	32	32	2, 3 PCLKB	2 ICLK	section 33
000A 1418h	RSCI8	Control Register 4	SCR4	32	32	2, 3 PCLKB	2 ICLK	section 33
000A 141Eh	RSCI8	HBS Support Mode Control Register	HBSCR	8	8	2, 3 PCLKB	2 ICLK	section 33
000A 1420h	RSCI8	I2C Mode Register	SIMR	32	32	2, 3 PCLKB	2 ICLK	section 33
000A 1430h	RSCI8	DE Signal Control Register	DECR	32	32	2, 3 PCLKB	2 ICLK	section 33
000A 1448h	RSCI8	Status Register	SSR	32	32	2, 3 PCLKB	2 ICLK	section 33
000A 144Ch	RSCI8	I2C Status Register	SISR	32	32	2, 3 PCLKB	2 ICLK	section 33
000A 1468h	RSCI8	Status Clear Register	SSCR	32	32	2, 3 PCLKB	2 ICLK	section 33
000A 146Ch	RSCI8	I2C Status Clear Register	SISCR	32	32	2, 3 PCLKB	2 ICLK	section 33
000A 1480h	RSCI9	Receive Data Register	RDR	32	8, 16, 32	2, 3 PCLKB	2 ICLK	section 33
000A 1484h	RSCI9	Transmit Data Register	TDR	32	8, 16, 32	2, 3 PCLKB	2 ICLK	section 33
000A 1488h	RSCI9	Control Register 0	SCR0	32	32	2, 3 PCLKB	2 ICLK	section 33
000A 148Ch	RSCI9	Control Register 1	SCR1	32	32	2, 3 PCLKB	2 ICLK	section 33
000A 1490h	RSCI9	Control Register 2	SCR2	32	32	2, 3 PCLKB	2 ICLK	section 33
000A 1494h	RSCI9	Control Register 3	SCR3	32	32	2, 3 PCLKB	2 ICLK	section 33
000A 1498h	RSCI9	Control Register 4	SCR4	32	32	2, 3 PCLKB	2 ICLK	section 33
000A 149Eh	RSCI9	HBS Support Mode Control Register	HBSCR	8	8	2, 3 PCLKB	2 ICLK	section 33
000A 14A0h	RSCI9	I2C Mode Register	SIMR	32	32	2, 3 PCLKB	2 ICLK	section 33
000A 14ACh	RSCI9	Manchester Mode Control Register	MMCR	32	32	2, 3 PCLKB	2 ICLK	section 33
000A 14B0h	RSCI9	DE Signal Control Register	DECR	32	32	2, 3 PCLKB	2 ICLK	section 33
000A 14B4h	RSCI9	Extended Serial Mode Control Register 0	XCR0	32	32	2, 3 PCLKB	2 ICLK	section 33
000A 14B8h	RSCI9	Extended Serial Mode Control Register 1	XCR1	32	32	2, 3 PCLKB	2 ICLK	section 33
000A 14BCh	RSCI9	Extended Serial Mode Control Register 2	XCR2	32	32	2, 3 PCLKB	2 ICLK	section 33
000A 14C8h	RSCI9	Status Register	SSR	32	32	2, 3 PCLKB	2 ICLK	section 33
000A 14CCh	RSCI9	I2C Status Register	SISR	32	32	2, 3 PCLKB	2 ICLK	section 33
000A 14D8h	RSCI9	Manchester Mode Status Register	MMSR	32	32	2, 3 PCLKB	2 ICLK	section 33
000A 14DCh	RSCI9	Extended Serial Mode Status Register 0	XSR0	32	32	2, 3 PCLKB	2 ICLK	section 33
000A 14E0h	RSCI9	Extended Serial Mode Status Register 1	XSR1	32	32	2, 3 PCLKB	2 ICLK	section 33
000A 14E8h	RSCI9	Status Clear Register	SSCR	32	32	2, 3 PCLKB	2 ICLK	section 33
000A 14ECh	RSCI9	I2C Status Clear Register	SISCR	32	32	2, 3 PCLKB	2 ICLK	section 33
000A 14F4h	RSCI9	Manchester Mode Status Clear Register	MMSCR	32	32	2, 3 PCLKB	2 ICLK	section 33
000A 14F8h	RSCI9	Extended Serial Mode Status Clear Register	XSCR	32	32	2, 3 PCLKB	2 ICLK	section 33
000A 8000h	CANFD0	Nominal Bit Rate Configuration Register	NBCR	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 8004h	CANFD0	Channel Control Register	CHCR	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 8008h	CANFD0	Channel Status Register	CHSR	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 800Ch	CANFD0	Channel Error Status Register	CHESR	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 8014h	CANFD	Global Configuration Register	GCFG	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 8018h	CANFD	Global Control Register	GCR	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 801Ch	CANFD	Global Status Register	GSR	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 8020h	CANFD	Global Error Status Register	GESR	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 8024h	CANFD	Timestamp Counter Register	TSCR	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 8028h	CANFD	Acceptance Filter List Control Register	AFCR	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 802Ch	CANFD	Acceptance Filter List Configuration Register	AFCFG	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 8030h	CANFD	Receive Message Buffer Configuration Register	RMCR	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 8034h	CANFD	Receive Message Buffer New Data Register	RMNDR	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 8038h	CANFD	Receive Message Buffer Interrupt Enable Register	RMIER	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 803Ch	CANFD	Receive FIFO 0 Configuration Register	RFCR0	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 8040h	CANFD	Receive FIFO 1 Configuration Register	RFCR1	32	32	2, 3 PCLKB	1, 2 ICLK	section 36

Table 5.1 List of I/O Registers (Address Order) (27 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000A 8044h	CANFD	Receive FIFO 0 Status Register	RFSR0	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 8048h	CANFD	Receive FIFO 1 Status Register	RFSR1	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 804Ch	CANFD	Receive FIFO 0 Pointer Control Register	RFPCR0	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 8050h	CANFD	Receive FIFO 1 Pointer Control Register	RFPCR1	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 8054h	CANFD	Common FIFO 0 Configuration Register	CFCR0	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 8058h	CANFD	Common FIFO 0 Status Register	CFSR0	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 805Ch	CANFD	Common FIFO 0 Pointer Control Register	CFPCR0	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 8060h	CANFD	FIFO Empty Status Register	FESR	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 8064h	CANFD	FIFO Full Status Register	FFSR	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 8068h	CANFD	FIFO Message Lost Status Register	FMLSR	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 806Ch	CANFD	Receive FIFO Interrupt Status Register	RFISR	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 8070h	CANFD	Transmit Message Buffer 0 Control Register	TMCR0	8	8	2, 3 PCLKB	1, 2 ICLK	section 36
000A 8071h	CANFD	Transmit Message Buffer 1 Control Register	TMCR1	8	8	2, 3 PCLKB	1, 2 ICLK	section 36
000A 8072h	CANFD	Transmit Message Buffer 2 Control Register	TMCR2	8	8	2, 3 PCLKB	1, 2 ICLK	section 36
000A 8073h	CANFD	Transmit Message Buffer 3 Control Register	TMCR3	8	8	2, 3 PCLKB	1, 2 ICLK	section 36
000A 8074h	CANFD	Transmit Message Buffer 0 Status Register	TMSR0	8	8	2, 3 PCLKB	1, 2 ICLK	section 36
000A 8075h	CANFD	Transmit Message Buffer 1 Status Register	TMSR1	8	8	2, 3 PCLKB	1, 2 ICLK	section 36
000A 8076h	CANFD	Transmit Message Buffer 2 Status Register	TMSR2	8	8	2, 3 PCLKB	1, 2 ICLK	section 36
000A 8077h	CANFD	Transmit Message Buffer 3 Status Register	TMSR3	8	8	2, 3 PCLKB	1, 2 ICLK	section 36
000A 8078h	CANFD	Transmit Message Buffer Transmission Request Status Register 0	TMTRSR0	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 807Ch	CANFD	Transmit Message Buffer Transmission Abort Request Status Register 0	TMARSR0	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 8080h	CANFD	Transmit Message Buffer Transmission Completion Status Register 0	TMTCSR0	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 8084h	CANFD	Transmit Message Buffer Transmission Abort Status Register 0	TMTASR0	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 8088h	CANFD	Transmit Message Buffer Interrupt Enable Register	TMIER0	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 808Ch	CANFD0	Transmit Queue 0 Configuration Register	TQCR0	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 8090h	CANFD0	Transmit Queue 0 Status Register	TQSR0	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 8094h	CANFD0	Transmit Queue 0 Pointer Control Register	TQPCR0	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 8098h	CANFD0	Transmission History Configuration Register	THCR	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 809Ch	CANFD0	Transmission History Status Register	THSR	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 80A0h	CANFD0	Transmission History Pointer Control Register	THPCR	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 80A4h	CANFD	Transmit Interrupt Status Register	TISR	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 80A8h	CANFD	Global Test Mode Configuration Register	GTMCR	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 80ACh	CANFD	Global Test Mode Enable Register	GTMER	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 80B0h	CANFD	Global CAN FD Configuration Register	GFDCFG	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 80B8h	CANFD	Global Test Mode Lock Key Register	GTMLKR	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 80C0h	CANFD	Acceptance Filter List Ignore Entry Setting Register	AFIGSR	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 80C4h	CANFD	Acceptance Filter List Ignore Entry Enable Register	AFIGER	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 80C8h	CANFD	DMA Transfer Control Register	DTCR	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 80CCh	CANFD	DMA Transfer Status Register	DTSR	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 80D8h	CANFD	Global Reset Control Register	GRCR	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 8100h	CANFD0	Data Bit Rate Configuration Register	DBCR	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 8104h	CANFD0	CAN FD Configuration Register	FDCFG	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 8108h	CANFD0	CAN FD Control Register	FDCTR	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 810Ch	CANFD0	CAN FD Status Register	FDSTS	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 8110h	CANFD0	CAN FD CRC Register	FDCRC	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 8120h to 000A 812Ch	CANFD	Acceptance Filter List 0	AFL0	128	32	3, 4 PCLKB	1, 2 ICLK	section 36

Table 5.1 List of I/O Registers (Address Order) (28 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000A 8130h to 000A 813Ch	CANFD	Acceptance Filter List 1	AFL1	128	32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 8140h to 000A 814Ch	CANFD	Acceptance Filter List 2	AFL2	128	32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 8150h to 000A 815Ch	CANFD	Acceptance Filter List 3	AFL3	128	32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 8160h to 000A 816Ch	CANFD	Acceptance Filter List 4	AFL4	128	32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 8170h to 000A 817Ch	CANFD	Acceptance Filter List 5	AFL5	128	32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 8180h to 000A 818Ch	CANFD	Acceptance Filter List 6	AFL6	128	32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 8190h to 000A 819Ch	CANFD	Acceptance Filter List 7	AFL7	128	32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 81A0h to 000A 81ACh	CANFD	Acceptance Filter List 8	AFL8	128	32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 81B0h to 000A 81BCh	CANFD	Acceptance Filter List 9	AFL9	128	32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 81C0h to 000A 81CCh	CANFD	Acceptance Filter List 10	AFL10	128	32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 81D0h to 000A 81DCh	CANFD	Acceptance Filter List 11	AFL11	128	32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 81E0h to 000A 81ECh	CANFD	Acceptance Filter List 12	AFL12	128	32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 81F0h to 000A 81FCh	CANFD	Acceptance Filter List 13	AFL13	128	32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 8200h to 000A 820Ch	CANFD	Acceptance Filter List 14	AFL14	128	32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 8210h to 000A 821Ch	CANFD	Acceptance Filter List 15	AFL15	128	32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 8280h to 000A 837Ch	CANFD	RAM Test Page Access Register 0 to RAM Test Page Access Register 63	RTPAR0 to RTPAR63	32	32	2, 3 PCLKB	1, 2 ICLK	section 36
000A 8520h to 000A 856Bh	CANFD	Receive FIFO 0	RFB0	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 856Ch to 000A 85B7h	CANFD	Receive FIFO 1	RFB1	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 85B8h to 000A 8603h	CANFD	Common FIFO 0	CFB0	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 8604h to 000A 864Fh	CANFD	Transmit Message Buffer 0	TMB0	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 8650h to 000A 869Bh	CANFD	Transmit Message Buffer 1	TMB1	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 869Ch to 000A 86E7h	CANFD	Transmit Message Buffer 2	TMB2	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 86E8h to 000A 8733h	CANFD	Transmit Message Buffer 3	TMB3	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 8740h	CANFD0	Transmission History Access Register 0	THACR0	32	32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 8744h	CANFD0	Transmission History Access Register 1	THACR1	32	32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 8920h to 000A 896Bh	CANFD	Receive Message Buffer 0	RMB0	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 896Ch to 000A 89B7h	CANFD	Receive Message Buffer 1	RMB1	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 89B8h to 000A 8A03h	CANFD	Receive Message Buffer 2	RMB2	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 8A04h to 000A 8A4Fh	CANFD	Receive Message Buffer 3	RMB3	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 8A50h to 000A 8A9Bh	CANFD	Receive Message Buffer 4	RMB4	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 8A9Ch to 000A 8AE7h	CANFD	Receive Message Buffer 5	RMB5	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 8AE8h to 000A 8B33h	CANFD	Receive Message Buffer 6	RMB6	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 36

Table 5.1 List of I/O Registers (Address Order) (29 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000A 8B34h to 000A 8B7Fh	CANFD	Receive Message Buffer 7	RMB7	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 8D20h to 000A 8D6Bh	CANFD	Receive Message Buffer 8	RMB8	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 8D6Ch to 000A 8DB7h	CANFD	Receive Message Buffer 9	RMB9	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 8DB8h to 000A 8E03h	CANFD	Receive Message Buffer 10	RMB10	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 8E04h to 000A 8E4Fh	CANFD	Receive Message Buffer 11	RMB11	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 8E50h to 000A 8E9Bh	CANFD	Receive Message Buffer 12	RMB12	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 8E9Ch to 000A 8EE7h	CANFD	Receive Message Buffer 13	RMB13	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 8EE8h to 000A 8F33h	CANFD	Receive Message Buffer 14	RMB14	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 8F34h to 000A 8F7Fh	CANFD	Receive Message Buffer 15	RMB15	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 9120h to 000A 916Bh	CANFD	Receive Message Buffer 16	RMB16	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 916Ch to 000A 91B7h	CANFD	Receive Message Buffer 17	RMB17	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 91B8h to 000A 9203h	CANFD	Receive Message Buffer 18	RMB18	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 9204h to 000A 924Fh	CANFD	Receive Message Buffer 19	RMB19	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 9250h to 000A 929Bh	CANFD	Receive Message Buffer 20	RMB20	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 929Ch to 000A 92E7h	CANFD	Receive Message Buffer 21	RMB21	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 92E8h to 000A 9333h	CANFD	Receive Message Buffer 22	RMB22	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 9334h to 000A 937Fh	CANFD	Receive Message Buffer 23	RMB23	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 9520h to 000A 956Bh	CANFD	Receive Message Buffer 24	RMB24	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 956Ch to 000A 95B7h	CANFD	Receive Message Buffer 25	RMB25	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 95B8h to 000A 9603h	CANFD	Receive Message Buffer 26	RMB26	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 9604h to 000A 964Fh	CANFD	Receive Message Buffer 27	RMB27	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 9650h to 000A 969Bh	CANFD	Receive Message Buffer 28	RMB28	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 969Ch to 000A 96E7h	CANFD	Receive Message Buffer 29	RMB29	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 96E8h to 000A 9733h	CANFD	Receive Message Buffer 30	RMB30	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 36
000A 9734h to 000A 977Fh	CANFD	Receive Message Buffer 31	RMB31	608	8, 16, 32	3, 4 PCLKB	1, 2 ICLK	section 36
000C 1200h	MTU3	Timer Control Register	TCR	8	8, 16, 32	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1201h	MTU4	Timer Control Register	TCR	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1202h	MTU3	Timer Mode Register 1	TMDR1	8	8, 16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1203h	MTU4	Timer Mode Register 1	TMDR1	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1204h	MTU3	Timer I/O Control Register H	TIORH	8	8, 16, 32	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1205h	MTU3	Timer I/O Control Register L	TIORL	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1206h	MTU4	Timer I/O Control Register H	TIORH	8	8, 16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1207h	MTU4	Timer I/O Control Register L	TIORL	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1208h	MTU3	Timer Interrupt Enable Register	TIER	8	8, 16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1209h	MTU4	Timer Interrupt Enable Register	TIER	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22

Table 5.1 List of I/O Registers (Address Order) (30 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000C 120Ah	MTU	Timer Output Master Enable Register A	TOERA	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 120Dh	MTU	Timer Gate Control Register A	TGCRA	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 120Eh	MTU	Timer Output Control Register 1A	TOCR1A	8	8, 16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 120Fh	MTU	Timer Output Control Register 2A	TOCR2A	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1210h	MTU3	Timer Counter	TCNT	16	16, 32	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1212h	MTU4	Timer Counter	TCNT	16	16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1214h	MTU	Timer Period Data Register A	TCDRA	16	16, 32	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1216h	MTU	Timer Dead Time Data Register A	TDDRA	16	16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1218h	MTU3	Timer General Register A	TGRA	16	16, 32	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 121Ah	MTU3	Timer General Register B	TGRB	16	16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 121Ch	MTU4	Timer General Register A	TGRA	16	16, 32	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 121Eh	MTU4	Timer General Register B	TGRB	16	16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1220h	MTU	Timer Subcounter A	TCNTSA	16	16, 32	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1222h	MTU	Timer Period Buffer Register A	TCBRA	16	16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1224h	MTU3	Timer General Register C	TGRC	16	16, 32	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1226h	MTU3	Timer General Register D	TGRD	16	16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1228h	MTU4	Timer General Register C	TGRC	16	16, 32	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 122Ah	MTU4	Timer General Register D	TGRD	16	16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 122Ch	MTU3	Timer Status Register	TSR	8	8, 16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 122Dh	MTU4	Timer Status Register	TSR	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1230h	MTU	Timer Interrupt Skipping Set Register 1A	TITCR1A	8	8, 16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1231h	MTU	Timer Interrupt Skipping Counter 1A	TITCNT1A	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1232h	MTU	Timer Buffer Transfer Set Register A	TBTERA	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1234h	MTU	Timer Dead Time Enable Register A	TDERA	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1236h	MTU	Timer Output Level Buffer Register A	TOLBRA	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1238h	MTU3	Timer Buffer Operation Transfer Mode Register	TBTM	8	8, 16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1239h	MTU4	Timer Buffer Operation Transfer Mode Register	TBTM	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 123Ah	MTU	Timer Interrupt Skipping Mode Register A	TITMRA	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 123Bh	MTU	Timer Interrupt Skipping Set Register 2A	TITCR2A	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 123Ch	MTU	Timer Interrupt Skipping Counter 2A	TITCNT2A	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1240h	MTU4	Timer A/D Conversion Start Request Control Register	TADCR	16	16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1244h	MTU4	Timer A/D Conversion Start Request Cycle Set Register A	TADCORA	16	16, 32	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1246h	MTU4	Timer A/D Conversion Start Request Cycle Set Register B	TADCORB	16	16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1248h	MTU4	Timer A/D Conversion Start Request Cycle Set Buffer Register A	TADCOBRA	16	16, 32	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 124Ah	MTU4	Timer A/D Conversion Start Request Cycle Set Buffer Register B	TADCOBRB	16	16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 124Ch	MTU3	Timer Control Register 2	TCR2	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 124Dh	MTU4	Timer Control Register 2	TCR2	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1260h	MTU	Timer Waveform Control Register A	TWCRA	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1270h	MTU	Timer Mode Register 2A	TMDR2A	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1272h	MTU3	Timer General Register E	TGRE	16	16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1274h	MTU4	Timer General Register E	TGRE	16	16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1276h	MTU4	Timer General Register F	TGRF	16	16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1280h	MTU	Timer Start Register A	TSTRA	8	8, 16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1281h	MTU	Timer Synchronous Register A	TSYRA	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1282h	MTU	Timer Counter Synchronous Start Register	TCSYSTR	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1284h	MTU	Timer Read/Write Enable Register A	TRWERA	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1290h	MTU0	Noise Filter Control Register 0	NFCR0	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22

Table 5.1 List of I/O Registers (Address Order) (31 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000C 1291h	MTU1	Noise Filter Control Register 1	NFCR1	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1292h	MTU2	Noise Filter Control Register 2	NFCR2	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1293h	MTU3	Noise Filter Control Register 3	NFCR3	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1294h	MTU4	Noise Filter Control Register 4	NFCR4	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1296h	MTU9	Noise Filter Control Register 9	NFCR9	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1299h	MTU0	Noise Filter Control Register C	NFCRC	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1300h	MTU0	Timer Control Register	TCR	8	8, 16, 32	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1301h	MTU0	Timer Mode Register 1	TMDR1	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1302h	MTU0	Timer I/O Control Register H	TIORH	8	8, 16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1303h	MTU0	Timer I/O Control Register L	TIORL	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1304h	MTU0	Timer Interrupt Enable Register	TIER	8	8, 16, 32	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1306h	MTU0	Timer Counter	TCNT	16	16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1308h	MTU0	Timer General Register A	TGRA	16	16, 32	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 130Ah	MTU0	Timer General Register B	TGRB	16	16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 130Ch	MTU0	Timer General Register C	TGRC	16	16, 32	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 130Eh	MTU0	Timer General Register D	TGRD	16	16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1320h	MTU0	Timer General Register E	TGRE	16	16, 32	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1322h	MTU0	Timer General Register F	TGRF	16	16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1324h	MTU0	Timer Interrupt Enable Register 2	TIER2	8	8, 16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1326h	MTU0	Timer Buffer Operation Transfer Mode Register	TBTM	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1328h	MTU0	Timer Control Register 2	TCR2	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1380h	MTU1	Timer Control Register	TCR	8	8, 16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1381h	MTU1	Timer Mode Register 1	TMDR1	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1382h	MTU1	Timer I/O Control Register	TIOR	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1384h	MTU1	Timer Interrupt Enable Register	TIER	8	8, 16, 32	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1385h	MTU1	Timer Status Register	TSR	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1386h	MTU1	Timer Counter	TCNT	16	16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1388h	MTU1	Timer General Register A	TGRA	16	16, 32	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 138Ah	MTU1	Timer General Register B	TGRB	16	16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1390h	MTU1	Timer Input Capture Control Register	TICCR	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1391h	MTU1	Timer Mode Register 3	TMDR3	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1394h	MTU1	Timer Control Register 2	TCR2	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 13A0h	MTU1	Timer Longword Counter	TCNTLW	32	32	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 13A4h	MTU1	Timer Longword General Register A	TGRALW	32	32	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 13A8h	MTU1	Timer Longword General Register B	TGRBLW	32	32	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1400h	MTU2	Timer Control Register	TCR	8	8, 16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1401h	MTU2	Timer Mode Register 1	TMDR1	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1402h	MTU2	Timer I/O Control Register	TIOR	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1404h	MTU2	Timer Interrupt Enable Register	TIER	8	8, 16, 32	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1405h	MTU2	Timer Status Register	TSR	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1406h	MTU2	Timer Counter	TCNT	16	16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1408h	MTU2	Timer General Register A	TGRA	16	16, 32	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 140Ah	MTU2	Timer General Register B	TGRB	16	16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 140Ch	MTU2	Timer Control Register 2	TCR2	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1580h	MTU9	Timer Control Register	TCR	8	8, 16, 32	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1581h	MTU9	Timer Mode Register 1	TMDR1	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1582h	MTU9	Timer I/O Control Register H	TIORH	8	8, 16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1583h	MTU9	Timer I/O Control Register L	TIORL	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1584h	MTU9	Timer Interrupt Enable Register	TIER	8	8, 16, 32	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1586h	MTU9	Timer Counter	TCNT	16	16	4 to 7 PCLKA	2 to 4 ICLK	section 22

Table 5.1 List of I/O Registers (Address Order) (32 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000C 1588h	MTU9	Timer General Register A	TGRA	16	16, 32	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 158Ah	MTU9	Timer General Register B	TGRB	16	16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 158Ch	MTU9	Timer General Register C	TGRC	16	16, 32	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 158Eh	MTU9	Timer General Register D	TGRD	16	16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 15A0h	MTU9	Timer General Register E	TGRE	16	16, 32	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 15A2h	MTU9	Timer General Register F	TGRF	16	16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 15A4h	MTU9	Timer Interrupt Enable Register 2	TIER2	8	8, 16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 15A6h	MTU9	Timer Buffer Operation Transfer Mode Register	TBTM	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 15A8h	MTU9	Timer Control Register 2	TCR2	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A00h	MTU6	Timer Control Register	TCR	8	8, 16, 32	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A01h	MTU7	Timer Control Register	TCR	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A02h	MTU6	Timer Mode Register 1	TMDR1	8	8, 16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A03h	MTU7	Timer Mode Register 1	TMDR1	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A04h	MTU6	Timer I/O Control Register H	TIORH	8	8, 16, 32	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A05h	MTU6	Timer I/O Control Register L	TIORL	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A06h	MTU7	Timer I/O Control Register H	TIORH	8	8, 16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A07h	MTU7	Timer I/O Control Register L	TIORL	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A08h	MTU6	Timer Interrupt Enable Register	TIER	8	8, 16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A09h	MTU7	Timer Interrupt Enable Register	TIER	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A0Ah	MTU	Timer Output Master Enable Register B	TOERB	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A0Dh	MTU	Timer Gate Control Register B	TGCRB	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A0Eh	MTU	Timer Output Control Register 1B	TOCR1B	8	8, 16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A0Fh	MTU	Timer Output Control Register 2B	TOCR2B	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A10h	MTU6	Timer Counter	TCNT	16	16, 32	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A12h	MTU7	Timer Counter	TCNT	16	16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A14h	MTU	Timer Period Data Register B	TCDRB	16	16, 32	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A16h	MTU	Timer Dead Time Data Register B	TDDRB	16	16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A18h	MTU6	Timer General Register A	TGRA	16	16, 32	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A1Ah	MTU6	Timer General Register B	TGRB	16	16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A1Ch	MTU7	Timer General Register A	TGRA	16	16, 32	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A1Eh	MTU7	Timer General Register B	TGRB	16	16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A20h	MTU	Timer Subcounter B	TCNTSB	16	16, 32	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A22h	MTU	Timer Period Buffer Register B	TCBRB	16	16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A24h	MTU6	Timer General Register C	TGRC	16	16, 32	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A26h	MTU6	Timer General Register D	TGRD	16	16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A28h	MTU7	Timer General Register C	TGRC	16	16, 32	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A2Ah	MTU7	Timer General Register D	TGRD	16	16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A2Ch	MTU6	Timer Status Register	TSR	8	8, 16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A2Dh	MTU7	Timer Status Register	TSR	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A30h	MTU	Timer Interrupt Skipping Set Register 1B	TITCR1B	8	8, 16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A31h	MTU	Timer Interrupt Skipping Counter 1B	TITCNT1B	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A32h	MTU	Timer Buffer Transfer Set Register B	TBTERB	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A34h	MTU	Timer Dead Time Enable Register B	TDERB	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A36h	MTU	Timer Output Level Buffer Register B	TOLBRB	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A38h	MTU6	Timer Buffer Operation Transfer Mode Register	TBTM	8	8, 16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A39h	MTU7	Timer Buffer Operation Transfer Mode Register	TBTM	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A3Ah	MTU	Timer Interrupt Skipping Mode Register B	TITMRB	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A3Bh	MTU	Timer Interrupt Skipping Set Register 2B	TITCR2B	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A3Ch	MTU	Timer Interrupt Skipping Counter 2B	TITCNT2B	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A40h	MTU7	Timer A/D Conversion Start Request Control Register	TADCR	16	16	4 to 7 PCLKA	2 to 4 ICLK	section 22

Table 5.1 List of I/O Registers (Address Order) (33 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000C 1A44h	MTU7	Timer A/D Conversion Start Request Cycle Set Register A	TADCORA	16	16, 32	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A46h	MTU7	Timer A/D Conversion Start Request Cycle Set Register B	TADCORB	16	16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A48h	MTU7	Timer A/D Conversion Start Request Cycle Set Buffer Register A	TADCOBRA	16	16, 32	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A4Ah	MTU7	Timer A/D Conversion Start Request Cycle Set Buffer Register B	TADCOBRB	16	16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A4Ch	MTU6	Timer Control Register 2	TCR2	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A4Dh	MTU7	Timer Control Register 2	TCR2	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A50h	MTU6	Timer Synchronous Clear Register	TSYCR	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A60h	MTU	Timer Waveform Control Register B	TWCRB	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A70h	MTU	Timer Mode Register 2B	TMDR2B	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A72h	MTU6	Timer General Register E	TGRE	16	16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A74h	MTU7	Timer General Register E	TGRE	16	16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A76h	MTU7	Timer General Register F	TGRF	16	16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A80h	MTU	Timer Start Register B	TSTRB	8	8, 16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A81h	MTU	Timer Synchronous Register B	TSYRB	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A84h	MTU	Timer Read/Write Enable Register B	TRWERB	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A93h	MTU6	Noise Filter Control Register 6	NFCR6	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A94h	MTU7	Noise Filter Control Register 7	NFCR7	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1A95h	MTU5	Noise Filter Control Register 5	NFCR5	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1C80h	MTU5	Timer Counter U	TCNTU	16	16, 32	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1C82h	MTU5	Timer General Register U	TGRU	16	16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1C84h	MTU5	Timer Control Register U	TCRU	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1C85h	MTU5	Timer Control Register 2U	TCR2U	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1C86h	MTU5	Timer I/O Control Register U	TIORU	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1C90h	MTU5	Timer Counter V	TCNTV	16	16, 32	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1C92h	MTU5	Timer General Register V	TGRV	16	16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1C94h	MTU5	Timer Control Register V	TCRV	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1C95h	MTU5	Timer Control Register 2V	TCR2V	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1C96h	MTU5	Timer I/O Control Register V	TIORV	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1CA0h	MTU5	Timer Counter W	TCNTW	16	16, 32	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1CA2h	MTU5	Timer General Register W	TGRW	16	16	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1CA4h	MTU5	Timer Control Register W	TCRW	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1CA5h	MTU5	Timer Control Register 2W	TCR2W	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1CA6h	MTU5	Timer I/O Control Register W	TIORW	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1CB2h	MTU5	Timer Interrupt Enable Register	TIER	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1CB4h	MTU5	Timer Start Register	TSTR	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1CB6h	MTU5	Timer Compare Match Clear Register	TCNTCMPCLR	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1D30h	MTU	A/D Conversion Start Request Select Register 0	TADSTRGR0	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 1D32h	MTU	A/D Conversion Start Request Select Register 1	TADSTRGR1	8	8	4 to 7 PCLKA	2 to 4 ICLK	section 22
000C 2000h	GPTW0	General PWM Timer Write-Protection Register	GTWP	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2004h	GPTW0	General PWM Timer Software Start Register	GTSTR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2008h	GPTW0	General PWM Timer Software Stop Register	GTSTP	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 200Ch	GPTW0	General PWM Timer Software Clear Register	GTCLR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2010h	GPTW0	General PWM Timer Start Source Select Register	GTSSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2014h	GPTW0	General PWM Timer Stop Source Select Register	GTPSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2018h	GPTW0	General PWM Timer Clear Source Select Register	GTCSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 201Ch	GPTW0	General PWM Timer Count-Up Source Select Register	GTUPSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24

Table 5.1 List of I/O Registers (Address Order) (34 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000C 2020h	GPTW0	General PWM Timer Count-Down Source Select Register	GTDNSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2024h	GPTW0	General PWM Timer Input Capture Source Select Register A	GTICASR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2028h	GPTW0	General PWM Timer Input Capture Source Select Register B	GTICBSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 202Ch	GPTW0	General PWM Timer Control Register	GTCR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2030h	GPTW0	General PWM Timer Count Direction and Duty Setting Register	GTUDDTYC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2034h	GPTW0	General PWM Timer I/O Control Register	GTIOR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2038h	GPTW0	General PWM Timer Interrupt Output Setting Register	GTINTAD	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 203Ch	GPTW0	General PWM Timer Status Register	GTST	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2040h	GPTW0	General PWM Timer Buffer Enable Register	GTBER	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2044h	GPTW0	General PWM Timer Interrupt and A/D Conversion Start Request Skipping Setting Register	GTITC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2048h	GPTW0	General PWM Timer Counter	GTCNT	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 204Ch	GPTW0	General PWM Timer Compare Capture Register A	GTCCRA	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2050h	GPTW0	General PWM Timer Compare Capture Register B	GTCCRB	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2054h	GPTW0	General PWM Timer Compare Capture Register C	GTCCRC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2058h	GPTW0	General PWM Timer Compare Capture Register E	GTCCRE	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 205Ch	GPTW0	General PWM Timer Compare Capture Register D	GTCCRD	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2060h	GPTW0	General PWM Timer Compare Capture Register F	GTCCRF	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2064h	GPTW0	General PWM Timer Period Setting Register	GTPR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2068h	GPTW0	General PWM Timer Period Setting Buffer Register	GTPBR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 206Ch	GPTW0	General PWM Timer Period Setting Double-Buffer Register	GTPDBR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2070h	GPTW0	A/D Conversion Start Request Timing Register A	GTADTRA	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2074h	GPTW0	A/D Conversion Start Request Timing Buffer Register A	GTADTBRA	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2078h	GPTW0	A/D Conversion Start Request Timing Double-Buffer Register A	GTADTDBRA	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 207Ch	GPTW0	A/D Conversion Start Request Timing Register B	GTADTRB	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2080h	GPTW0	A/D Conversion Start Request Timing Buffer Register B	GTADTBRB	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2084h	GPTW0	A/D Conversion Start Request Timing Double-Buffer Register B	GTADTDBRB	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2088h	GPTW0	General PWM Timer Dead Time Control Register	GTDTCR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 208Ch	GPTW0	General PWM Timer Dead Time Value Register U	GTDVU	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2090h	GPTW0	General PWM Timer Dead Time Value Register D	GTDVD	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2094h	GPTW0	General PWM Timer Dead Time Value Buffer Register U	GTDBU	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2098h	GPTW0	General PWM Timer Dead Time Value Buffer Register D	GTDBD	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 209Ch	GPTW0	General PWM Timer Output Protection Function Status Register	GTSOS	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 20A0h	GPTW0	General PWM Timer Output Protection Function Temporary Release Register	GTSOTR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 20A4h	GPTW0	General PWM Timer A/D Conversion Start Request Signal Monitoring Register	GTADSMR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 20A8h	GPTW0	General PWM Timer Extended Interrupt Skipping Counter Control Register	GTEITC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 20ACh	GPTW0	General PWM Timer Extended Interrupt Skipping Setting Register 1	GTEITLI1	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 20B0h	GPTW0	General PWM Timer Extended Interrupt Skipping Setting Register 2	GTEITLI2	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 20B4h	GPTW0	General PWM Timer Extended Buffer Transfer Skipping Setting Register	GTEITLB	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 20B8h	GPTW0	General PWM Timer Inter Channel Logical Operation Function Setting Register	GTICLF	32	32	4, 5 PCLKA	2, 3 ICLK	section 24

Table 5.1 List of I/O Registers (Address Order) (35 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000C 20BCh	GPTW0	General PWM Timer Cycle Count Register	GTPC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 20C0h	GPTW0	General PWM Timer A/D Conversion Start Request Compare Match Skipping Control Register	GTADCMSC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 20C4h	GPTW0	General PWM Timer A/D Conversion Start Request Compare Match Skipping Setting Register	GTADCMSS	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 20D0h	GPTW0	General PWM Timer Operation Enable Bit Simultaneous Control Channel Select Register	GTSECSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 20D4h	GPTW0	General PWM Timer Operation Enable Bit Simultaneous Control Register	GTSECR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 20E0h	GPTW0	General PWM Timer Buffer Enable Register 2	GTBER2	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 20E4h	GPTW0	General PWM Timer Output Level Buffer Register	GTOLBR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 20ECh	GPTW0	General PWM Timer Inter Channel Cooperation Input Capture Control Register	GTICCR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2100h	GPTW1	General PWM Timer Write-Protection Register	GTWP	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2104h	GPTW1	General PWM Timer Software Start Register	GTSTR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2108h	GPTW1	General PWM Timer Software Stop Register	GTSTP	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 210Ch	GPTW1	General PWM Timer Software Clear Register	GTCLR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2110h	GPTW1	General PWM Timer Start Source Select Register	GTSSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2114h	GPTW1	General PWM Timer Stop Source Select Register	GTPSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2118h	GPTW1	General PWM Timer Clear Source Select Register	GTCSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 211Ch	GPTW1	General PWM Timer Count-Up Source Select Register	GTUPSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2120h	GPTW1	General PWM Timer Count-Down Source Select Register	GTDNSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2124h	GPTW1	General PWM Timer Input Capture Source Select Register A	GTICASR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2128h	GPTW1	General PWM Timer Input Capture Source Select Register B	GTICBSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 212Ch	GPTW1	General PWM Timer Control Register	GTCR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2130h	GPTW1	General PWM Timer Count Direction and Duty Setting Register	GTUDDTYC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2134h	GPTW1	General PWM Timer I/O Control Register	GTIOR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2138h	GPTW1	General PWM Timer Interrupt Output Setting Register	GTINTAD	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 213Ch	GPTW1	General PWM Timer Status Register	GTST	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2140h	GPTW1	General PWM Timer Buffer Enable Register	GTBER	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2144h	GPTW1	General PWM Timer Interrupt and A/D Conversion Start Request Skipping Setting Register	GTITC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2148h	GPTW1	General PWM Timer Counter	GTCNT	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 214Ch	GPTW1	General PWM Timer Compare Capture Register A	GTCCRA	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2150h	GPTW1	General PWM Timer Compare Capture Register B	GTCCRB	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2154h	GPTW1	General PWM Timer Compare Capture Register C	GTCCRC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2158h	GPTW1	General PWM Timer Compare Capture Register E	GTCCRE	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 215Ch	GPTW1	General PWM Timer Compare Capture Register D	GTCCRD	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2160h	GPTW1	General PWM Timer Compare Capture Register F	GTCCRF	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2164h	GPTW1	General PWM Timer Period Setting Register	GTPR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2168h	GPTW1	General PWM Timer Period Setting Buffer Register	GTPBR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 216Ch	GPTW1	General PWM Timer Period Setting Double-Buffer Register	GTPDBR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2170h	GPTW1	A/D Conversion Start Request Timing Register A	GTADTRA	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2174h	GPTW1	A/D Conversion Start Request Timing Buffer Register A	GTADTBRA	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2178h	GPTW1	A/D Conversion Start Request Timing Double-Buffer Register A	GTADTDBRA	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 217Ch	GPTW1	A/D Conversion Start Request Timing Register B	GTADTRB	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2180h	GPTW1	A/D Conversion Start Request Timing Buffer Register B	GTADTBRB	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2184h	GPTW1	A/D Conversion Start Request Timing Double-Buffer Register B	GTADTDBRB	32	32	4, 5 PCLKA	2, 3 ICLK	section 24

Table 5.1 List of I/O Registers (Address Order) (36 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000C 2188h	GPTW1	General PWM Timer Dead Time Control Register	GTDTCR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 218Ch	GPTW1	General PWM Timer Dead Time Value Register U	GTDVU	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2190h	GPTW1	General PWM Timer Dead Time Value Register D	GTDVD	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2194h	GPTW1	General PWM Timer Dead Time Value Buffer Register U	GTDBU	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2198h	GPTW1	General PWM Timer Dead Time Value Buffer Register D	GTDBD	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 219Ch	GPTW1	General PWM Timer Output Protection Function Status Register	GTSOS	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 21A0h	GPTW1	General PWM Timer Output Protection Function Temporary Release Register	GTSOTR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 21A4h	GPTW1	General PWM Timer A/D Conversion Start Request Signal Monitoring Register	GTADSMR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 21A8h	GPTW1	General PWM Timer Extended Interrupt Skipping Counter Control Register	GTEITC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 21ACh	GPTW1	General PWM Timer Extended Interrupt Skipping Setting Register 1	GTEITLI1	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 21B0h	GPTW1	General PWM Timer Extended Interrupt Skipping Setting Register 2	GTEITLI2	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 21B4h	GPTW1	General PWM Timer Extended Buffer Transfer Skipping Setting Register	GTEITLB	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 21B8h	GPTW1	General PWM Timer Inter Channel Logical Operation Function Setting Register	GTICLF	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 21BCh	GPTW1	General PWM Timer Cycle Count Register	GTPC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 21C0h	GPTW1	General PWM Timer A/D Conversion Start Request Compare Match Skipping Control Register	GTADCMSC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 21C4h	GPTW1	General PWM Timer A/D Conversion Start Request Compare Match Skipping Setting Register	GTADCMSS	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 21D0h	GPTW1	General PWM Timer Operation Enable Bit Simultaneous Control Channel Select Register	GTSECSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 21D4h	GPTW1	General PWM Timer Operation Enable Bit Simultaneous Control Register	GTSECR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 21E0h	GPTW1	General PWM Timer Buffer Enable Register 2	GTBER2	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 21E4h	GPTW1	General PWM Timer Output Level Buffer Register	GTOLBR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 21ECh	GPTW1	General PWM Timer Inter Channel Cooperation Input Capture Control Register	GTICCR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2200h	GPTW2	General PWM Timer Write-Protection Register	GTWP	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2204h	GPTW2	General PWM Timer Software Start Register	GTSTR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2208h	GPTW2	General PWM Timer Software Stop Register	GTSTP	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 220Ch	GPTW2	General PWM Timer Software Clear Register	GTCLR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2210h	GPTW2	General PWM Timer Start Source Select Register	GTSSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2214h	GPTW2	General PWM Timer Stop Source Select Register	GTPSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2218h	GPTW2	General PWM Timer Clear Source Select Register	GTCSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 221Ch	GPTW2	General PWM Timer Count-Up Source Select Register	GTUPSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2220h	GPTW2	General PWM Timer Count-Down Source Select Register	GTDNSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2224h	GPTW2	General PWM Timer Input Capture Source Select Register A	GTICASR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2228h	GPTW2	General PWM Timer Input Capture Source Select Register B	GTICBSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 222Ch	GPTW2	General PWM Timer Control Register	GTCR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2230h	GPTW2	General PWM Timer Count Direction and Duty Setting Register	GTUDDTYC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2234h	GPTW2	General PWM Timer I/O Control Register	GTIOR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2238h	GPTW2	General PWM Timer Interrupt Output Setting Register	GTINTAD	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 223Ch	GPTW2	General PWM Timer Status Register	GTST	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2240h	GPTW2	General PWM Timer Buffer Enable Register	GTBER	32	32	4, 5 PCLKA	2, 3 ICLK	section 24

Table 5.1 List of I/O Registers (Address Order) (37 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000C 2244h	GPTW2	General PWM Timer Interrupt and A/D Conversion Start Request Skipping Setting Register	GTITC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2248h	GPTW2	General PWM Timer Counter	GTCNT	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 224Ch	GPTW2	General PWM Timer Compare Capture Register A	GTCCRA	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2250h	GPTW2	General PWM Timer Compare Capture Register B	GTCCRB	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2254h	GPTW2	General PWM Timer Compare Capture Register C	GTCCRC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2258h	GPTW2	General PWM Timer Compare Capture Register E	GTCCRE	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 225Ch	GPTW2	General PWM Timer Compare Capture Register D	GTCCRD	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2260h	GPTW2	General PWM Timer Compare Capture Register F	GTCCRF	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2264h	GPTW2	General PWM Timer Period Setting Register	GTPR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2268h	GPTW2	General PWM Timer Period Setting Buffer Register	GTPBR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 226Ch	GPTW2	General PWM Timer Period Setting Double-Buffer Register	GTPDBR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2270h	GPTW2	A/D Conversion Start Request Timing Register A	GTADTRA	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2274h	GPTW2	A/D Conversion Start Request Timing Buffer Register A	GTADTBRA	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2278h	GPTW2	A/D Conversion Start Request Timing Double-Buffer Register A	GTADTDBRA	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 227Ch	GPTW2	A/D Conversion Start Request Timing Register B	GTADTRB	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2280h	GPTW2	A/D Conversion Start Request Timing Buffer Register B	GTADTBRB	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2284h	GPTW2	A/D Conversion Start Request Timing Double-Buffer Register B	GTADTDBRB	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2288h	GPTW2	General PWM Timer Dead Time Control Register	GTDTCR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 228Ch	GPTW2	General PWM Timer Dead Time Value Register U	GTDVU	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2290h	GPTW2	General PWM Timer Dead Time Value Register D	GTDVD	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2294h	GPTW2	General PWM Timer Dead Time Value Buffer Register U	GTDBU	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2298h	GPTW2	General PWM Timer Dead Time Value Buffer Register D	GTDBD	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 229Ch	GPTW2	General PWM Timer Output Protection Function Status Register	GTSOS	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 22A0h	GPTW2	General PWM Timer Output Protection Function Temporary Release Register	GTSOTR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 22A4h	GPTW2	General PWM Timer A/D Conversion Start Request Signal Monitoring Register	GTADSMR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 22A8h	GPTW2	General PWM Timer Extended Interrupt Skipping Counter Control Register	GTEITC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 22ACh	GPTW2	General PWM Timer Extended Interrupt Skipping Setting Register 1	GTEITLI1	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 22B0h	GPTW2	General PWM Timer Extended Interrupt Skipping Setting Register 2	GTEITLI2	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 22B4h	GPTW2	General PWM Timer Extended Buffer Transfer Skipping Setting Register	GTEITLB	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 22B8h	GPTW2	General PWM Timer Inter Channel Logical Operation Function Setting Register	GTICLF	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 22BCh	GPTW2	General PWM Timer Cycle Count Register	GTPC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 22C0h	GPTW2	General PWM Timer A/D Conversion Start Request Compare Match Skipping Control Register	GTADCMSC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 22C4h	GPTW2	General PWM Timer A/D Conversion Start Request Compare Match Skipping Setting Register	GTADCMSS	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 22D0h	GPTW2	General PWM Timer Operation Enable Bit Simultaneous Control Channel Select Register	GTSECSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 22D4h	GPTW2	General PWM Timer Operation Enable Bit Simultaneous Control Register	GTSECR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 22E0h	GPTW2	General PWM Timer Buffer Enable Register 2	GTBER2	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 22E4h	GPTW2	General PWM Timer Output Level Buffer Register	GTOLBR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 22ECh	GPTW2	General PWM Timer Inter Channel Cooperation Input Capture Control Register	GTICCR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2300h	GPTW3	General PWM Timer Write-Protection Register	GTWP	32	32	4, 5 PCLKA	2, 3 ICLK	section 24

Table 5.1 List of I/O Registers (Address Order) (38 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000C 2304h	GPTW3	General PWM Timer Software Start Register	GTSTR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2308h	GPTW3	General PWM Timer Software Stop Register	GTSTP	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 230Ch	GPTW3	General PWM Timer Software Clear Register	GTCLR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2310h	GPTW3	General PWM Timer Start Source Select Register	GTSSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2314h	GPTW3	General PWM Timer Stop Source Select Register	GTPSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2318h	GPTW3	General PWM Timer Clear Source Select Register	GTCSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 231Ch	GPTW3	General PWM Timer Count-Up Source Select Register	GTUPSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2320h	GPTW3	General PWM Timer Count-Down Source Select Register	GTDNSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2324h	GPTW3	General PWM Timer Input Capture Source Select Register A	GTICASR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2328h	GPTW3	General PWM Timer Input Capture Source Select Register B	GTICBSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 232Ch	GPTW3	General PWM Timer Control Register	GTCR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2330h	GPTW3	General PWM Timer Count Direction and Duty Setting Register	GTUDDTYC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2334h	GPTW3	General PWM Timer I/O Control Register	GTIOR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2338h	GPTW3	General PWM Timer Interrupt Output Setting Register	GTINTAD	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 233Ch	GPTW3	General PWM Timer Status Register	GTST	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2340h	GPTW3	General PWM Timer Buffer Enable Register	GTBER	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2344h	GPTW3	General PWM Timer Interrupt and A/D Conversion Start Request Skipping Setting Register	GTITC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2348h	GPTW3	General PWM Timer Counter	GTCNT	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 234Ch	GPTW3	General PWM Timer Compare Capture Register A	GTCCRA	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2350h	GPTW3	General PWM Timer Compare Capture Register B	GTCCRB	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2354h	GPTW3	General PWM Timer Compare Capture Register C	GTCCRC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2358h	GPTW3	General PWM Timer Compare Capture Register E	GTCCRE	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 235Ch	GPTW3	General PWM Timer Compare Capture Register D	GTCCRD	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2360h	GPTW3	General PWM Timer Compare Capture Register F	GTCCRF	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2364h	GPTW3	General PWM Timer Period Setting Register	GTPR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2368h	GPTW3	General PWM Timer Period Setting Buffer Register	GTPBR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 236Ch	GPTW3	General PWM Timer Period Setting Double-Buffer Register	GTPDBR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2370h	GPTW3	A/D Conversion Start Request Timing Register A	GTADTRA	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2374h	GPTW3	A/D Conversion Start Request Timing Buffer Register A	GTADTBRA	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2378h	GPTW3	A/D Conversion Start Request Timing Double-Buffer Register A	GTADTDBRA	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 237Ch	GPTW3	A/D Conversion Start Request Timing Register B	GTADTRB	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2380h	GPTW3	A/D Conversion Start Request Timing Buffer Register B	GTADTBRB	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2384h	GPTW3	A/D Conversion Start Request Timing Double-Buffer Register B	GTADTDBRB	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2388h	GPTW3	General PWM Timer Dead Time Control Register	GTDTCR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 238Ch	GPTW3	General PWM Timer Dead Time Value Register U	GTDVU	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2390h	GPTW3	General PWM Timer Dead Time Value Register D	GTDVD	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2394h	GPTW3	General PWM Timer Dead Time Value Buffer Register U	GTDBU	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2398h	GPTW3	General PWM Timer Dead Time Value Buffer Register D	GTDBD	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 239Ch	GPTW3	General PWM Timer Output Protection Function Status Register	GTSOS	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 23A0h	GPTW3	General PWM Timer Output Protection Function Temporary Release Register	GTSOTR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 23A4h	GPTW3	General PWM Timer A/D Conversion Start Request Signal Monitoring Register	GTADSMR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 23A8h	GPTW3	General PWM Timer Extended Interrupt Skipping Counter Control Register	GTEITC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24

Table 5.1 List of I/O Registers (Address Order) (39 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000C 23ACh	GPTW3	General PWM Timer Extended Interrupt Skipping Setting Register 1	GTEITLI1	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 23B0h	GPTW3	General PWM Timer Extended Interrupt Skipping Setting Register 2	GTEITLI2	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 23B4h	GPTW3	General PWM Timer Extended Buffer Transfer Skipping Setting Register	GTEITLB	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 23B8h	GPTW3	General PWM Timer Inter Channel Logical Operation Function Setting Register	GTICLF	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 23BCh	GPTW3	General PWM Timer Cycle Count Register	GTPC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 23C0h	GPTW3	General PWM Timer A/D Conversion Start Request Compare Match Skipping Control Register	GTADCMSC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 23C4h	GPTW3	General PWM Timer A/D Conversion Start Request Compare Match Skipping Setting Register	GTADCMSS	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 23D0h	GPTW3	General PWM Timer Operation Enable Bit Simultaneous Control Channel Select Register	GTSECSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 23D4h	GPTW3	General PWM Timer Operation Enable Bit Simultaneous Control Register	GTSECR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 23E0h	GPTW3	General PWM Timer Buffer Enable Register 2	GTBER2	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 23E4h	GPTW3	General PWM Timer Output Level Buffer Register	GTOLBR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 23ECh	GPTW3	General PWM Timer Inter Channel Cooperation Input Capture Control Register	GTICCR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2400h	GPTW4	General PWM Timer Write-Protection Register	GTWP	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2404h	GPTW4	General PWM Timer Software Start Register	GTSTR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2408h	GPTW4	General PWM Timer Software Stop Register	GTSTP	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 240Ch	GPTW4	General PWM Timer Software Clear Register	GTCLR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2410h	GPTW4	General PWM Timer Start Source Select Register	GTSSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2414h	GPTW4	General PWM Timer Stop Source Select Register	GTPSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2418h	GPTW4	General PWM Timer Clear Source Select Register	GTCSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 241Ch	GPTW4	General PWM Timer Count-Up Source Select Register	GTUPSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2420h	GPTW4	General PWM Timer Count-Down Source Select Register	GTDNSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2424h	GPTW4	General PWM Timer Input Capture Source Select Register A	GTICASR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2428h	GPTW4	General PWM Timer Input Capture Source Select Register B	GTICBSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 242Ch	GPTW4	General PWM Timer Control Register	GTCR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2430h	GPTW4	General PWM Timer Count Direction and Duty Setting Register	GTUDDTYC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2434h	GPTW4	General PWM Timer I/O Control Register	GTIOR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2438h	GPTW4	General PWM Timer Interrupt Output Setting Register	GTINTAD	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 243Ch	GPTW4	General PWM Timer Status Register	GTST	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2440h	GPTW4	General PWM Timer Buffer Enable Register	GTBER	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2444h	GPTW4	General PWM Timer Interrupt and A/D Conversion Start Request Skipping Setting Register	GTITC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2448h	GPTW4	General PWM Timer Counter	GTCNT	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 244Ch	GPTW4	General PWM Timer Compare Capture Register A	GTCCRA	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2450h	GPTW4	General PWM Timer Compare Capture Register B	GTCCRB	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2454h	GPTW4	General PWM Timer Compare Capture Register C	GTCCRC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2458h	GPTW4	General PWM Timer Compare Capture Register E	GTCCRE	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 245Ch	GPTW4	General PWM Timer Compare Capture Register D	GTCCRD	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2460h	GPTW4	General PWM Timer Compare Capture Register F	GTCCRF	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2464h	GPTW4	General PWM Timer Period Setting Register	GTPR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2468h	GPTW4	General PWM Timer Period Setting Buffer Register	GTPBR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 246Ch	GPTW4	General PWM Timer Period Setting Double-Buffer Register	GTPDBR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2470h	GPTW4	A/D Conversion Start Request Timing Register A	GTADTRA	32	32	4, 5 PCLKA	2, 3 ICLK	section 24

Table 5.1 List of I/O Registers (Address Order) (40 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000C 2474h	GPTW4	A/D Conversion Start Request Timing Buffer Register A	GTADTBRA	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2478h	GPTW4	A/D Conversion Start Request Timing Double-Buffer Register A	GTADTDBRA	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 247Ch	GPTW4	A/D Conversion Start Request Timing Register B	GTADTRB	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2480h	GPTW4	A/D Conversion Start Request Timing Buffer Register B	GTADTBRB	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2484h	GPTW4	A/D Conversion Start Request Timing Double-Buffer Register B	GTADTDBRB	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2488h	GPTW4	General PWM Timer Dead Time Control Register	GTDTCR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 248Ch	GPTW4	General PWM Timer Dead Time Value Register U	GTDVU	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2490h	GPTW4	General PWM Timer Dead Time Value Register D	GTDVD	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2494h	GPTW4	General PWM Timer Dead Time Value Buffer Register U	GTDBU	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2498h	GPTW4	General PWM Timer Dead Time Value Buffer Register D	GTDBD	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 249Ch	GPTW4	General PWM Timer Output Protection Function Status Register	GTSOS	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 24A0h	GPTW4	General PWM Timer Output Protection Function Temporary Release Register	GTSOTR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 24A4h	GPTW4	General PWM Timer A/D Conversion Start Request Signal Monitoring Register	GTADSMR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 24A8h	GPTW4	General PWM Timer Extended Interrupt Skipping Counter Control Register	GTEITC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 24ACh	GPTW4	General PWM Timer Extended Interrupt Skipping Setting Register 1	GTEITLI1	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 24B0h	GPTW4	General PWM Timer Extended Interrupt Skipping Setting Register 2	GTEITLI2	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 24B4h	GPTW4	General PWM Timer Extended Buffer Transfer Skipping Setting Register	GTEITLB	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 24B8h	GPTW4	General PWM Timer Inter Channel Logical Operation Function Setting Register	GTICLF	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 24BCh	GPTW4	General PWM Timer Cycle Count Register	GTPC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 24C0h	GPTW4	General PWM Timer A/D Conversion Start Request Compare Match Skipping Control Register	GTADCMSC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 24C4h	GPTW4	General PWM Timer A/D Conversion Start Request Compare Match Skipping Setting Register	GTADCMSS	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 24D0h	GPTW4	General PWM Timer Operation Enable Bit Simultaneous Control Channel Select Register	GTSECSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 24D4h	GPTW4	General PWM Timer Operation Enable Bit Simultaneous Control Register	GTSECR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 24E0h	GPTW4	General PWM Timer Buffer Enable Register 2	GTBER2	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 24E4h	GPTW4	General PWM Timer Output Level Buffer Register	GTOLBR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 24ECh	GPTW4	General PWM Timer Inter Channel Cooperation Input Capture Control Register	GTICCR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2500h	GPTW5	General PWM Timer Write-Protection Register	GTWP	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2504h	GPTW5	General PWM Timer Software Start Register	GTSTR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2508h	GPTW5	General PWM Timer Software Stop Register	GTSTP	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 250Ch	GPTW5	General PWM Timer Software Clear Register	GTCLR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2510h	GPTW5	General PWM Timer Start Source Select Register	GTSSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2514h	GPTW5	General PWM Timer Stop Source Select Register	GTPSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2518h	GPTW5	General PWM Timer Clear Source Select Register	GTCSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 251Ch	GPTW5	General PWM Timer Count-Up Source Select Register	GTUPSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2520h	GPTW5	General PWM Timer Count-Down Source Select Register	GTDNSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2524h	GPTW5	General PWM Timer Input Capture Source Select Register A	GTICASR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2528h	GPTW5	General PWM Timer Input Capture Source Select Register B	GTICBSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 252Ch	GPTW5	General PWM Timer Control Register	GTCR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24

Table 5.1 List of I/O Registers (Address Order) (41 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000C 2530h	GPTW5	General PWM Timer Count Direction and Duty Setting Register	GTUDDTYC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2534h	GPTW5	General PWM Timer I/O Control Register	GTIOR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2538h	GPTW5	General PWM Timer Interrupt Output Setting Register	GTINTAD	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 253Ch	GPTW5	General PWM Timer Status Register	GTST	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2540h	GPTW5	General PWM Timer Buffer Enable Register	GTBER	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2544h	GPTW5	General PWM Timer Interrupt and A/D Conversion Start Request Skipping Setting Register	GTITC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2548h	GPTW5	General PWM Timer Counter	GTCNT	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 254Ch	GPTW5	General PWM Timer Compare Capture Register A	GTCCRA	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2550h	GPTW5	General PWM Timer Compare Capture Register B	GTCCRB	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2554h	GPTW5	General PWM Timer Compare Capture Register C	GTCCRC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2558h	GPTW5	General PWM Timer Compare Capture Register E	GTCCRE	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 255Ch	GPTW5	General PWM Timer Compare Capture Register D	GTCCRD	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2560h	GPTW5	General PWM Timer Compare Capture Register F	GTCCRF	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2564h	GPTW5	General PWM Timer Period Setting Register	GTPR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2568h	GPTW5	General PWM Timer Period Setting Buffer Register	GTPBR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 256Ch	GPTW5	General PWM Timer Period Setting Double-Buffer Register	GTPDBR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2570h	GPTW5	A/D Conversion Start Request Timing Register A	GTADTRA	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2574h	GPTW5	A/D Conversion Start Request Timing Buffer Register A	GTADTBRA	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2578h	GPTW5	A/D Conversion Start Request Timing Double-Buffer Register A	GTADTDBRA	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 257Ch	GPTW5	A/D Conversion Start Request Timing Register B	GTADTRB	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2580h	GPTW5	A/D Conversion Start Request Timing Buffer Register B	GTADTBRB	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2584h	GPTW5	A/D Conversion Start Request Timing Double-Buffer Register B	GTADTDBRB	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2588h	GPTW5	General PWM Timer Dead Time Control Register	GTDTCR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 258Ch	GPTW5	General PWM Timer Dead Time Value Register U	GTDVU	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2590h	GPTW5	General PWM Timer Dead Time Value Register D	GTDVD	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2594h	GPTW5	General PWM Timer Dead Time Value Buffer Register U	GTDBU	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2598h	GPTW5	General PWM Timer Dead Time Value Buffer Register D	GTDBD	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 259Ch	GPTW5	General PWM Timer Output Protection Function Status Register	GTSOS	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 25A0h	GPTW5	General PWM Timer Output Protection Function Temporary Release Register	GTSOTR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 25A4h	GPTW5	General PWM Timer A/D Conversion Start Request Signal Monitoring Register	GTADSMR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 25A8h	GPTW5	General PWM Timer Extended Interrupt Skipping Counter Control Register	GTEITC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 25ACh	GPTW5	General PWM Timer Extended Interrupt Skipping Setting Register 1	GTEITLI1	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 25B0h	GPTW5	General PWM Timer Extended Interrupt Skipping Setting Register 2	GTEITLI2	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 25B4h	GPTW5	General PWM Timer Extended Buffer Transfer Skipping Setting Register	GTEITLB	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 25B8h	GPTW5	General PWM Timer Inter Channel Logical Operation Function Setting Register	GTICLF	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 25BCh	GPTW5	General PWM Timer Cycle Count Register	GTPC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 25C0h	GPTW5	General PWM Timer A/D Conversion Start Request Compare Match Skipping Control Register	GTADCMSC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 25C4h	GPTW5	General PWM Timer A/D Conversion Start Request Compare Match Skipping Setting Register	GTADCMSS	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 25D0h	GPTW5	General PWM Timer Operation Enable Bit Simultaneous Control Channel Select Register	GTSECSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24

Table 5.1 List of I/O Registers (Address Order) (42 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000C 25D4h	GPTW5	General PWM Timer Operation Enable Bit Simultaneous Control Register	GTSECR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 25E0h	GPTW5	General PWM Timer Buffer Enable Register 2	GTBER2	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 25E4h	GPTW5	General PWM Timer Output Level Buffer Register	GTOLBR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 25ECh	GPTW5	General PWM Timer Inter Channel Cooperation Input Capture Control Register	GTICCR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2600h	GPTW6	General PWM Timer Write-Protection Register	GTWP	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2604h	GPTW6	General PWM Timer Software Start Register	GTSTR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2608h	GPTW6	General PWM Timer Software Stop Register	GTSTP	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 260Ch	GPTW6	General PWM Timer Software Clear Register	GTCLR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2610h	GPTW6	General PWM Timer Start Source Select Register	GTSSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2614h	GPTW6	General PWM Timer Stop Source Select Register	GTPSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2618h	GPTW6	General PWM Timer Clear Source Select Register	GTCSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 261Ch	GPTW6	General PWM Timer Count-Up Source Select Register	GTUPSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2620h	GPTW6	General PWM Timer Count-Down Source Select Register	GTDNSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2624h	GPTW6	General PWM Timer Input Capture Source Select Register A	GTICASR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2628h	GPTW6	General PWM Timer Input Capture Source Select Register B	GTICBSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 262Ch	GPTW6	General PWM Timer Control Register	GTCR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2630h	GPTW6	General PWM Timer Count Direction and Duty Setting Register	GTUDDTYC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2634h	GPTW6	General PWM Timer I/O Control Register	GTIOR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2638h	GPTW6	General PWM Timer Interrupt Output Setting Register	GTINTAD	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 263Ch	GPTW6	General PWM Timer Status Register	GTST	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2640h	GPTW6	General PWM Timer Buffer Enable Register	GTBER	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2644h	GPTW6	General PWM Timer Interrupt and A/D Conversion Start Request Skipping Setting Register	GTITC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2648h	GPTW6	General PWM Timer Counter	GTCNT	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 264Ch	GPTW6	General PWM Timer Compare Capture Register A	GTCCRA	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2650h	GPTW6	General PWM Timer Compare Capture Register B	GTCCRB	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2654h	GPTW6	General PWM Timer Compare Capture Register C	GTCCRC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2658h	GPTW6	General PWM Timer Compare Capture Register E	GTCCRE	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 265Ch	GPTW6	General PWM Timer Compare Capture Register D	GTCCRD	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2660h	GPTW6	General PWM Timer Compare Capture Register F	GTCCRF	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2664h	GPTW6	General PWM Timer Period Setting Register	GTPR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2668h	GPTW6	General PWM Timer Period Setting Buffer Register	GTPBR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 266Ch	GPTW6	General PWM Timer Period Setting Double-Buffer Register	GTPDBR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2670h	GPTW6	A/D Conversion Start Request Timing Register A	GTADTRA	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2674h	GPTW6	A/D Conversion Start Request Timing Buffer Register A	GTADTBRA	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2678h	GPTW6	A/D Conversion Start Request Timing Double-Buffer Register A	GTADTDBRA	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 267Ch	GPTW6	A/D Conversion Start Request Timing Register B	GTADTRB	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2680h	GPTW6	A/D Conversion Start Request Timing Buffer Register B	GTADTBRB	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2684h	GPTW6	A/D Conversion Start Request Timing Double-Buffer Register B	GTADTDBRB	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2688h	GPTW6	General PWM Timer Dead Time Control Register	GTDTCR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 268Ch	GPTW6	General PWM Timer Dead Time Value Register U	GTDVU	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2690h	GPTW6	General PWM Timer Dead Time Value Register D	GTDVD	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2694h	GPTW6	General PWM Timer Dead Time Value Buffer Register U	GTDBU	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2698h	GPTW6	General PWM Timer Dead Time Value Buffer Register D	GTDBD	32	32	4, 5 PCLKA	2, 3 ICLK	section 24

Table 5.1 List of I/O Registers (Address Order) (43 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000C 269Ch	GPTW6	General PWM Timer Output Protection Function Status Register	GTSOS	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 26A0h	GPTW6	General PWM Timer Output Protection Function Temporary Release Register	GTSOTR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 26A4h	GPTW6	General PWM Timer A/D Conversion Start Request Signal Monitoring Register	GTADSMR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 26A8h	GPTW6	General PWM Timer Extended Interrupt Skipping Counter Control Register	GTEITC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 26ACh	GPTW6	General PWM Timer Extended Interrupt Skipping Setting Register 1	GTEITLI1	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 26B0h	GPTW6	General PWM Timer Extended Interrupt Skipping Setting Register 2	GTEITLI2	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 26B4h	GPTW6	General PWM Timer Extended Buffer Transfer Skipping Setting Register	GTEITLB	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 26B8h	GPTW6	General PWM Timer Inter Channel Logical Operation Function Setting Register	GTICLF	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 26BCh	GPTW6	General PWM Timer Cycle Count Register	GTPC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 26C0h	GPTW6	General PWM Timer A/D Conversion Start Request Compare Match Skipping Control Register	GTADCMSC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 26C4h	GPTW6	General PWM Timer A/D Conversion Start Request Compare Match Skipping Setting Register	GTADCMSS	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 26D0h	GPTW6	General PWM Timer Operation Enable Bit Simultaneous Control Channel Select Register	GTSECSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 26D4h	GPTW6	General PWM Timer Operation Enable Bit Simultaneous Control Register	GTSECR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 26E0h	GPTW6	General PWM Timer Buffer Enable Register 2	GTBER2	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 26E4h	GPTW6	General PWM Timer Output Level Buffer Register	GTOLBR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 26ECh	GPTW6	General PWM Timer Inter Channel Cooperation Input Capture Control Register	GTICCR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2700h	GPTW7	General PWM Timer Write-Protection Register	GTWP	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2704h	GPTW7	General PWM Timer Software Start Register	GTSTR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2708h	GPTW7	General PWM Timer Software Stop Register	GTSTP	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 270Ch	GPTW7	General PWM Timer Software Clear Register	GTCLR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2710h	GPTW7	General PWM Timer Start Source Select Register	GTSSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2714h	GPTW7	General PWM Timer Stop Source Select Register	GTPSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2718h	GPTW7	General PWM Timer Clear Source Select Register	GTCSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 271Ch	GPTW7	General PWM Timer Count-Up Source Select Register	GTUPSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2720h	GPTW7	General PWM Timer Count-Down Source Select Register	GTDNSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2724h	GPTW7	General PWM Timer Input Capture Source Select Register A	GTICASR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2728h	GPTW7	General PWM Timer Input Capture Source Select Register B	GTICBSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 272Ch	GPTW7	General PWM Timer Control Register	GTCR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2730h	GPTW7	General PWM Timer Count Direction and Duty Setting Register	GTUDDTYC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2734h	GPTW7	General PWM Timer I/O Control Register	GTIOR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2738h	GPTW7	General PWM Timer Interrupt Output Setting Register	GTINTAD	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 273Ch	GPTW7	General PWM Timer Status Register	GTST	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2740h	GPTW7	General PWM Timer Buffer Enable Register	GTBER	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2744h	GPTW7	General PWM Timer Interrupt and A/D Conversion Start Request Skipping Setting Register	GTITC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2748h	GPTW7	General PWM Timer Counter	GTCNT	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 274Ch	GPTW7	General PWM Timer Compare Capture Register A	GTCCRA	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2750h	GPTW7	General PWM Timer Compare Capture Register B	GTCCRB	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2754h	GPTW7	General PWM Timer Compare Capture Register C	GTCCRC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2758h	GPTW7	General PWM Timer Compare Capture Register E	GTCCRE	32	32	4, 5 PCLKA	2, 3 ICLK	section 24

Table 5.1 List of I/O Registers (Address Order) (44 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000C 275Ch	GPTW7	General PWM Timer Compare Capture Register D	GTCCRD	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2760h	GPTW7	General PWM Timer Compare Capture Register F	GTCCRF	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2764h	GPTW7	General PWM Timer Period Setting Register	GTPR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2768h	GPTW7	General PWM Timer Period Setting Buffer Register	GTPBR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 276Ch	GPTW7	General PWM Timer Period Setting Double-Buffer Register	GTPDBR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2770h	GPTW7	A/D Conversion Start Request Timing Register A	GTADTRA	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2774h	GPTW7	A/D Conversion Start Request Timing Buffer Register A	GTADTBRA	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2778h	GPTW7	A/D Conversion Start Request Timing Double-Buffer Register A	GTADTDBRA	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 277Ch	GPTW7	A/D Conversion Start Request Timing Register B	GTADTRB	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2780h	GPTW7	A/D Conversion Start Request Timing Buffer Register B	GTADTBRB	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2784h	GPTW7	A/D Conversion Start Request Timing Double-Buffer Register B	GTADTDBRB	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2788h	GPTW7	General PWM Timer Dead Time Control Register	GTDTCR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 278Ch	GPTW7	General PWM Timer Dead Time Value Register U	GTDVU	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2790h	GPTW7	General PWM Timer Dead Time Value Register D	GTDVD	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2794h	GPTW7	General PWM Timer Dead Time Value Buffer Register U	GTDBU	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2798h	GPTW7	General PWM Timer Dead Time Value Buffer Register D	GTDBD	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 279Ch	GPTW7	General PWM Timer Output Protection Function Status Register	GTSOS	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 27A0h	GPTW7	General PWM Timer Output Protection Function Temporary Release Register	GTSOTR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 27A4h	GPTW7	General PWM Timer A/D Conversion Start Request Signal Monitoring Register	GTADSMR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 27A8h	GPTW7	General PWM Timer Extended Interrupt Skipping Counter Control Register	GTEITC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 27ACh	GPTW7	General PWM Timer Extended Interrupt Skipping Setting Register 1	GTEITLI1	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 27B0h	GPTW7	General PWM Timer Extended Interrupt Skipping Setting Register 2	GTEITLI2	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 27B4h	GPTW7	General PWM Timer Extended Buffer Transfer Skipping Setting Register	GTEITLB	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 27B8h	GPTW7	General PWM Timer Inter Channel Logical Operation Function Setting Register	GTICLF	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 27BCh	GPTW7	General PWM Timer Cycle Count Register	GTPC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 27C0h	GPTW7	General PWM Timer A/D Conversion Start Request Compare Match Skipping Control Register	GTADCMSC	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 27C4h	GPTW7	General PWM Timer A/D Conversion Start Request Compare Match Skipping Setting Register	GTADCMSS	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 27D0h	GPTW7	General PWM Timer Operation Enable Bit Simultaneous Control Channel Select Register	GTSECSR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 27D4h	GPTW7	General PWM Timer Operation Enable Bit Simultaneous Control Register	GTSECR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 27E0h	GPTW7	General PWM Timer Buffer Enable Register 2	GTBER2	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 27E4h	GPTW7	General PWM Timer Output Level Buffer Register	GTOLBR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 27ECh	GPTW7	General PWM Timer Inter Channel Cooperation Input Capture Control Register	GTICCR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000C 2A00h	HRPWM	HRPWM Operation Control Register	HROCR	16	16	4, 5 PCLKA	2, 3 ICLK	section 25
000C 2A02h	HRPWM	HRPWM Operation Control Register 2	HROCR2	16	16	4, 5 PCLKA	2, 3 ICLK	section 25
000C 2A18h	HRPWM	GTIOC0A Pin Rising Edge Adjustment Register	HRREAR0A	16	16	4, 5 PCLKA	2, 3 ICLK	section 25
000C 2A1Ah	HRPWM	GTIOC0B Pin Rising Edge Adjustment Register	HRREAR0B	16	16	4, 5 PCLKA	2, 3 ICLK	section 25
000C 2A1Ch	HRPWM	GTIOC1A Pin Rising Edge Adjustment Register	HRREAR1A	16	16	4, 5 PCLKA	2, 3 ICLK	section 25
000C 2A1Eh	HRPWM	GTIOC1B Pin Rising Edge Adjustment Register	HRREAR1B	16	16	4, 5 PCLKA	2, 3 ICLK	section 25
000C 2A20h	HRPWM	GTIOC2A Pin Rising Edge Adjustment Register	HRREAR2A	16	16	4, 5 PCLKA	2, 3 ICLK	section 25

Table 5.1 List of I/O Registers (Address Order) (45 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000C 2A22h	HRPWM	GTIOC2B Pin Rising Edge Adjustment Register	HRREAR2B	16	16	4, 5 PCLKA	2, 3 ICLK	section 25
000C 2A24h	HRPWM	GTIOC3A Pin Rising Edge Adjustment Register	HRREAR3A	16	16	4, 5 PCLKA	2, 3 ICLK	section 25
000C 2A26h	HRPWM	GTIOC3B Pin Rising Edge Adjustment Register	HRREAR3B	16	16	4, 5 PCLKA	2, 3 ICLK	section 25
000C 2A28h	HRPWM	GTIOC0A Pin Falling Edge Adjustment Register	HRFEAR0A	16	16	4, 5 PCLKA	2, 3 ICLK	section 25
000C 2A2Ah	HRPWM	GTIOC0B Pin Falling Edge Adjustment Register	HRFEAR0B	16	16	4, 5 PCLKA	2, 3 ICLK	section 25
000C 2A2Ch	HRPWM	GTIOC1A Pin Falling Edge Adjustment Register	HRFEAR1A	16	16	4, 5 PCLKA	2, 3 ICLK	section 25
000C 2A2Eh	HRPWM	GTIOC1B Pin Falling Edge Adjustment Register	HRFEAR1B	16	16	4, 5 PCLKA	2, 3 ICLK	section 25
000C 2A30h	HRPWM	GTIOC2A Pin Falling Edge Adjustment Register	HRFEAR2A	16	16	4, 5 PCLKA	2, 3 ICLK	section 25
000C 2A32h	HRPWM	GTIOC2B Pin Falling Edge Adjustment Register	HRFEAR2B	16	16	4, 5 PCLKA	2, 3 ICLK	section 25
000C 2A34h	HRPWM	GTIOC3A Pin Falling Edge Adjustment Register	HRFEAR3A	16	16	4, 5 PCLKA	2, 3 ICLK	section 25
000C 2A36h	HRPWM	GTIOC3B Pin Falling Edge Adjustment Register	HRFEAR3B	16	16	4, 5 PCLKA	2, 3 ICLK	section 25
000C 2B00h	GPTW	Output Phase Switching Control Register	OPSCR	32	32	4, 5 PCLKA	2, 3 ICLK	section 24
000D 0100h	RSPI0	RSPI Control Register	SPCR	8	8	3, 4 PCLKA	1, 2 ICLK	section 37
000D 0101h	RSPI0	RSPI Slave Select Polarity Register	SSLP	8	8	3, 4 PCLKA	1, 2 ICLK	section 37
000D 0102h	RSPI0	RSPI Pin Control Register	SPPCR	8	8	3, 4 PCLKA	1, 2 ICLK	section 37
000D 0103h	RSPI0	RSPI Status Register	SPSR	8	8	3, 4 PCLKA	1, 2 ICLK	section 37
000D 0104h	RSPI0	RSPI Data Register	SPDR	32	8, 16, 32	3, 4 PCLKA	1, 2 ICLK	section 37
000D 0108h	RSPI0	RSPI Sequence Control Register	SPSCR	8	8	3, 4 PCLKA	1, 2 ICLK	section 37
000D 0109h	RSPI0	RSPI Sequence Status Register	SPSSR	8	8	3, 4 PCLKA	1, 2 ICLK	section 37
000D 010Ah	RSPI0	RSPI Bit Rate Register	SPBR	8	8	3, 4 PCLKA	1, 2 ICLK	section 37
000D 010Bh	RSPI0	RSPI Data Control Register	SPDCR	8	8	3, 4 PCLKA	1, 2 ICLK	section 37
000D 010Ch	RSPI0	RSPI Clock Delay Register	SPCKD	8	8	3, 4 PCLKA	1, 2 ICLK	section 37
000D 010Dh	RSPI0	RSPI Slave Select Negation Delay Register	SSLND	8	8	3, 4 PCLKA	1, 2 ICLK	section 37
000D 010Eh	RSPI0	RSPI Next-Access Delay Register	SPND	8	8	3, 4 PCLKA	1, 2 ICLK	section 37
000D 010Fh	RSPI0	RSPI Control Register 2	SPCR2	8	8	3, 4 PCLKA	1, 2 ICLK	section 37
000D 0110h	RSPI0	RSPI Command Register 0	SPCMD0	16	16	3, 4 PCLKA	1, 2 ICLK	section 37
000D 0112h	RSPI0	RSPI Command Register 1	SPCMD1	16	16	3, 4 PCLKA	1, 2 ICLK	section 37
000D 0114h	RSPI0	RSPI Command Register 2	SPCMD2	16	16	3, 4 PCLKA	1, 2 ICLK	section 37
000D 0116h	RSPI0	RSPI Command Register 3	SPCMD3	16	16	3, 4 PCLKA	1, 2 ICLK	section 37
000D 0118h	RSPI0	RSPI Command Register 4	SPCMD4	16	16	3, 4 PCLKA	1, 2 ICLK	section 37
000D 011Ah	RSPI0	RSPI Command Register 5	SPCMD5	16	16	3, 4 PCLKA	1, 2 ICLK	section 37
000D 011Ch	RSPI0	RSPI Command Register 6	SPCMD6	16	16	3, 4 PCLKA	1, 2 ICLK	section 37
000D 011Eh	RSPI0	RSPI Command Register 7	SPCMD7	16	16	3, 4 PCLKA	1, 2 ICLK	section 37
000D 0120h	RSPI0	RSPI Data Control Register 2	SPDCR2	8	8	3, 4 PCLKA	1, 2 ICLK	section 37
000D 0121h	RSPI0	RSPI Control Register 3	SPCR3	8	8	3, 4 PCLKA	1, 2 ICLK	section 37
000E 2080h	RSCI11	Receive Data Register	RDR	32	8, 16, 32	2, 3 PCLKA	2 ICLK	section 33
000E 2084h	RSCI11	Transmit Data Register	TDR	32	8, 16, 32	2, 3 PCLKA	2 ICLK	section 33
000E 2088h	RSCI11	Control Register 0	SCR0	32	32	2, 3 PCLKA	2 ICLK	section 33
000E 208Ch	RSCI11	Control Register 1	SCR1	32	32	2, 3 PCLKA	2 ICLK	section 33
000E 2090h	RSCI11	Control Register 2	SCR2	32	32	2, 3 PCLKA	2 ICLK	section 33
000E 2094h	RSCI11	Control Register 3	SCR3	32	32	2, 3 PCLKA	2 ICLK	section 33
000E 2098h	RSCI11	Control Register 4	SCR4	32	32	2, 3 PCLKA	2 ICLK	section 33
000E 209Eh	RSCI11	HBS Support Mode Control Register	HBSCR	8	8	2, 3 PCLKA	2 ICLK	section 33
000E 20A0h	RSCI11	I²C Mode Register	SIMR	32	32	2, 3 PCLKA	2 ICLK	section 33
000E 20A4h	RSCI11	FIFO Control Register	FCR	32	32	2, 3 PCLKA	2 ICLK	section 33
000E 20ACh	RSCI11	Manchester Mode Control Register	MMCR	32	32	2, 3 PCLKA	2 ICLK	section 33
000E 20B0h	RSCI11	DE Signal Control Register	DECR	32	32	2, 3 PCLKA	2 ICLK	section 33
000E 20B4h	RSCI11	Extended Serial Mode Control Register 0	XCR0	32	32	2, 3 PCLKA	2 ICLK	section 33
000E 20B8h	RSCI11	Extended Serial Mode Control Register 1	XCR1	32	32	2, 3 PCLKA	2 ICLK	section 33
000E 20BCh	RSCI11	Extended Serial Mode Control Register 2	XCR2	32	32	2, 3 PCLKA	2 ICLK	section 33

Table 5.1 List of I/O Registers (Address Order) (46 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000E 20C8h	RSCI11	Status Register	SSR	32	32	2, 3 PCLKA	2 ICLK	section 33
000E 20CCh	RSCI11	I2C Status Register	SISR	32	32	2, 3 PCLKA	2 ICLK	section 33
000E 20D0h	RSCI11	Receive FIFO Status Register	RFSR	32	32	2, 3 PCLKA	2 ICLK	section 33
000E 20D4h	RSCI11	Transmit FIFO Status Register	TFSR	32	32	2, 3 PCLKA	2 ICLK	section 33
000E 20D8h	RSCI11	Manchester Mode Status Register	MMSR	32	32	2, 3 PCLKA	2 ICLK	section 33
000E 20DCh	RSCI11	Extended Serial Mode Status Register 0	XSR0	32	32	2, 3 PCLKA	2 ICLK	section 33
000E 20E0h	RSCI11	Extended Serial Mode Status Register 1	XSR1	32	32	2, 3 PCLKA	2 ICLK	section 33
000E 20E8h	RSCI11	Status Clear Register	SSCR	32	32	2, 3 PCLKA	2 ICLK	section 33
000E 20ECh	RSCI11	I2C Status Clear Register	SISCR	32	32	2, 3 PCLKA	2 ICLK	section 33
000E 20F0h	RSCI11	Receive FIFO Status Clear Register	RFSCR	32	32	2, 3 PCLKA	2 ICLK	section 33
000E 20F4h	RSCI11	Manchester Mode Status Clear Register	MMSCR	32	32	2, 3 PCLKA	2 ICLK	section 33
000E 20F8h	RSCI11	Extended Serial Mode Status Clear Register	XSCR	32	32	2, 3 PCLKA	2 ICLK	section 33
000E 2800h	RSPIA0	RSPI Data Register	SPDR	32	8, 16, 32	2, 3 PCLKA	2 ICLK	section 38
000E 2804h	RSPIA0	RSPI Clock Delay Register	SPCKD	8	8	2, 3 PCLKA	2 ICLK	section 38
000E 2805h	RSPIA0	RSPI Slave Select Negation Delay Register	SSLND	8	8	2, 3 PCLKA	2 ICLK	section 38
000E 2806h	RSPIA0	RSPI Next-Access Delay Register	SPND	8	8	2, 3 PCLKA	2 ICLK	section 38
000E 2808h	RSPIA0	RSPI Control Register	SPCR	32	32	2, 3 PCLKA	2 ICLK	section 38
000E 280Ch	RSPIA0	RSPI Receive-Only Mode Control Register	SPRMCR	8	8	2, 3 PCLKA	2 ICLK	section 38
000E 280Dh	RSPIA0	RSPI Receive Data Ready Detect Condition Setting Register	SPDRCSR	8	8	2, 3 PCLKA	2 ICLK	section 38
000E 280Eh	RSPIA0	RSPI Pin Control Register	SPPCR	8	8	2, 3 PCLKA	2 ICLK	section 38
000E 2810h	RSPIA0	RSPI Slave Select Polarity Register	SSLP	8	8	2, 3 PCLKA	2 ICLK	section 38
000E 2811h	RSPIA0	RSPI Bit Rate Register	SPBR	8	8	2, 3 PCLKA	2 ICLK	section 38
000E 2813h	RSPIA0	RSPI Sequence Control Register	SPSCR	8	8	2, 3 PCLKA	2 ICLK	section 38
000E 2814h	RSPIA0	RSPI Command Register 0	SPCMD0	32	32	2, 3 PCLKA	2 ICLK	section 38
000E 2818h	RSPIA0	RSPI Command Register 1	SPCMD1	32	32	2, 3 PCLKA	2 ICLK	section 38
000E 281Ch	RSPIA0	RSPI Command Register 2	SPCMD2	32	32	2, 3 PCLKA	2 ICLK	section 38
000E 2820h	RSPIA0	RSPI Command Register 3	SPCMD3	32	32	2, 3 PCLKA	2 ICLK	section 38
000E 2824h	RSPIA0	RSPI Command Register 4	SPCMD4	32	32	2, 3 PCLKA	2 ICLK	section 38
000E 2828h	RSPIA0	RSPI Command Register 5	SPCMD5	32	32	2, 3 PCLKA	2 ICLK	section 38
000E 282Ch	RSPIA0	RSPI Command Register 6	SPCMD6	32	32	2, 3 PCLKA	2 ICLK	section 38
000E 2830h	RSPIA0	RSPI Command Register 7	SPCMD7	32	32	2, 3 PCLKA	2 ICLK	section 38
000E 2840h	RSPIA0	RSPI Data Control Register	SPDCR	16	16	2, 3 PCLKA	2 ICLK	section 38
000E 2844h	RSPIA0	RSPI FIFO Control Register	SPFCR	16	16	2, 3 PCLKA	2 ICLK	section 38
000E 2851h	RSPIA0	RSPI Sequence Status Register	SPSSR	8	8	2, 3 PCLKA	2 ICLK	section 38
000E 2852h	RSPIA0	RSPI Status Register	SPSR	16	16	2, 3 PCLKA	2 ICLK	section 38
000E 2858h	RSPIA0	RSPI Transmit FIFO Status Register	SPTFSR	8	8	2, 3 PCLKA	2 ICLK	section 38
000E 285Ch	RSPIA0	RSPI Receive FIFO Status Register	SPRFSR	8	8	2, 3 PCLKA	2 ICLK	section 38
000E 286Ah	RSPIA0	RSPI Status Clear Register	SPSCLR	16	16	2, 3 PCLKA	2 ICLK	section 38
000E 286Ch	RSPIA0	RSPI FIFO Clear Register	SPFCLR	8	8	2, 3 PCLKA	2 ICLK	section 38
000E C000h	RI3C0	Mode Register	ICMR	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C014h	RI3C0	Control Register	ICCR	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C018h	RI3C0	Controller Device Address Register	ICCAR	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C020h	RI3C0	Reset Control Register	ICRCR	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C024h	RI3C0	Operating Mode Monitor Register	ICMMR	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C030h	RI3C0	Internal Status Register	ICISR	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C034h	RI3C0	Internal Status Detection Enable Register	ICISER	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C038h	RI3C0	Internal Status Interrupt Enable Register	ICISIER	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C044h	RI3C0	Device Characteristics Table Index Register	ICDCTIR	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C058h	RI3C0	IBI Notify Control Register	ICINCR	32	32	2, 3 PCLKA	2 ICLK	section 35

Table 5.1 List of I/O Registers (Address Order) (47 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000E C064h	RI3C0	Target Mode Control Register	ICTCR	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C074h	RI3C0	Standard Bitrate Register	ICSBR	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C078h	RI3C0	Extended Bitrate Register	ICEBR	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C07Ch	RI3C0	Bus Free Time Setting Register	ICBFTR	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C080h	RI3C0	Bus Available Time Setting Register	ICBATR	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C084h	RI3C0	Bus Idle Time Setting Register	ICBITR	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C088h	RI3C0	Output Signal Control Register	ICOCR	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C090h	RI3C0	Timeout Control Register	ICTOR	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C0B0h	RI3C0	Clock Stall Control Register	ICSTCR	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C0C0h	RI3C0	Target Transmit/Receive Data Length Register	ICTDLR	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C150h	RI3C0	Command Queue Register	ICCQR	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C154h	RI3C0	Response Queue Register	ICRQR	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C158h	RI3C0	Transmit/Receive Data Register	ICDR	32	8, 32	2, 3 PCLKA	2 ICLK	section 35
000E C17Ch	RI3C0	IBI Queue Register	ICIQR	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C180h	RI3C0	Receive Status Queue Register	ICSQR	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C190h	RI3C0	Queue Buffer Threshold Control Register	ICQBTCR	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C194h	RI3C0	Data Buffer Threshold Control Register	ICDBTCR	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C1C0h	RI3C0	Receive Status Queue Threshold Control Register	ICSQTCR	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C1D0h	RI3C0	Status Register 2	ICSR2	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C1D4h	RI3C0	Status Detection Enable Register	ICSER	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C1D8h	RI3C0	Status Interrupt Enable Register	ICSIER	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C1E0h	RI3C0	Communication Status Register	ICCSR	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C1E4h	RI3C0	Communication Status Detection Enable Register	ICCSER	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C1E8h	RI3C0	Communication Status Interrupt Enable Register	ICCSIER	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C210h	RI3C0	Bus Status Register	ICBSR	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C224h	RI3C0	Target Device Address Table Register 0	ICTDATR0	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C22Ch	RI3C0	Target Device Address Table Register 1	ICTDATR1	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C234h	RI3C0	Target Device Address Table Register 2	ICTDATR2	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C23Ch	RI3C0	Target Device Address Table Register 3	ICTDATR3	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C2A0h	RI3C0	Extended Target Device Address Table Register	ICEDATR	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C2B0h	RI3C0	Device Address Register 0	ICDAR0	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C2D0h	RI3C0	Target Device Characteristics Table Register 0	ICTDCTR0	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C2D4h	RI3C0	Target Device Characteristics Table Register 1	ICTDCTR1	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C2D8h	RI3C0	Target Device Characteristics Table Register 2	ICTDCTR2	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C2DCh	RI3C0	Target Device Characteristics Table Register 3	ICTDCTR3	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C320h	RI3C0	Device Characteristics Table Register	ICDCTR	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C324h	RI3C0	Provisioned ID Low Register	ICPIDLR	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C328h	RI3C0	Provisioned ID High Register	ICPIDHR	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C330h	RI3C0	Device Address Monitor Register 0	ICDAMR0	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C350h	RI3C0	Target Event Register	ICTEVR	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C354h	RI3C0	Activity State Register	ICASR	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C358h	RI3C0	Max Write Length Register	ICMWLR	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C35Ch	RI3C0	Max Read Length Register	ICMRLR	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C360h	RI3C0	Test Mode Register	ICTMR	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C364h	RI3C0	Device Status Register	ICDSR	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C368h	RI3C0	Max Write Speed Register	ICMWSR	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C36Ch	RI3C0	Max Read Speed Register	ICMRSR	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C370h	RI3C0	Maximum Read Turnaround Time Register	ICMTTR	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C374h	RI3C0	Timing Support Information Register	ICTSIR	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C380h	RI3C0	Bit Count Register	ICBCR	32	32	2, 3 PCLKA	2 ICLK	section 35

Table 5.1 List of I/O Registers (Address Order) (48 / 48)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000E C394h	RI3C0	Queue Buffer Status Register	ICQBSR	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C398h	RI3C0	Data Buffer Status Register	ICDBSR	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C3C0h	RI3C0	Receive Status Queue Status Register	ICSQSR	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C3CCh	RI3C0	Internal Status Monitor Register	ICIMR	32	32	2, 3 PCLKA	2 ICLK	section 35
000E C3D0h	RI3C0	Controller Error Count Register	ICCECR	32	32	2, 3 PCLKA	2 ICLK	section 35
000E D000h	CANFD	ECC Control/Status Register	ECCSR	32	32	2, 3 PCLKA	1, 2 ICLK	section 36
000E D004h	CANFD	ECC Test Mode Register	ECTMR	16	16	2, 3 PCLKA	1, 2 ICLK	section 36
000E D00Ch	CANFD	ECC Decoder Test Data Register	ECTDR	32	32	2, 3 PCLKA	1, 2 ICLK	section 36
000E D010h	CANFD	ECC Error Address Register	ECEAR	32	32	2, 3 PCLKA	1, 2 ICLK	section 36
0012 0040h	OFSM	Serial Programmer Command Control Register	SPCC	32	32	8 FCLK		section 7
0012 0048h	OFSM	TM Enable Flag Register	TMEF	32	32	8 FCLK		section 7
0012 0050h	OFSM	OCD/Serial Programmer ID Setting Register	OSIS	128	32	8 FCLK		section 7
0012 0060h	OFSM	TM Identification Data Register	TMINF	32	32	8 FCLK		section 7
0012 0064h	OFSM	Endian Select Register	MDE	32	32	8 FCLK		section 7
0012 0068h	OFSM	Option Function Select Register 0	OFS0	32	32	8 FCLK		section 7
0012 006Ch	OFSM	Option Function Select Register 1	OFS1	32	32	8 FCLK		section 7
0012 0090h	OFSM	Bank Select Register	BANKSEL	32	32	8 FCLK		section 7
0012 00A0h	OFSM	Flash Access Window Setting Register	FAW	32	32	8 FCLK		section 7
007F B174h	FLASH	Unique ID Register 0	UIDR0	32	32	3 to 5 FCLK	3, 4 ICLK	section 48
007F B17Ch	TEMPS	Temperature Sensor Calibration Data Register	TSCDR	32	32	3 to 5 FCLK	3, 4 ICLK	section 44
007F B1E4h	FLASH	Unique ID Register 1	UIDR1	32	32	3 to 5 FCLK	3, 4 ICLK	section 48
007F B1E8h	FLASH	Unique ID Register 2	UIDR2	32	32	3 to 5 FCLK	3, 4 ICLK	section 48
007F E010h	FLASH	Flash Access Status Register	FASTAT	8	8	2 to 4 FCLK	2, 3 ICLK	section 48
007F E014h	FLASH	Flash Access Error Interrupt Enable Register	FAEINT	8	8	2 to 4 FCLK	2, 3 ICLK	section 48
007F E018h	FLASH	Flash Ready Interrupt Enable Register	FRDYIE	8	8	2 to 4 FCLK	2, 3 ICLK	section 48
007F E030h	FLASH	FACI Command Processing Start Address Register	FSADDR	32	32	2 to 4 FCLK	2, 3 ICLK	section 48
007F E034h	FLASH	FACI Command Processing End Address Register	FEADDR	32	32	2 to 4 FCLK	2, 3 ICLK	section 48
007F E080h	FLASH	Flash Status Register	FSTATR	32	32	2 to 4 FCLK	2, 3 ICLK	section 48
007F E084h	FLASH	Flash P/E Mode Entry Register	FENTRYR	16	16	2 to 4 FCLK	2, 3 ICLK	section 48
007F E08Ch	FLASH	Flash Sequencer Set-Up Initialization Register	FSUINITR	16	16	2 to 4 FCLK	2, 3 ICLK	section 48
007F E0A0h	FLASH	FACI Command Register	FCMDR	16	16	2 to 4 FCLK	2, 3 ICLK	section 48
007F E0C0h	FLASH	Flash P/E Status Register	FPESTAT	16	16	2 to 4 FCLK	2, 3 ICLK	section 48
007F E0D0h	FLASH	Data Flash Blank Check Control Register	FBCCNT	8	8	2 to 4 FCLK	2, 3 ICLK	section 48
007F E0D4h	FLASH	Data Flash Blank Check Status Register	FBCSTAT	8	8	2 to 4 FCLK	2, 3 ICLK	section 48
007F E0D8h	FLASH	Data Flash Programming Start Address Register	FPSADDR	32	32	2 to 4 FCLK	2, 3 ICLK	section 48
007F E0DCh	FLASH	Flash Access Window Monitor Register	FAWMON	32	32	2 to 4 FCLK	2, 3 ICLK	section 48
007F E0E0h	FLASH	Flash Sequencer Processing Switching Register	FCPSR	16	16	2 to 4 FCLK	2, 3 ICLK	section 48
007F E0E4h	FLASH	Flash Sequencer Processing Clock Frequency Notification Register	FPCKAR	16	16	2 to 4 FCLK	2, 3 ICLK	section 48
007F E0E8h	FLASH	Start-Up Area Control Register	FSUACR	16	16	2 to 4 FCLK	2, 3 ICLK	section 48

6. Resets

6.1 Overview

There are nine types of resets: RES# pin reset, power-on reset, voltage-monitoring 0 reset, voltage-monitoring 1 reset, voltage-monitoring 2 reset, independent watchdog timer reset, watchdog timer reset, and software reset.
Table 6.1 lists the reset names and sources.

Table 6.1 Reset Names and Sources

Reset Name	Source
RES# pin reset	Voltage input to the RES# pin is driven low.
Power-on reset	VCC rises (voltage detection: VPOR)*1
Voltage-monitoring 0 reset	VCC falls (voltage detection: Vdet0)*1
Voltage-monitoring 1 reset	VCC falls (voltage detection: Vdet1)*1
Voltage-monitoring 2 reset	VCC falls (voltage detection: Vdet2)*1
Independent watchdog timer reset	The independent watchdog timer underflows, or a refresh error occurs.
Watchdog timer reset	The watchdog timer underflows, or a refresh error occurs.
Software reset	Register setting

Note 1. For details on the voltages to be monitored (VPOR, Vdet0, Vdet1, and Vdet2), see section 8, Voltage Detection Circuit (LVDA) and section 49, Electrical Characteristics.

The internal state and pins are initialized by a reset.
Table 6.2 lists the reset targets to be initialized.

Table 6.2 Targets to be Initialized by Each Reset Source

Targets to be Initialized	Reset Source							
	RES# Pin Reset	Power-On Reset	Voltage-Monitoring 0 Reset	Independent Watchdog Timer Reset	Watchdog Timer Reset	Voltage-Monitoring 1 Reset	Voltage-Monitoring 2 Reset	Software Reset
Power-on reset detect flag (RSTSR0.PORF)	✓	—	—	—	—	—	—	—
Cold start/warm start determination flag (RSTSR1.CWSF)	—	✓	—	—	—	—	—	—
Voltage-monitoring 0 reset detect flag (RSTSR0.LVD0RF)	✓	✓	—	—	—	—	—	—
Voltage Level Setting Register (VOLSR)	✓	✓	✓	—	—	—	—	—
Independent watchdog timer reset detect flag (RSTSR2.IWDTRF)	✓	✓	✓	—	—	—	—	—
Independent watchdog timer (IWDTRR, IWDTCR, IWDTSR, IWDTRCR, IWDTCSTPR, ILOCOCR)	✓	✓	✓	—	—	—	—	—
Watchdog timer reset detect flag (RSTSR2.WDTRF)	✓	✓	✓	✓	—	—	—	—
Registers related to the watchdog timer (WDTRR, WDTCR, WDTSR, WDTRCR)	✓	✓	✓	✓	—	—	—	—
Voltage-monitoring 1 reset detect flag (RSTSR0.LVD1RF)	✓	✓	✓	✓	✓	—	—	—
Registers related to the voltage monitor function 1 (LVD1CR0, LVCMPCR.LVD1E, LVDLVLR.LVD1LVL[3:0])	✓	✓	✓	✓	✓	—	—	—
(LVD1CR1, LVD1SR)	✓	✓	✓	✓	✓	—	—	—
Voltage-monitoring 2 reset detect flag (RSTSR0.LVD2RF)	✓	✓	✓	✓	✓	✓	—	—
Registers related to the voltage monitor function 2 (LVD2CR0, LVCMPCR.LVD2E, LVDLVLR.LVD2LVL[3:0])	✓	✓	✓	✓	✓	✓	—	—
(LVD2CR1, LVD2SR)	✓	✓	✓	✓	✓	✓	—	—
Software reset detect flag (RSTSR2.SWRF)	✓	✓	✓	✓	✓	✓	✓	—
Register related to high-speed on-chip oscillator (HOCOPCR.HOCOPCNT)	✓	✓	✓	✓	✓	✓	✓	✓
Register related to main clock oscillator (MOFCR)	✓	✓	✓	✓	✓	✓	✓	✓
Pin state	✓	✓	✓	✓	✓	✓	✓	✓
Operating mode*1	✓	✓	✓	—	—	—	—	—
Registers other than the above, CPU, and internal state	✓	✓	✓	✓	✓	✓	✓	✓

✓: Targets to be initialized, —: No change occurs.

Note 1. The operating mode is determined by the level of the mode setting pin when the reset is released. For details, see section 3, Operating Modes.

When a reset is canceled, the reset exception handling starts. For details on the reset exception handling, see section 13, Exception Handling.

Table 6.3 lists the pin related to the reset.

Table 6.3 Pin Related to Reset

Pin Name	I/O	Function
RES#	Input	Reset pin

6.2 Register Descriptions

6.2.1 Reset Status Register 0 (RSTSR0)

Address(es): 0008 C290h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	LVD2RF	LVD1RF	LVD0RF	PORF
Value after reset:	0	0	0	0	0*1	0*1	0*1	0*1

Bit	Symbol	Bit Name	Description	R/W
b0	PORF	Power-On Reset Detect Flag	0: Power-on reset not detected. 1: Power-on reset detected.	R(/W) *2
b1	LVD0RF	Voltage-Monitoring 0 Reset Detect Flag	0: Voltage-monitoring 0 reset not detected. 1: Voltage-monitoring 0 reset detected.	R(/W) *2
b2	LVD1RF	Voltage-Monitoring 1 Reset Detect Flag	0: Voltage-monitoring 1 reset not detected. 1: Voltage-monitoring 1 reset detected.	R(/W) *2
b3	LVD2RF	Voltage-Monitoring 2 Reset Detect Flag	0: Voltage-monitoring 2 reset not detected. 1: Voltage-monitoring 2 reset detected.	R(/W) *2
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. The value after reset depends on the reset source.
Note 2. Only 0 can be written to clear the flag.

PORF Flag (Power-On Reset Detect Flag)

The PORF flag indicates that a power-on reset has occurred.
[Setting condition]

- When a power-on reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs.
- When PORF is read as 1 and then 0 is written to PORF.

LVD0RF Flag (Voltage-Monitoring 0 Reset Detect Flag)

The LVD0RF flag indicates that a voltage-monitoring 0 reset has occurred due to the VCC voltage falling below Vdet0.
[Setting condition]

- When a voltage-monitoring 0 reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs.
- When LVD0RF is read as 1 and then 0 is written to LVD0RF.

LVD1RF Flag (Voltage-Monitoring 1 Reset Detect Flag)

The LVD1RF flag indicates that a voltage-monitoring 1 reset has occurred due to the VCC voltage falling below Vdet1.

[Setting condition]

- When a voltage-monitoring 1 reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs.
- When LVD1RF is read as 1 and then 0 is written to LVD1RF.

LVD2RF Flag (Voltage-Monitoring 2 Reset Detect Flag)

The LVD2RF flag indicates that a voltage-monitoring 2 reset has occurred due to the VCC voltage falling below Vdet2.

[Setting condition]

- When a voltage-monitoring 2 reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs.
- When LVD2RF is read as 1 and then 0 is written to LVD2RF.

6.2.2 Reset Status Register 1 (RSTSR1)

Address(es): 0008 C291h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CWSF
Value after reset:	0	0	0	0	0	0	0	0/1*1

Bit	Symbol	Bit Name	Description	R/W
b0	CWSF	Cold/Warm Start Determination Flag	0: Cold start 1: Warm start	R(/W) *2
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. The value after reset depends on the reset source.
Note 2. Only 1 can be written to set the flag.

RSTSR1 determines whether a power-on reset has caused the reset processing (cold start) or a reset signal input during operation has caused the reset processing (warm start).

CWSF Flag (Cold/Warm Start Determination Flag)

The CWSF flag indicates the type of reset processing: cold start or warm start.
The CWSF flag is initialized by a power-on reset. It is not initialized by a reset signal generated by the RES# pin.
[Setting condition]

- When 1 is written through programming; it is not set to 0 even when 0 is written.

[Clearing condition]

- When a reset listed in Table 6.2 occurs.

6.2.3 Reset Status Register 2 (RSTSR2)

Address(es): 0008 00C0h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SWRF	WDTRF	IWDTRF
Value after reset:	0	0	0	0	0	0*1	0*1	0*1

Bit	Symbol	Bit Name	Description	R/W
b0	IWDTRF	Independent Watchdog Timer Reset Detect Flag	0: Independent watchdog timer reset not detected. 1: Independent watchdog timer reset detected.	R(/W) *2
b1	WDTRF	Watchdog Timer Reset Detect Flag	0: Watchdog timer reset not detected. 1: Watchdog timer reset detected.	R(/W) *2
b2	SWRF	Software Reset Detect Flag	0: Software reset not detected. 1: Software reset detected.	R(/W) *2
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. The value after reset depends on the reset source.
Note 2. Only 0 can be written to clear the flag.

IWDTRF Flag (Independent Watchdog Timer Reset Detect Flag)

The IWDTRF flag indicates that an independent watchdog timer reset has occurred.
[Setting condition]

- When an independent watchdog timer reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs.
- When IWDTRF is read as 1 and then 0 is written to IWDTRF.

WDTRF Flag (Watchdog Timer Reset Detect Flag)

The WDTRF flag indicates that a watchdog timer reset has occurred.
[Setting condition]

- When a watchdog timer reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs.
- When WDTRF is read as 1 and then 0 is written to WDTRF.

SWRF Flag (Software Reset Detect Flag)

The SWRF flag indicates that a software reset has occurred.
[Setting condition]

- When a software reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs.
- When SWRF is read as 1 and then 0 is written to SWRF.

6.2.4 Software Reset Register (SWRR)

Address(es): 0008 00C2h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SWRR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	SWRR[15:0]	Software Reset	Writing A501h resets the LSI. These bits are read as 0000h.	R/W

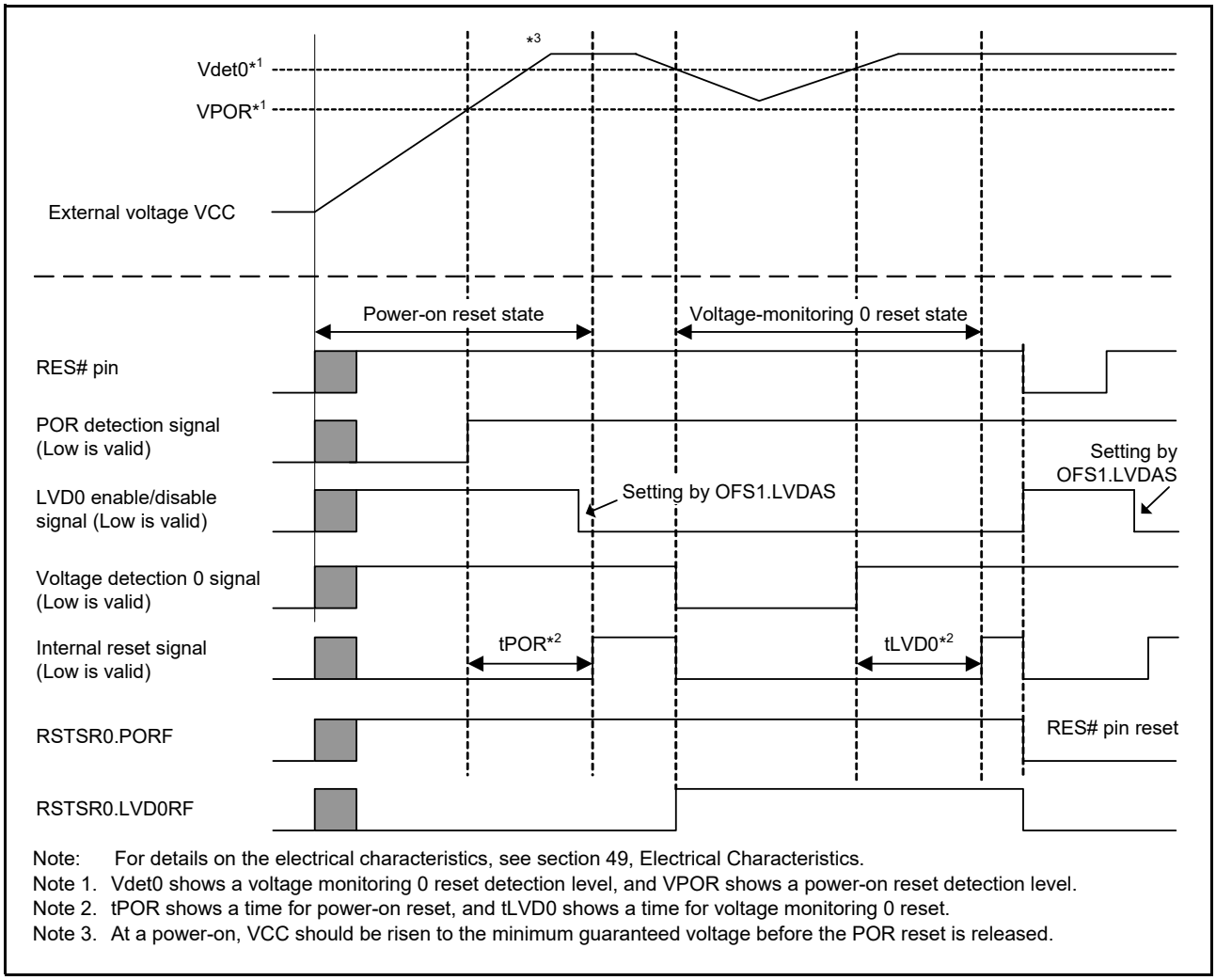
6.3 Operation

6.3.1 RES# Pin Reset

This is a reset generated by the RES# pin.
When the RES# pin is driven low, all the processing in progress is aborted and the LSI enters a reset state.
In order to unfailingly reset the LSI, the RES# pin should be held low for the specified power supply stabilization time at a power-on.
When the RES# pin is driven high from low, the internal reset is canceled after the post-RES# cancelation wait time (tRESWT) has elapsed, and then the CPU starts the reset exception handling.
For details, see section 49, Electrical Characteristics.

6.3.2 Power-On Reset and Voltage Monitoring 0 Reset

The power-on reset is an internal reset generated by the power-on reset circuit.
If the RES# pin is in a high level state when power is supplied, a power-on reset is generated. In addition, if the RES# pin is in a high level state when power falls (including the case when VCC falls below VPOR), a power-on reset is generated.
After VCC has exceeded VPOR and the specified period (power-on reset time) has elapsed, the internal reset is canceled and the CPU starts the reset exception handling. The power-on reset time is used for the stabilization of the power supply and the LSI circuit.
After a power-on reset has been generated, the PORF flag in the RSTSR0 is set to 1. The PORF flag is initialized by the RES# pin reset.
The voltage monitoring 0 reset is an internal reset generated by the voltage monitoring circuit. If the voltage detection 0 circuit start (LVDAS) bit in the option function select register 1 (OFS1) is 0 (voltage monitoring 0 reset is enabled after a reset) and VCC falls below Vdet0, the RSTSR0.LVD0RF flag becomes 1 and the voltage detection circuit generates voltage monitoring 0 reset. Clear the OFS1.LVDAS bit to 0 if the voltage monitoring 0 reset is to be used.
After VCC has exceeded Vdet0 and the voltage-monitoring 0 reset time (tLVD0) has elapsed, the internal reset is canceled and the CPU starts the reset exception handling. The Vdet0 voltage detection level can be changed by the setting of the VDSEL[1:0] bits in the option function select register 1 (OFS1).
Figure 6.1 shows operations during a power-on reset and voltage monitoring 0 reset.
For details on voltage monitoring 0 reset, refer to section 8, Voltage Detection Circuit (LVDA).



Note: For details on the electrical characteristics, see section 49, Electrical Characteristics.
Note 1. Vdet0 shows a voltage monitoring 0 reset detection level, and VPOR shows a power-on reset detection level.
Note 2. tPOR shows a time for power-on reset, and tLVD0 shows a time for voltage monitoring 0 reset.
Note 3. At a power-on, VCC should be risen to the minimum guaranteed voltage before the POR reset is released.

Figure 6.1 Operation Examples During a Power-On Reset and Voltage Monitoring 0 Reset

6.3.3 Voltage Monitoring 1 Reset and Voltage Monitoring 2 Reset

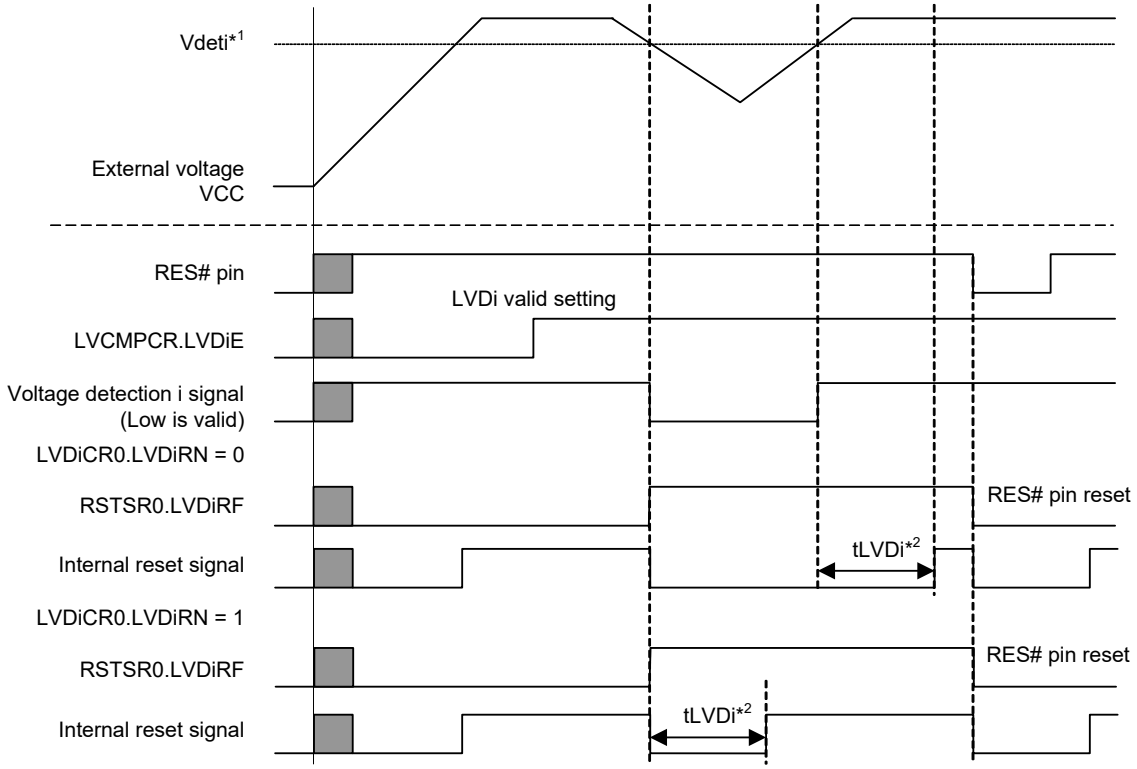
The voltage monitoring 1 reset and voltage monitoring 2 reset are internal resets generated by the voltage monitoring circuit.

When the voltage monitoring 1 interrupt/reset enable bit (LVD1RIE) is set to 1 (enabling generation of a reset or interrupt by the voltage detection circuit) and the voltage monitoring 1 circuit mode select bit (LVD1RI) is set to 1 (selecting generation of a reset in response to detection of a low voltage) in the voltage monitoring 1 circuit control register 0 (LVD1CR0), the RSTSR0.LVD1RF flag is set to 1 and the voltage-detection circuit generates a voltage monitoring 1 reset if VCC falls to or below Vdet1.

Likewise, when the voltage monitoring 2 interrupt/reset enable bit (LVD2RIE) is set to 1 (enabling generation of a reset or interrupt by the voltage detection circuit) and the voltage monitoring 2 circuit mode select bit (LVD2RI) is set to 1 (selecting generation of a reset in response to detection of a low voltage) in voltage monitoring 2 circuit control register 0 (LVD2CR0), the RSTSR0.LVD2RF flag is set to 1 and the voltage detection circuit generates a voltage monitoring 2 reset if VCC falls to or below Vdet2.

Timing for release from the voltage monitoring 1 reset state is selectable with the voltage monitoring 1 reset negate select bit (LVD1RN) in the LVD1CR0. When the LVD1CR0.LVD1RN bit is 0 and VCC has fallen to or below Vdet1, the CPU is released from the internal reset state and starts reset exception handling once the voltage-monitoring 1 reset time (tLVD1) has elapsed after VCC has risen above Vdet1. When the LVD1CR0.LVD1RN bit is 1 and VCC has fallen to or below Vdet1, the CPU is released from the internal reset state and starts reset exception handling once the voltage-

monitoring 1 reset time (tLVD1) has elapsed.
Likewise, timing for release from the voltage monitoring 2 reset state is selectable by setting the voltage monitoring 2 reset negate select bit (LVD2RN) in LVD2CR0 register.
Detection levels Vdet1 and Vdet2 can be changed by settings in the voltage detection select register (LVDLVLR).
Figure 6.2 shows examples of operations during voltage monitoring 1 and 2 resets.
For details on the voltage monitoring 1 reset and voltage monitoring 2 reset, refer to section 8, Voltage Detection Circuit (LVDA).



Note: For details on the electrical characteristics, see section 49, Electrical Characteristics.
Note 1. Vdeti shows a detection level of voltage monitoring 1 reset and voltage monitoring 2 reset. (i = 1, 2)
Note 2. tLVDi shows a time for voltage monitoring 1 reset and voltage monitoring 2 reset. (i = 1, 2)

Figure 6.2 Operation Examples During Voltage Monitoring 1 Reset and Voltage Monitoring 2 Reset

6.3.4 Independent Watchdog Timer Reset

Independent watchdog timer reset is an internal reset generated by the independent watchdog timer.
Output of the independent watchdog timer reset from the independent watchdog timer can be selected by settings in the IWDT reset control register (IWDTRCR) or option function select register 0 (OFS0).
When output of the independent watchdog timer reset is selected, an independent watchdog timer reset is generated if the independent watchdog timer underflows, or if data is written when refresh operation is disabled. When the internal reset time (tRESW2) has elapsed after the independent watchdog timer reset has been generated, the internal reset is canceled and the CPU starts the reset exception handling.
For details on the independent watchdog timer reset, see section 31, Independent Watchdog Timer (IWDTa).

6.3.5 Watchdog Timer Reset

The watchdog-timer reset is an internal reset from the watchdog timer.
Output of the independent watchdog timer reset from the independent watchdog timer can be selected by settings in the IWDT reset control register (IWDTRCR) or option function select register 0 (OFS0).
When output of the watchdog timer reset is selected, a watchdog timer reset is generated if the watchdog timer underflows, or if data is written when refresh operation is disabled. When the internal reset time (tRESW2) has elapsed after the watchdog timer reset is generated, the internal reset is canceled and the CPU starts the reset exception handling.
For details on the watchdog timer reset, see section 30, Watchdog Timer (WDTA).

6.3.6 Software Reset

The software reset is an internal reset generated by the software reset circuit.
When A501h is written to SWRR, a software reset is generated. When the internal reset time (tRESW2) has elapsed after the software reset is generated, the internal reset is canceled and the CPU starts the reset exception handling.

6.3.7 Determination of Cold/Warm Start

By reading the CWSF flag in RSTSR1, the type of reset processing caused can be identified; that is, whether a power-on reset has caused the reset processing (cold start) or a reset signal input during operation has caused the reset processing (warm start).

The CWSF flag in RSTSR1 is set to 0 when a power-on reset occurs (cold start); otherwise the flag is not set to 0. The flag is set to 1 when 1 is written to it through programming; it is not set to 0 even when 0 is written.

Figure 6.3 shows an example of cold/warm start determination operation.

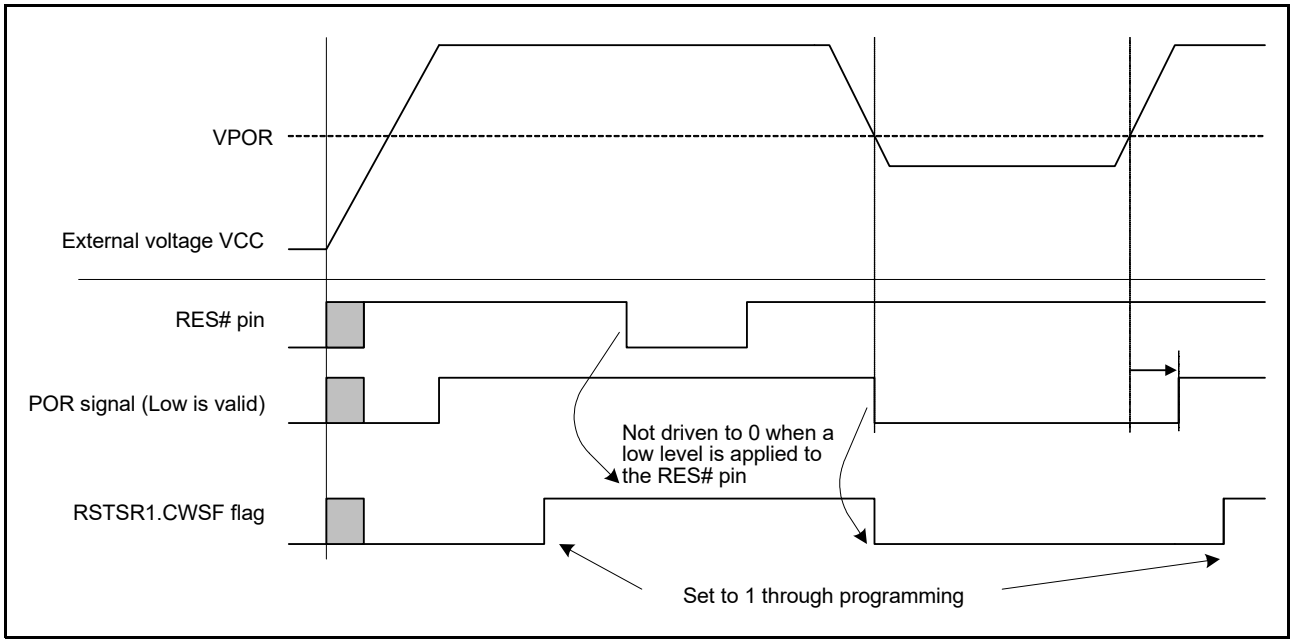


Figure 6.3 Example of Cold/Warm Start Determination Operation

6.3.8 Determination of Reset Generation Source

Reading RSTSR0 and RSTSR2 determines which reset was used to execute the reset exception handling.
Figure 6.4 shows an example of the flow to identify a reset generation source.

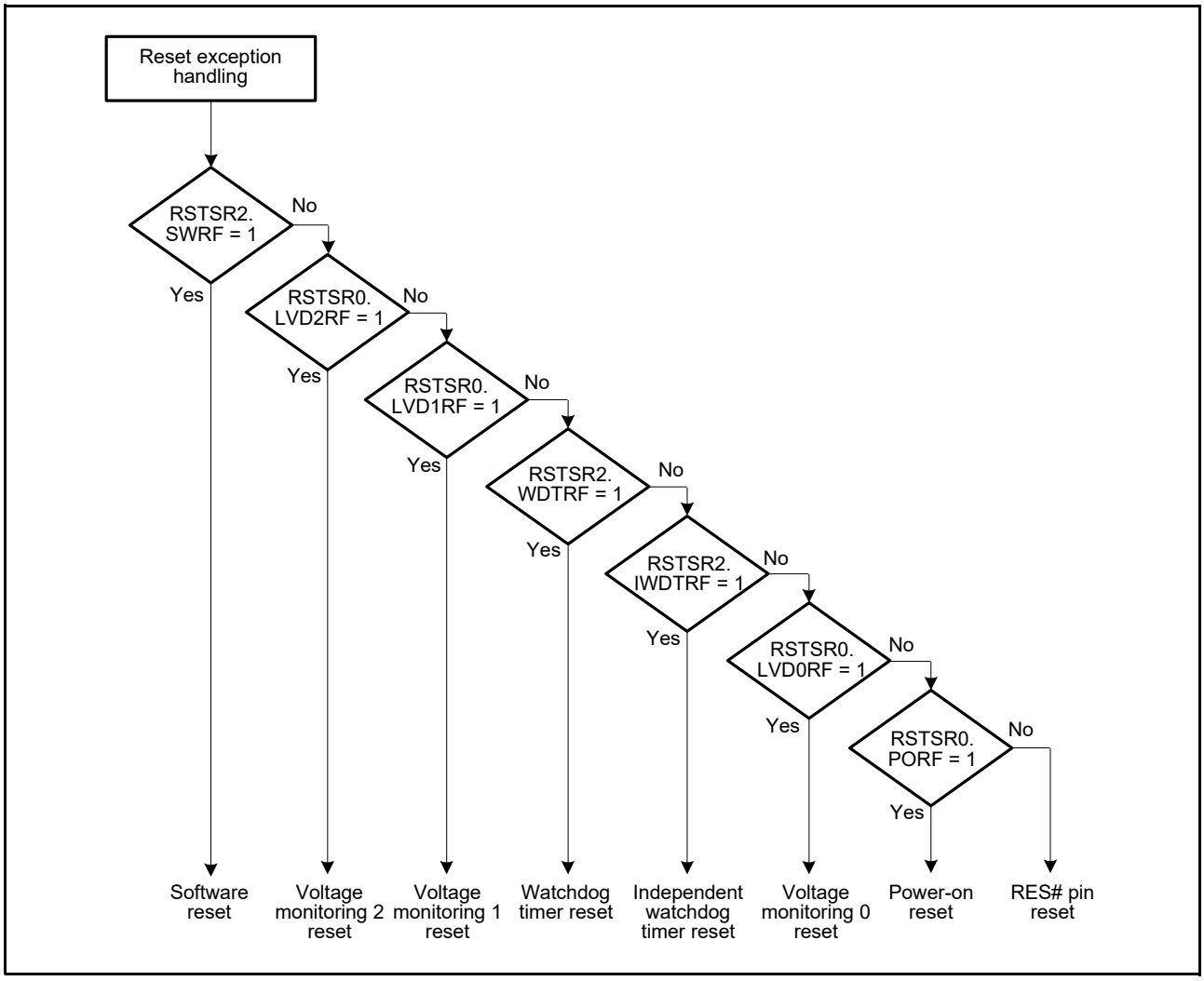


Figure 6.4 Example of Reset Generation Source Determination Flow

7. Option-Setting Memory (OFSM)

7.1 Overview

The option-setting memory (OFSM) is a collective term for the registers listed below.

- Serial programmer command control register (SPCC)
- OCD/serial programmer ID setting register (OSIS)
- Option function select register 0 (OFS0)
- Option function select register 1 (OFS1)
- Endian select register (MDE)
- TM enable flag register (TMEF)
- TM identification data register (TMINF)
- Bank select register (BANKSEL) (only for products with 512 Kbytes of code flash memory)
- Flash access window setting register (FAW)

The option-setting memory (configuration setting area) determines the state of this MCU after a reset.
The method of setting the option-setting memory is different from that of the I/O registers. For details, refer to section 7.5, Setting the Option-Setting Memory.
Figure 7.1 shows the option-setting memory.

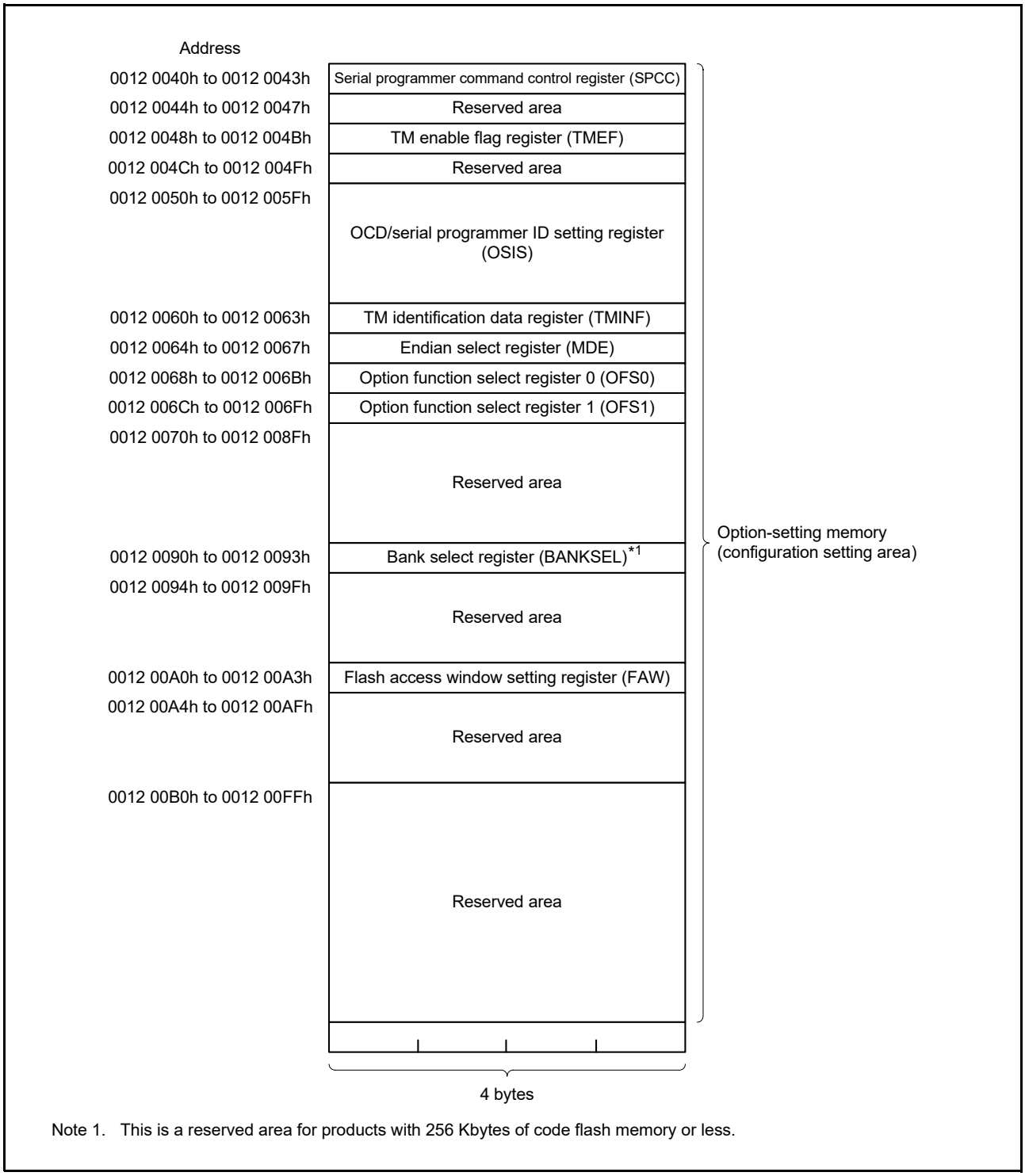


Figure 7.1 Option-Setting Memory

7.2 Register Descriptions

7.2.1 Serial Programmer Command Control Register (SPCC)

Address(es): OFSM.SPCC 0012 0040h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RDPR	WRPR	SEPR	—	SPE	—	—	IDE	—	—	—	—	—	—	OCDE	—
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	The value set by the user*1															

Bit	Symbol	Bit Name	Description	R/W
b16 to b0	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R
b17	OCDE	On-Chip Debugger Connection Enable	0: Connection to the on-chip debugger is prohibited after a reset. 1: Connection to the on-chip debugger is permitted after a reset.	R
b23 to b18	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R
b24	IDE	Serial Programmer ID Code Protection Enable	0: Serial programmer ID code protection is enabled after a reset.*2 1: Serial programmer ID code protection is disabled after a reset.*3	R
b26, b25	—	Reserved	When reading, this bit returns to the value written by the user. The write value should be 1.	R
b27	SPE	Serial Programmer Connection Enable	0: Connection of a serial programmer is prohibited after a reset. 1: Connection of a serial programmer is permitted after a reset.	R
b28	—	Reserved	When reading, this bit returns to the value written by the user. The write value should be 1.	R
b29	SEPR	Block Erase Command Protect*4	0: Execution of block erase commands is prohibited after a reset. 1: Execution of block erase commands is permitted after a reset.	R
b30	WRPR	Program Command Protect*4	0: Execution of program commands is prohibited after a reset. 1: Execution of program commands is permitted after a reset.	R
b31	RDPR	Read Command Protect*4	0: Execution of read commands is prohibited after a reset. 1: Execution of read commands is permitted after a reset.	R

Note 1. The value of the blank product is FFFF FFFFh. This register is set to a specified value after setting a value.
Note 2. When the serial programmer ID code protection is enabled, set the RDPR, WRPR, and SEPR bits to 0.
Note 3. When the serial programmer ID code protection is disabled, set the RDPR, WRPR, and SEPR bits to 1.
Note 4. Set these bits to the same value as the IDE bit.

This register is used to enable or disable the serial programmer ID code protection, and to permit or prohibit the connection to the on-chip debugger or the connection of a serial programmer.

OCDE Bit (On-Chip Debugger Connection Enable)

This bit enables or disables the connection to the on-chip debugger.

Resetting MCU after setting this bit to 0 disables the connection to the on-chip debugger.

IDE Bit (Serial Programmer ID Code Protection Enable)

This bit enables or disables the serial programmer ID code protection.
When setting this bit to 0 (serial programmer ID code protection is enabled), set the RDPR, WRPR, and SEPR bits to 0.
When setting this bit to 1 (serial programmer ID code protection is disabled), set the RDPR, WRPR, and SEPR bits to 1.

SPE Bit (Serial Programmer Connection Enable)

This bit enables or disables the connection of a serial programmer.
Resetting MCU after setting this bit to 0 disables the connection of a serial programmer.

SEPR Bit (Block Erase Command Protect)

This bit enables or disables execution of block erase commands by the serial programmer.
When the IDE bit is 0 and this bit is 0, the block erase command cannot be executed until the ID code matches.
When setting the IDE bit to 1, set this bit to 1. When setting the IDE bit to 0, set this bit to 0.

WRPR Bit (Program Command Protect)

This bit enables or disables execution of program commands by the serial programmer.
When the IDE bit is 0 and this bit is 0, the program command cannot be executed until the ID code matches.
When setting the IDE bit to 1, set this bit to 1. When setting the IDE bit to 0, set this bit to 0.

RDPR Bit (Read Command Protect)

This bit enables or disables execution of read commands by the serial programmer.
When the IDE bit is 0 and this bit is 0, the read command cannot be executed until the ID code matches.
When setting the IDE bit to 1, set this bit to 1. When setting the IDE bit to 0, set this bit to 0.

7.2.2 OCD/Serial Programmer ID Setting Register (OSIS)

This register is used to store the control code or ID code for the on-chip debugger ID code protection and serial programmer ID code protection.
After the emulator or serial programmer sends a control code or ID code, it is tested for a match with the value stored in this register.
Connection to the emulator/serial programmer can proceed if the codes match and cannot proceed if they do not.
Enabling the serial programmer ID code protection requires setting of the IDE, SPE, RDPR, WRPR, and SEPR bits in the SPCC register, in addition to setting of this register.
For the blank product, the value after a reset for ID code 1/control code to ID code 16 is FFh. These registers are set to the specified values after setting a value.

Address	Bit 31			Bit 0
0012 0050h to 0012 0053h	ID Code 4	ID Code 3	ID Code 2	ID Code 1/Control Code
0012 0054h to 0012 0057h	ID Code 8	ID Code 7	ID Code 6	ID Code 5
0012 0058h to 0012 005Bh	ID Code 12	ID Code 11	ID Code 10	ID Code 9
0012 005Ch to 0012 005Fh	ID Code 16	ID Code 15	ID Code 14	ID Code 13

ID Code 1/Control Code to ID Code 16

The control code or ID code for the on-chip debugger ID code protection and serial programmer ID code protection is stored in this register.
ID code 1 is used as a control code for connection to a serial programmer and as an ID code for connection to an emulator.
For details of the control code, refer to section 7.4, Settings of the Option-Setting Memory and Read, Program, and Erase Operations.

7.2.3 Option Function Select Register 0 (OFS0)

Address(es): OFSM.OFS0 0012 0068h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	WDTRSTIRQS	WDTRPSS[1:0]		WDTRPES[1:0]		WDTCKS[3:0]				WDTTOPS[1:0]		WDTSTRT	—
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	IWDTSLCSTP	—	IWDTRSTIRQS	IWDTRPSS[1:0]		IWDTRPES[1:0]		IWDTCKS[3:0]				IWDTTOPS[1:0]		IWDTSTRT	—
Value after reset:	The value set by the user*1															

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	When reading, this bit returns the value written by the user. The write value should be 1.	R
b1	IWDTSTRT	IWDT Start Mode Select	0: IWDT is automatically activated in auto-start mode after a reset 1: IWDT is halted after a reset	R
b3, b2	IWDTTOPS[1:0]	IWDT Timeout Period Select	b3 b2 0 0: 1024 cycles (03FFh) 0 1: 4096 cycles (0FFFh) 1 0: 8192 cycles (1FFFh) 1 1: 16384 cycles (3FFFh)	R
b7 to b4	IWDTCKS[3:0]	IWDT-Dedicated Clock Frequency Division Ratio Select	b7 b4 0 0 0 0: No division 0 0 1 0: Divide-by-16 0 0 1 1: Divide-by-32 0 1 0 0: Divide-by-64 1 1 1 1: Divide-by-128 0 1 0 1: Divide-by-256 Settings other than above are prohibited.	R
b9, b8	IWDTRPES[1:0]	IWDT Window End Position Select	b9 b8 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (No window end position setting)	R
b11, b10	IWDTRPSS[1:0]	IWDT Window Start Position Select	b11 b10 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (No window start position setting)	R
b12	IWDTRSTIRQS	IWDT Reset Interrupt Request Select	0: Non-maskable interrupt request or plain interrupt request is enabled 1: Reset is enabled	R
b13	—	Reserved	When reading, this bit returns the value written by the user. The write value should be 1.	R
b14	IWDTSLCSTP	IWDT Sleep Mode Count Stop Control	0: Counting stop is disabled 1: Counting stop is enabled when entering sleep, software standby, or all-module clock stop mode	R
b16, b15	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R
b17	WDTSTRT	WDT Start Mode Select	0: WDT is automatically activated in auto-start mode after a reset 1: WDT is stopped after a reset	R
b19, b18	WDTTOPS[1:0]	WDT Timeout Period Select	b19 b18 0 0: 1024 cycles (03FFh) 0 1: 4096 cycles (0FFFh) 1 0: 8192 cycles (1FFFh) 1 1: 16384 cycles (3FFFh)	R

Bit	Symbol	Bit Name	Description	R/W
b23 to b20	WDTCKS[3:0]	WDT Clock Frequency Division Ratio Select	b23 b20 0 0 0 1: Divide-by-4 0 1 0 0: Divide-by-64 1 1 1 1: Divide-by-128 0 1 1 0: Divide-by-512 0 1 1 1: Divide-by-2048 1 0 0 0: Divide-by-8192 Settings other than above are prohibited.	R
b25, b24	WDTRPES[1:0]	WDT Window End Position Select	b25 b24 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (No window end position setting)	R
b27, b26	WDTRPSS[1:0]	WDT Window Start Position Select	b27 b26 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (No window start position setting)	R
b28	WDTRSTIRQS	WDT Reset Interrupt Request Select	0: Non-maskable interrupt request or plain interrupt request is enabled 1: Reset is enabled	R
b31 to b29	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R

Note 1. The value of the blank product is FFFF FFFFh. This register is set to a specified value after setting a value.

IWDTSTRT Bit (IWDT Start Mode Select)

This bit selects the mode in which the IWDT is activated after a reset (stopped state or activated in auto-start mode). When activated in auto-start mode, the OFS0 register setting for the IWDT is effective.

IWDTTOPS[1:0] Bits (IWDT Timeout Period Select)

These bits select the timeout period, i.e. the time it takes for the down-counter to underflow, as 1024, 4096, 8192, or 16384 cycles of the frequency-divided clock set by the IWDTCKS[3:0] bits. The time (number of clock cycles for the IWDT) it takes to underflow after a refresh operation is determined by the combination of the IWDTCKS[3:0] bits and IWDTTOPS[1:0] bits.

For details, refer to section 31, Independent Watchdog Timer (IWDTa).

IWDTCKS[3:0] Bits (IWDT-Dedicated Clock Frequency Division Ratio Select)

These bits select, from 1/1, 1/16, 1/32, 1/64, 1/128, and 1/256, the division ratio of the prescaler to divide the frequency of the clock for the IWDT. Using the setting of these bits together with the IWDTTOPS[1:0] bit setting, the IWDT counting period can be set from 1024 to 4194304 clock cycles for the IWDT.

For details, refer to section 31, Independent Watchdog Timer (IWDTa).

IWDTRPES[1:0] Bits (IWDT Window End Position Select)

These bits select the position of the end of the window for the down-counter as 0%, 25%, 50%, or 75% of the value being counted by the counter. The value of the window end position must be smaller than the value of the window start position (window start position > window end position). If the value for the window end position is greater than the value for the window start position, only the value for the window start position is effective.

The counter values corresponding to the settings for the start and end positions of the window in the IWDTRPSS[1:0] and IWDTRPES[1:0] bits vary with the setting of the IWDTTOPS[1:0] bits.

For details, refer to section 31, Independent Watchdog Timer (IWDTa).

IWDTRPSS[1:0] Bits (IWDT Window Start Position Select)

These bits select the position where the window for the down-counter starts as 25%, 50%, 75%, or 100% of the value being counted (the point at which counting starts is 100% and the point at which an underflow occurs is 0%). The interval between the positions where the window starts and ends becomes the period in which refreshing is possible, and refreshing is not possible outside this period.

For details, refer to section 31, Independent Watchdog Timer (IWDTa).

IWDTRSTIRQS Bit (IWDT Reset Interrupt Request Select)

The setting of this bit selects the operation on an underflow of the down-counter or generation of a refresh error. An independent watchdog timer reset, a non-maskable interrupt request, or an interrupt request is selectable.

For details, refer to section 31, Independent Watchdog Timer (IWDTa).

IWDTSLCSTP Bit (IWDT Sleep Mode Count Stop Control)

This bit selects to stop counting when entering sleep, software standby, or all-module clock stop mode.

For details, refer to section 31, Independent Watchdog Timer (IWDTa).

WDTSTRT Bit (WDT Start Mode Select)

This bit selects the mode in which the WDT is activated after a reset (stopped state or activated in auto-start mode). When activated in auto-start mode, the OFS0 register setting for the WDT is effective.

WDTTOPS[1:0] Bits (WDT Timeout Period Select)

These bits select the timeout period, i.e. the time it takes for the down-counter to underflow, as 1024, 4096, 8192, or 16384 cycles of the frequency-divided clock set by the WDTCKS[3:0] bits. The time (number of PCLKB cycles) it takes to underflow after a refresh operation is determined by a combination of the WDTCKS[3:0] bits and WDTTOPS[1:0] bits.

For details, refer to section 30, Watchdog Timer (WDTA).

WDTCKS[3:0] Bits (WDT Clock Frequency Division Ratio Select)

These bits select, from 1/4, 1/64, 1/128, 1/512, 1/2048, and 1/8192, the division ratio of the prescaler to divide the frequency of PCLKB. Using the setting of these bits together with the WDTTOPS[1:0] bit setting, the WDT counting period can be set from 4096 to 134217728 PCLKB cycles.

For details, refer to section 30, Watchdog Timer (WDTA).

WDTRPES[1:0] Bits (WDT Window End Position Select)

These bits select the position of the end of the window on the down-counter as 0%, 25%, 50%, or 75% of the value being counted by the counter. The value of the window end position must be smaller than the value of the window start position (window start position > window end position). If the value for the window end position is greater than the value for the window start position, only the value for the window start position is effective.

The counter values corresponding to the settings for the start and end positions of the window in the WDTRPSS[1:0] and WDTRPES[1:0] bits vary with the setting of the WDTTOPS[1:0] bits.

For details, refer to section 30, Watchdog Timer (WDTA).

WDTRPSS[1:0] Bits (WDT Window Start Position Select)

These bits select the position where the window for the down-counter starts as 25%, 50%, 75%, or 100% of the value being counted (the point at which counting starts is 100% and the point at which an underflow occurs is 0%). The interval between the positions where the window starts and ends becomes the period in which refreshing is possible, and refreshing is not possible outside this period.

For details, refer to section 30, Watchdog Timer (WDTA).

WDTRSTIRQS Bit (WDT Reset Interrupt Request Select)

The setting of this bit selects the operation on an underflow of the down-counter or generation of a refresh error. A watchdog timer reset, a non-maskable interrupt request, or an interrupt request is selectable.
For details, refer to section 30, Watchdog Timer (WDTA).

7.2.4 Option Function Select Register 1 (OFS1)

Address(es): OFSM.OFS1 0012 006Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	HOCOEN	—	—	—	—	—	LVDAS	VDSEL[1:0]	
Value after reset:	The value set by the user*1															

Bit	Symbol	Bit Name	Description	R/W
b1, b0	VDSEL[1:0]	Voltage Detection 0 Level Select	b1 b0 0 0: Reserved 0 1: Reserved 1 0: Selects 2.83 V 1 1: Selects 4.22 V	R
b2	LVDAS	Voltage Detection 0 Circuit Start	0: Voltage monitor 0 reset is enabled after a reset 1: Voltage monitor 0 reset is disabled after a reset	R
b7 to b3	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R
b8	HOCOEN	HOCO Oscillation Enable	0: HOCO oscillation is enabled after a reset 1: HOCO oscillation is disabled after a reset	R
b31 to b9	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R

Note 1. The value of the blank product is FFFF FFFFh. This register is set to a specified value after setting a value.

VDSEL[1:0] Bits (Voltage Detection 0 Level Select)

These bits select the voltage detection level of the voltage detection 0 circuit.

LVDAS Bit (Voltage Detection 0 Circuit Start)

This bit selects whether the voltage monitor 0 reset is enabled or disabled after a reset.

HOCOEN Bit (HOCO Oscillation Enable)

This bit selects whether the HOCO oscillation enable bit is effective or not after a reset.

Setting the HOCOEN bit to 0 allows the HOCO oscillation to be started before the CPU starts operation, and therefore reduces the waiting time for oscillation stabilization.

Note that even if the HOCOEN bit is set to 0, the system clock source is not switched to HOCO. The system clock source is switched to HOCO only by modifying the clock source select bits (SCKCR3.CKSEL[2:0]) from the CPU.

7.2.5 Endian Select Register (MDE)

Address(es): OFSM.MDE 0012 0064h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	BANKMD[2:0]			—	MDE[2:0]		
Value after reset:	The value set by the user*1															

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	MDE[2:0]	Endian Select	b2 b0 0 0 0: Big endian 1 1 1: Little endian Settings other than above are prohibited.	R
b3	—	Reserved	When reading, this bit returns the value written by the user. The write value should be 1.	R
b6 to b4	BANKMD[2:0]	Bank Mode Select*2	b6 b4 0 0 0: Dual mode 1 1 1: Linear mode Settings other than above are prohibited.	R
b31 to b7	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R

Note 1. The value of the blank product is FFFF FFFFh. This register is set to a specified value after setting a value.
Note 2. Dual mode cannot be used for products with 256 Kbytes of code flash memory or less. Set these bits to 111b (linear mode).

This register selects the endian for the CPU and bank mode of the dual bank function of the code flash memory.

MDE[2:0] Bits (Endian Select)

These bits select little endian or big endian for the CPU.

BANKMD[2:0] Bits (Bank Mode Select)

These bits select bank mode of the dual bank function of the code flash memory. These bits cannot be rewritten while the TM function is enabled. Set the value of these bits while the TM function is disabled.

7.2.6 TM Enable Flag Register (TMEF)

Address(es): OFSM.TMEF 0012 0048h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	TMEFDB[2:0]			—	TMEF[2:0]			—	—	—	—	—	—	—	—
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	The value set by the user*1															

Bit	Symbol	Bit Name	Description	R/W
b23 to b0	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R
b26 to b24	TMEF[2:0]	TM Enable	b26 b24 0 0 0: The TM function for blocks 8 and 9 in the code flash memory is enabled. 1 1 1: The TM function for blocks 8 and 9 in the code flash memory is disabled. Settings other than above are prohibited.	R
b27	—	Reserved	When reading, this bit returns the value written by the user. The write value should be 1.	R
b30 to b28	TMEFDB[2:0]	Dual-Bank TM Enable	b30 b28 0 0 0: The TM function for blocks 30 and 31 in the code flash memory is enabled in dual mode. 1 1 1: The TM function for blocks 30 and 31 in the code flash memory is disabled in dual mode. Settings other than above are prohibited.	R
b31	—	Reserved	When reading, this bit returns the value written by the user. The write value should be 1.	R

Note 1. The value of the blank product is FFFF FFFFh. This register is set to a specified value after setting a value.

This register is used to enable the TM function in the code flash memory.
To enable the TM function, refer to section 48.8.22, Configuration Program Command. When the TMEF[2:0] bits or TMEFDB[2:0] bits are rewritten with the TM function enabled, rewriting the bits is ignored.
To disable the TM function, refer to section 48.8.21, Configuration Clear Command.

TMEF[2:0] Bits (TM Enable)

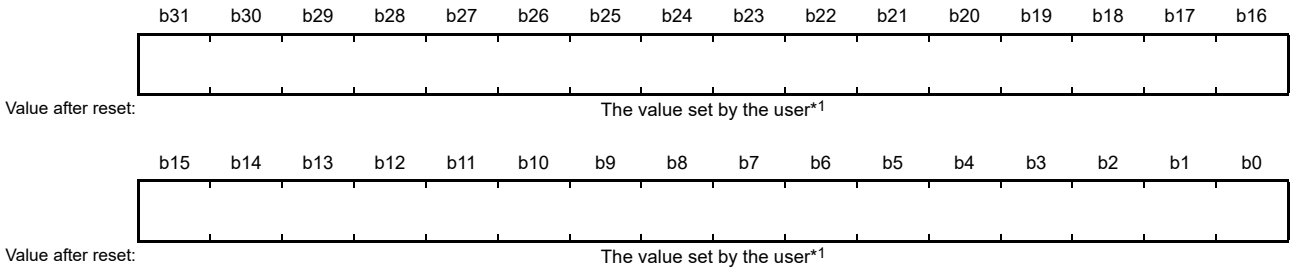
These bits enable or disable the TM function in the code flash memory. When enabling the TMEF[2:0] bits (000b) in dual mode, also enable the TMEFDB[2:0] (000b).

TMEFDB[2:0] Bits (Dual-Bank TM Enable)

These bits enable or disable the TM function of the code flash memory in dual mode. Set these bits to disable dual mode (111b) if linear mode is to be used.
When enabling the TMEFDB[2:0] bits, also enable the TMEF[2:0] bits.
When the TMEF[2:0] bits are 111b (with the TM function disabled), this setting is disabled.

7.2.7 TM Identification Data Register (TMINF)

Address(es): OFSM.TMINF 0012 0060h



Note 1. The value of the blank product is FFFF FFFFh. This register is set to a specified value after setting a value.

The user can store any desired 32-bit value in this register.
This register is used to store codes that identify the program stored in the TM-target area.
When the TMINF register is rewritten by serial programming while the TM function is enabled, rewriting this register is ignored. To erase the contents of the TMINF register, refer to section 48.8.21, Configuration Clear Command.

7.2.8 Bank Select Register (BANKSEL)

Address(es): OFSM.BANKSEL 0012 0090h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	BANKSWP[2:0]		
Value after reset:	The value set by the user*1															

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	BANKSWP[2:0]	Startup Bank Switch	b2 b0 0 0 0: These bits specify the address range of bank 1 from FFFC 0000h to FFFF FFFFh and bank 0 from FFF8 0000h to FFFB FFFFh. 1 1 1: These bits specify the address range of bank 1 from FFF8 0000h to FFFB FFFFh and bank 0 from FFFC 0000h to FFFF FFFFh. Settings other than above are prohibited.	R
b31 to b3	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R

Note 1. The value of the blank product is FFFF FFFFh. This register is set to a specified value after setting a value.

This register selects the startup bank of the program when the code flash memory is in dual mode.

BANKSWP[2:0] Bits (Startup Bank Switch)

These bits select the address range of the bank 0 and bank 1 of the code flash memory when the code flash memory is in dual mode.

Selecting the addresses of bank 0 and bank 1 selects the program to be started up.

The setting of these bits in linear mode is invalid.

For details of the startup bank selection, refer to section 48.5.6.2, Selecting the Startup Bank.

7.2.9 Flash Access Window Setting Register (FAW)

Address(es): OFSM.FAW 0012 00A0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	BTFLG	—	—	FAWE[12:0]												
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FSPR	—	—	FAWS[12:0]												
Value after reset:	The value set by the user*1															

Bit	Symbol	Bit Name	Description	R/W
b12 to b0	FAWS[12:0]	Flash Access Window Start Address*2	Flash access window start address	R
b14, b13	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R
b15	FSPR	Access Window Protection	0: With protection 1: Without protection	R
b28 to b16	FAWE[12:0]	Flash Access Window End Address*2	Flash access window end address	R
b30, b29	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R
b31	BTFLG	Start-up Area Select	0: FFFF 8000h to FFFF BFFFh are used as the start-up area 1: FFFF C000h to FFFF FFFFh are used as the start-up area	R

Note 1. The value of the blank product is FFFF FFFFh. This register is set to a specified value after setting a value.
Note 2. Set the value of the FAWS[12:0] bits to be no greater than that of the FAWE[12:0] bits.
If the value of the FAWS[12:0] bits is equal to that of the FAWE[12:0] bits, the entire user area of the code flash memory becomes programmable and erasable.

This register is used to set the write protection bit and start-up area select bit for setting the flash access window start address, flash access window end address, and access window.

FAWS[12:0] Bits (Flash Access Window Start Address)

These bits are used to set the access window start address.
The access window can be set in block units.
Set the FAWS[11:0] bits to b23 to b12 of the access window start address. Set the FAWS[12] bit to 0.

FSPR Bit (Access Window Protection)

Setting the FSPR bit protects the following operations.

- Setting the areas including the FAW register by using the configuration setting command of the FACI commands.
- Setting the areas including the FAW register by using the configuration program command in boot mode
- Erasing the option-setting memory by using the configuration clearing command in boot mode
- Changing the setting of the start-up area protection by using the FSUACR register.
- Erasing the all data in the flash memory when results of judgment did not match 3 times in a row while the control code is set to 45h in boot mode.

Once 0 is written to this bit, the bit can never be restored to 1.
Therefore, the access window and the start-up area protection never be set again or the TM function never be disabled once it has been enabled.

Exercise extra caution when handling the FSPR bit.

FAWE[12:0] Bits (Flash Access Window End Address)

These bits are used to set the access window end address.
The access window can be set in block units.
Set the FAWE[11:0] bits to b23 to b12 of the access window end address. Set the FAWE[12] bit to 0. However, if the access window end address is 1 0000 0000h (= FFFF FFFFh + 1), set the FAWE[12:0] bits to 1000h.

BTFLG Bit (Start-up Area Select)

This bit is used to set whether the start-up area is switched by using the start-up program protection. In dual mode (the MDE.BANKMD[2:0] bits are 000b), write 1 to this bit.
For details, refer to section 48.5.4, Start-Up Program Protection.

7.3 Programming and Erasing of the Option-Setting Memory in Individual Operating Modes

Table 7.1 shows programming and erasing of the option-setting memory in the individual operating modes.

Table 7.1 Programming and Erasing of the Option-Setting Memory in Individual Operating Modes

Option-Setting Memory	Boot Mode (SCI Interface, FINE Interface)		Self-Programming	
	Program	Erase	Program	Erase
SPCC register, OSIS register, MDE register, OFS0 register, OFS1 register, TMEF register, TMINF register, BANKSEL register, and FAW register	✓*1	✓*1	✓*2	×

✓: Possible

×: Not possible

Note 1. The commands for boot mode (SCI and FINE interfaces) are used for programming or erasing. For details, refer to section 48.7, Boot Mode.

Note 2. The configuration setting command is used for programming. For how to use the configuration setting command, refer to section 48.6.7.10, Configuration Set Command.

7.4 Settings of the Option-Setting Memory and Read, Program, and Erase Operations

Table 7.2 shows the settings of the option-setting memory and read, program, and erase operations when the MCU is connected to a serial programmer.
Table 7.3 shows the settings of the option-setting memory and judgment on ID codes when the MCU is connected to an OCD.

Table 7.2 Settings of the Option-Setting Memory and Read, Program, and Erase Operations When the MCU is Connected to a Serial Programmer

No.	SPCC. SPE	SPCC. IDE	OSIS (Control Code)	OSIS (ID code 2 to 16)	SPCC. RDPR	SPCC. WRPR	SPCC. SEPR	Connection of a Serial Programmer	Read, Program and Erase Operations after Connecting a Serial Programmer
1	0	x	Any value	Any value	x	x	x	Connection prohibited	—
2	1	0	Other than 45h		0	0	0	Judgment on control codes and ID codes*1	Permitted: read, program, erase
3			45h					Judgment on control codes and ID codes*2	Permitted: read, program, erase
4	1	1	Any value		1	1	1	Connection permitted	Permitted: read, program, erase

x: Don't care

Note 1. This determines whether the control code or ID code sent by the serial programmer matches the control code or ID code set in the OSIS register. When the codes match, connection is permitted; if not, connection is not possible.

Note 2. This determines whether the control code or ID code sent by the serial programmer matches the control code or ID code set in the OSIS register. When the codes match, connection is permitted; if not, connection is not possible. However, when results of judgment do not match 3 times in a row, the all data in the flash memory will be erased.

Table 7.3 Settings of the Option-Setting Memory and Judgment on ID Codes When the MCU is Connected to an OCD

No.	SPCC. SPE	SPCC. IDE	SPCC. OCDE	OSIS (ID Code 1)	OSIS (ID Code 2 to 16)	SPCC. RDPR	SPCC. WRPR	SPCC. SEPR	Connection to an OCD
1	x	x	1	Any value	Any value	x	x	x	ID codes matched: Connection to an OCD is permitted ID codes unmatched: Waiting for input of ID code
2	x	x	0	—	—	x	x	x	Connection to an OCD is prohibited (this is independent of the state of ID code matching).

x: Don't care

7.5 Setting the Option-Setting Memory

7.5.1 Allocation of Data in the Option-Setting Memory

Data for programming in the option-setting memory should be allocated to the addresses shown in Figure 7.1.
An example of source code for setting the option-setting memory is shown below.

Note: Programming formats vary depending on the compiler. Refer to the compiler manual for details.

Setting 1EFFFFFFh in the serial programmer command control register (SPCC)
.ORG 00120040H
.LWORD 1EFFFFFFH

Setting the following ID codes in the OCD/serial programmer ID setting register (OSIS)
ID code 1/control code = FFh, ID code 2 = 02h, ID code 3 = 03h, ID code 4 = 04h,
ID code 5 = 05h, ID code 6 = 06h, ID code 7 = 07h, ID code 8 = 08h,
ID code 9 = 09h, ID code 10 = 0Ah, ID code 11 = 0Bh, ID code 12 = 0Ch,
ID code 13 = 0Dh, ID code 14 = 0Eh, ID code 15 = 0Fh, ID code 16 = 10h
.ORG 00120050H
.LWORD 040302FFH, 08070605H, 0C0B0A09H, 100F0E0DH

Setting EF67BA5Dh in the option function select register 0 (OFS0)
.ORG 00120068H
.LWORD 0EF67BA5DH

Setting FFFFFEFAh in the option function select register 1 (OFS1)
.ORG 0012006CH
.LWORD 0FFFFFEFAH

Setting FFFFFFF8h in the endian select register (MDE)
.ORG 00120064H
.LWORD 0FFFFFFF8H

Setting EFFC6FF9h in the flash access window setting register (FAW)
.ORG 001200A0H
.LWORD 0EFFC6FF9H

7.6 Usage Note

7.6.1 Data for Programming Reserved Areas and Reserved Bits in the Option-Setting Memory

When reserved areas and reserved bits in the option-setting memory are within the scope of programming, write 1 as the value for all bits of reserved areas and all reserved bits. Normal operation cannot be guaranteed if 0 is written to such bits.

8. Voltage Detection Circuit (LVDA)

The voltage detection circuit (LVDA) monitors the voltage level input to the VCC pin using a program.

8.1 Overview

For voltage detection 0, the detection voltage is selectable from among two different levels and the reset from voltage monitoring 0 can be enabled or disabled after a reset by using the option function select register 1 (OFS1).
For voltage detection 1 and voltage detection 2, the detection voltage is selectable from among five different levels by using the voltage detection level select register (LVDLVLR).
The reset from voltage monitoring 0, reset/interrupt from voltage monitoring 1, and reset/interrupt from voltage monitoring 2 can be used.

Table 8.1 lists the specifications of the voltage detection circuit. Figure 8.1 is a block diagram of the voltage detection circuit. Figure 8.2 is a block diagram of the voltage monitoring 1 interrupt/reset circuit. Figure 8.3 is a block diagram of the voltage monitoring 2 interrupt/reset circuit.

Table 8.1 Voltage Detection Circuit Specifications

Item		Voltage Monitoring 0	Voltage Monitoring 1	Voltage Monitoring 2
VCC monitoring	Monitored voltage	Vdet0	Vdet1	Vdet2
	Detection voltage	Selectable from among two different levels by using OFS1.VDSEL[1:0] bits	Selectable from among five different levels by using LVDLVLR.LVD1LVL[3:0] bits	Selectable from among five different levels by using LVDLVLR.LVD2LVL[3:0] bits
	Monitoring flag	None	LVD1SR.LVD1MON flag: Monitors whether voltage is higher or lower than Vdet1	LVD2SR.LVD2MON flag: Monitors whether voltage is higher or lower than Vdet2
			LVD1SR.LVD1DET flag: Vdet1 passage detection	LVD2SR.LVD2DET flag: Vdet2 passage detection
Process upon voltage detection	Reset	Voltage monitoring 0 reset	Voltage monitoring 1 reset	Voltage monitoring 2 reset
		Reset when Vdet0 > VCC CPU restart after specified time with VCC > Vdet0	Reset when Vdet1 > VCC CPU restart timing selectable: after specified time with VCC > Vdet1 or Vdet1 > VCC	Reset when Vdet2 > VCC CPU restart timing selectable: after specified time with VCC > Vdet2 or Vdet2 > VCC
	Interrupt	No interrupt	Voltage monitoring 1 interrupt	Voltage monitoring 2 interrupt
			Non-maskable interrupt or maskable interrupt selectable	Non-maskable interrupt or maskable interrupt selectable
			Interrupt request issued when Vdet1 > VCC and VCC > Vdet1 or either	Interrupt request issued when Vdet2 > VCC and VCC > Vdet2 or either
Digital filter	Enable/ Disable switching	Digital filter function not available	Available	Available
	Sampling time	—	1/n LOCO frequency × 2 (n: 2, 4, 8, 16)	1/n LOCO frequency × 2 (n: 2, 4, 8, 16)
Event linking		None	Available Output of event signals on detection of Vdet crossings	Available Output of event signals on detection of Vdet crossings

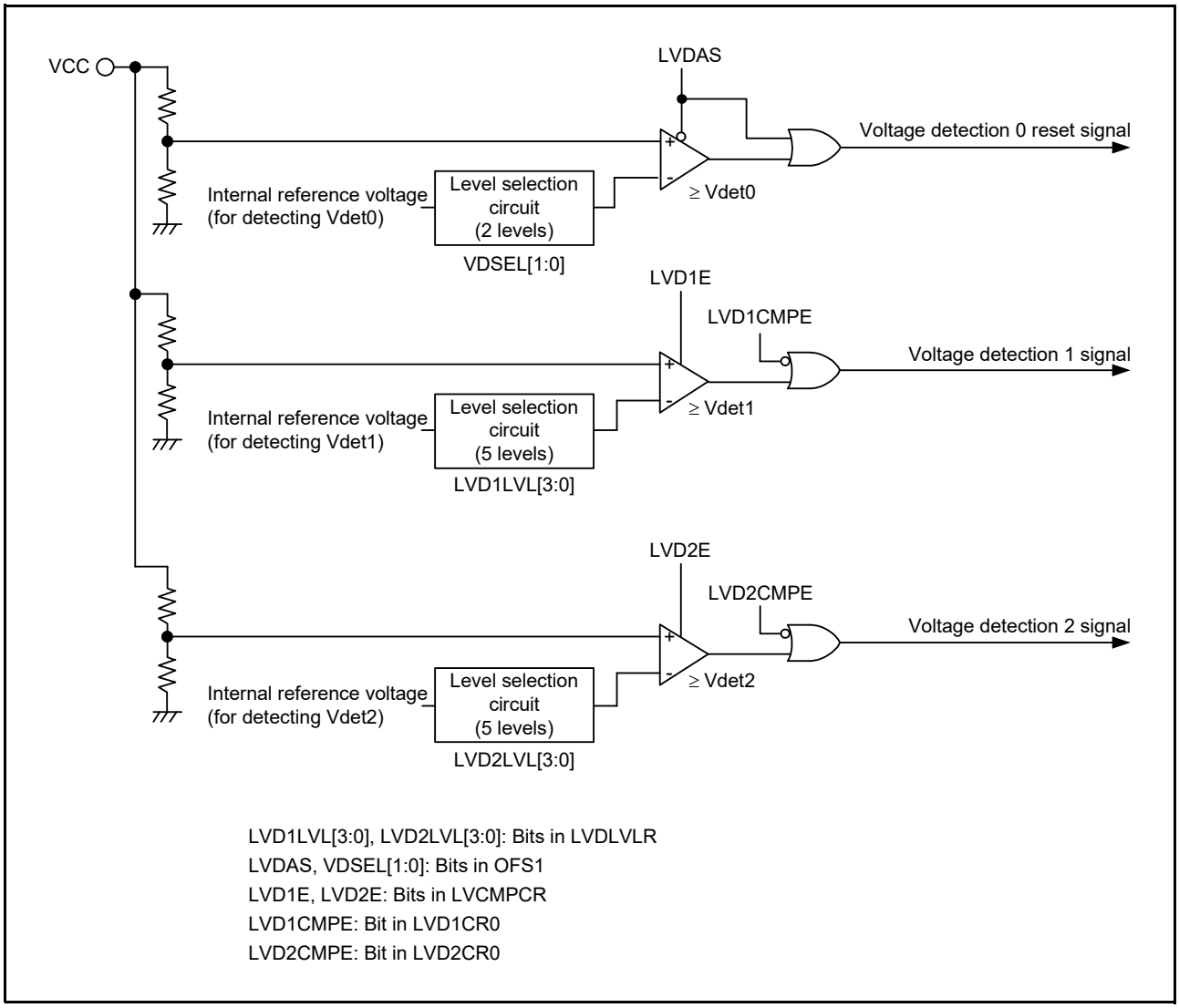


Figure 8.1 Block Diagram of Voltage Detection Circuit

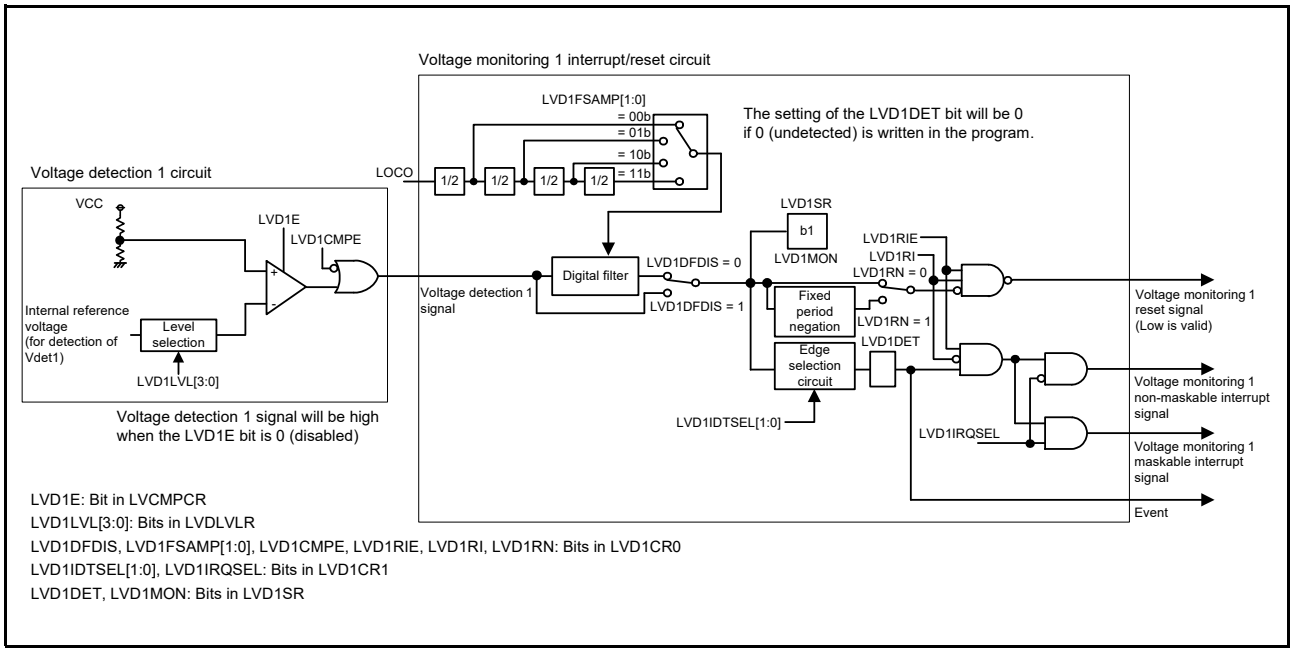


Figure 8.2 Block Diagram of Voltage Monitoring 1 Interrupt/Reset Circuit

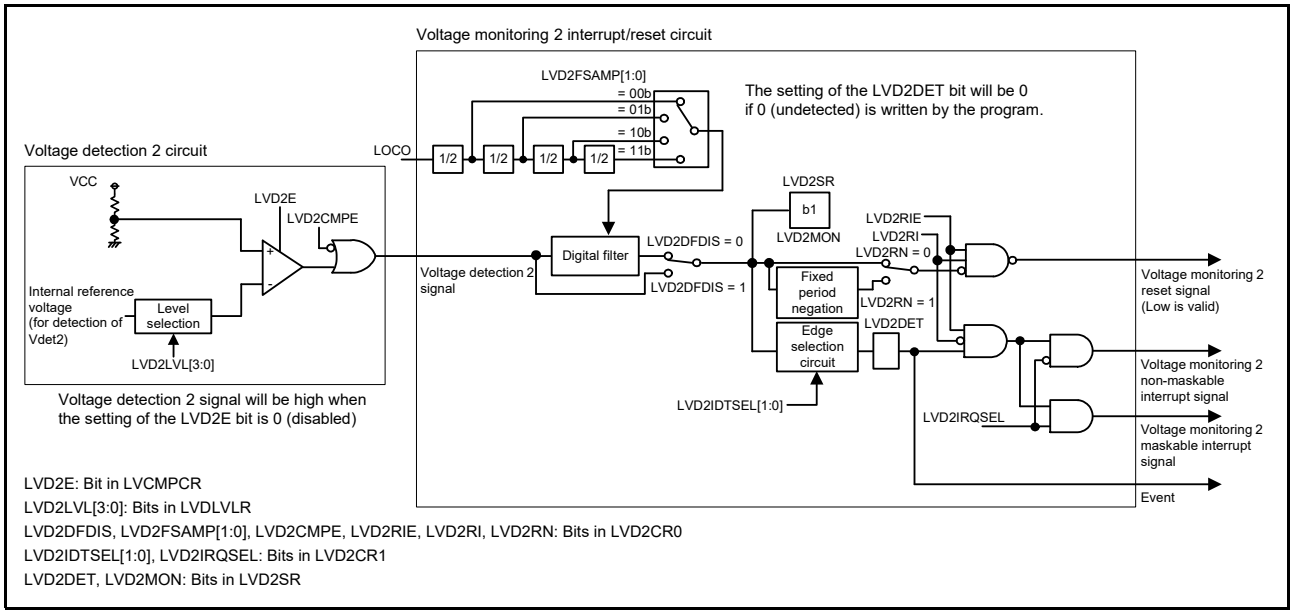


Figure 8.3 Block Diagram of Voltage Monitoring 2 Interrupt/Reset Circuit

8.2 Register Descriptions

8.2.1 Voltage Monitoring 1 Circuit Control Register 1 (LVD1CR1)

Address(es): 0008 00E0h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	LVD1IRQSEL	LVD1IDTSEL[1:0]	
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	LVD1IDTSEL[1:0]	Voltage Monitoring 1 Interrupt Generation Condition Select	b1 b0 0 0: When VCC ≥ Vdet1 (rise) is detected 0 1: When VCC < Vdet1 (drop) is detected 1 0: When drop and rise are detected 1 1: Settings prohibited	R/W
b2	LVD1IRQSEL	Voltage Monitoring 1 Interrupt Type Select	0: Non-maskable interrupt 1: Maskable interrupt*1	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When enabling maskable interrupts, do not change the value of the NMIER.LVD1EN bit on the ICU side from the reset state.

8.2.2 Voltage Monitoring 1 Circuit Status Register (LVD1SR)

Address(es): 0008 00E1h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	LVD1MON	LVD1DET
Value after reset:	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	LVD1DET	Voltage Monitoring 1 Voltage Change Detection Flag	0: Not detected 1: Vdet1 passage detection	R(/W) *1
b1	LVD1MON	Voltage Monitoring 1 Signal Monitor Flag	0: VCC < Vdet1 1: VCC ≥ Vdet1 or LVD1MON is disabled	R
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only 0 can be written to this bit. After writing 0 to this bit, it takes 2 system clock cycles for the bit to be read as 0.

LVD1DET Flag (Voltage Monitoring 1 Voltage Change Detection Flag)

The LVD1DET flag is enabled when the LVCMPCR.LVD1E bit is 1 (voltage detection 1 circuit enabled) and the LVD1CR0.LVD1CMPE bit is 1 (voltage monitoring 1 circuit comparison result output enabled).
The LVD1DET flag should be set to 0 after LVD1CR0.LVD1RIE is set to 0 (disabled). LVD1CR0.LVD1RIE can be set to 1 (enabled) after a period of two or more cycles of PCLKB has elapsed.
Depending on the number of cycles of PCLKB defined for access to read an I/O register, two or more cycles than PCLKB may have to be secured as waiting time.

LVD1MON Flag (Voltage Monitoring 1 Signal Monitor Flag)

The LVD1MON flag is enabled when the LVCMPCR.LVD1E bit is 1 (voltage detection 1 circuit enabled) and the LVD1CR0.LVD1CMPE bit is 1 (voltage monitoring 1 circuit comparison result output enabled).

8.2.3 Voltage Monitoring 2 Circuit Control Register 1 (LVD2CR1)

Address(es): 0008 00E2h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	LVD2IRQSEL	LVD2IDTSEL [1:0]	
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	LVD2IDTSEL [1:0]	Voltage Monitoring 2 Interrupt Generation Condition Select	b1 b0 0 0: When VCC ≥ Vdet2 (rise) is detected 0 1: When VCC < Vdet2 (drop) is detected 1 0: When drop and rise are detected 1 1: Settings prohibited	R/W
b2	LVD2IRQSEL	Voltage Monitoring 2 Interrupt Type Select	0: Non-maskable interrupt 1: Maskable interrupt*1	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When enabling maskable interrupts, do not change the value of the NMIER.LVD2EN bit on the ICU side from the reset state.

8.2.4 Voltage Monitoring 2 Circuit Status Register (LVD2SR)

Address(es): 0008 00E3h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	LVD2MON	LVD2DET
Value after reset:	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	LVD2DET	Voltage Monitoring 2 Voltage Change Detection Flag	0: Not detected 1: Vdet2 passage detection	R(/W) *1
b1	LVD2MON	Voltage Monitoring 2 Signal Monitor Flag	0: VCC < Vdet2 1: VCC ≥ Vdet2 or LVD2MON is disabled	R
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only 0 can be written to this bit. After writing 0 to this bit, it takes 2 system clock cycles for the bit to be read as 0.

LVD2DET Flag (Voltage Monitoring 2 Voltage Change Detection Flag)

The LVD2DET flag is enabled when the LVCMPCR.LVD2E bit is 1 (voltage detection 2 circuit enabled) and the LVD2CR0.LVD2CMPE bit is 1 (voltage monitoring 2 circuit comparison result output enabled).
The LVD2DET flag should be set to 0 after LVD2CR0.LVD2RIE bit is set to 0 (disabled). LVD2CR0.LVD2RIE bit can be set to 1 (enabled) after a period of two or more cycles of PCLKB has elapsed.
Depending on the number of cycles of PCLKB defined for access to read an I/O register, two or more cycles of PCLKB may have to be secured as waiting time.

LVD2MON Flag (Voltage Monitoring 2 Signal Monitor Flag)

The LVD2MON flag is enabled when the LVCMPCR.LVD2E bit is 1 (voltage detection 2 circuit enabled) and the LVD2CR0.LVD2CMPE bit is 1 (voltage monitoring 2 circuit comparison result output enabled).

8.2.5 Voltage Monitoring Circuit Control Register (LVCMPCR)

Address(es): 0008 C297h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	LVD2E	LVD1E	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	LVD1E	Voltage Detection 1 Enable	0: Voltage detection 1 circuit disabled 1: Voltage detection 1 circuit enabled	R/W
b6	LVD2E	Voltage Detection 2 Enable	0: Voltage detection 2 circuit disabled 1: Voltage detection 2 circuit enabled	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

LVD1E Bit (Voltage Detection 1 Enable)

When using voltage detection 1 interrupt/reset or the LVD1SR.LVD1MON bit, set the LVD1E bit to 1. The voltage detection 1 circuit starts once td(E-A) passes after the LVD1E bit value is changed from 0 to 1.

LVD2E Bit (Voltage Detection 2 Enable)

When using voltage detection 2 interrupt/reset or the LVD2SR.LVD2MON bit, set the LVD2E bit to 1. The voltage detection 2 circuit starts once td(E-A) passes after the LVD2E bit value is changed from 0 to 1.

8.2.6 Voltage Detection Level Select Register (LVDLVLR)

Address(es): 0008 C298h

	b7	b6	b5	b4	b3	b2	b1	b0
	LVD2LVL[3:0]				LVD1LVL[3:0]			
Value after reset:	0	1	1	0	0	1	1	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	LVD1LVL[3:0]	Voltage Detection 1 Level Select (Standard voltage during drop in voltage)	b3 b0 0 1 0 0: 4.57 V (Vdet1_0) 0 1 0 1: 4.47 V (Vdet1_1) 0 1 1 0: 4.32 V (Vdet1_2) 1 0 1 0: 2.93 V (Vdet1_3) 1 0 1 1: 2.88 V (Vdet1_4) Settings other than above are prohibited.	R/W
b7 to b4	LVD2LVL[3:0]	Voltage Detection 2 Level Select (Standard voltage during drop in voltage)	b7 b4 0 1 0 0: 4.57 V (Vdet2_0) 0 1 0 1: 4.47 V (Vdet2_1) 0 1 1 0: 4.32 V (Vdet2_2) 1 0 1 0: 2.93 V (Vdet2_3) 1 0 1 1: 2.88 V (Vdet2_4) Settings other than above are prohibited.	R/W

The contents of the LVDLVLR register can only be changed if the LVCMPCR.LVD1E and LVCMPCR.LVD2E bits (voltage detection n circuit disable; n = 1, 2) are both 0. The voltage detection circuits 1 and 2 should not be set at the same voltage detection level.

8.2.7 Voltage Monitoring 1 Circuit Control Register 0 (LVD1CR0)

Address(es): 0008 C29Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	LVD1RN	LVD1RI	LVD1FSAMP[1:0]		—	LVD1CMPE	LVD1DFDIS	LVD1RIE
Value after reset:	1	0	0	0	x	0	1	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	LVD1RIE	Voltage Monitoring 1 Interrupt/Reset Enable	0: Disabled 1: Enabled	R/W
b1	LVD1DFDIS	Voltage Monitoring 1 Digital Filter Disable Mode Select	0: Digital filter enabled 1: Digital filter disabled	R/W
b2	LVD1CMPE	Voltage Monitoring 1 Circuit Comparison Result Output Enable	0: Voltage monitoring 1 circuit comparison result output disabled. 1: Voltage monitoring 1 circuit comparison result output enabled.	R/W
b3	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b5, b4	LVD1FSAMP[1:0]	Sampling Clock Select	b5 b4 0 0: 1/2 LOCO frequency 0 1: 1/4 LOCO frequency 1 0: 1/8 LOCO frequency 1 1: 1/16 LOCO frequency	R/W
b6	LVD1RI	Voltage Monitoring 1 Circuit Mode Select	0: Voltage monitoring 1 interrupt during Vdet1 passage 1: Voltage monitoring 1 reset enabled when the voltage falls to and below Vdet1	R/W
b7	LVD1RN	Voltage Monitoring 1 Reset Negate Select	0: Negation follows a stabilization time (tLVD1) after VCC > Vdet1 is detected. 1: Negation follows a stabilization time (tLVD1) after assertion of the LVD1 reset.	R/W

LVD1RIE Bit (Voltage Monitoring 1 Interrupt/Reset Enable)

Ensure that neither a voltage monitoring 1 reset nor a voltage monitoring 1 interrupt is generated during programming or erasure of the flash memory.

LVD1DFDIS Bit (Voltage Monitoring 1 Digital Filter Disable Mode Select)

Set the LOCOCR.LCSTP bit to 0 (the LOCO operates) if the LVD1DFDIS bit is 0 (enabling the digital filter circuit). Set the LVD1DFDIS bit to 1 (digital filter circuit disabled) when using voltage monitoring 1 circuit in software standby mode.

LVD1FSAMP [1:0] Bits (Sampling Clock Select)

The LVD1FSAMP[1:0] bits can be modified only when the LVD1DFDIS bit is 1 (digital filter circuit disabled). The LVD1FSAMP[1:0] bits should not be modified when the LVD1DFDIS bit is 0 (digital filter circuit enabled).

LVD1RN Bit (Voltage Monitoring 1 Reset Negate Select)

If the LVD1RN bit is to be set to 1 (negation follows a stabilization time after assertion of the LVD1 reset signal), set the LOCOCR.LCSTP bit to 0 (the LOCO operates). Furthermore, if a transition to software standby is to be made, the only possible value for the LVD1RN bit is 0 (negation follows a stabilization time after VCC > Vdet1 is detected). Do not set the LVD1RN bit to 1 (negation follows a stabilization time after assertion of the LVD1 reset signal) when this is the case.

8.2.8 Voltage Monitoring 2 Circuit Control Register 0 (LVD2CR0)

Address(es): 0008 C29Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	LVD2RN	LVD2RI	LVD2FSAMP[1:0]		—	LVD2CMPE	LVD2DFDIS	LVD2RIE
Value after reset:	1	0	0	0	x	0	1	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	LVD2RIE	Voltage Monitoring 2 Interrupt/Reset Enable	0: Disabled 1: Enabled	R/W
b1	LVD2DFDIS	Voltage Monitoring 2 Digital Filter Disable Mode Select	0: Digital filter enabled 1: Digital filter disabled	R/W
b2	LVD2CMPE	Voltage Monitoring 2 Circuit Comparison Result Output Enable	0: Voltage monitoring 2 circuit comparison result output disabled. 1: Voltage monitoring 2 circuit comparison result output enabled.	R/W
b3	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b5, b4	LVD2FSAMP[1:0]	Sampling Clock Select	b5 b4 0 0: 1/2 LOCO frequency 0 1: 1/4 LOCO frequency 1 0: 1/8 LOCO frequency 1 1: 1/16 LOCO frequency	R/W
b6	LVD2RI	Voltage Monitoring 2 Circuit Mode Select	0: Voltage monitoring 2 interrupt during Vdet2 passage 1: Voltage monitoring 2 reset enabled when the voltage falls to and below Vdet2	R/W
b7	LVD2RN	Voltage Monitoring 2 Reset Negate Select	0: Negation follows a stabilization time (tLVD2) after VCC > Vdet2 is detected. 1: Negation follows a stabilization time (tLVD2) after assertion of the LVD2 reset.	R/W

LVD2RIE Bit (Voltage Monitoring 2 Interrupt/Reset Enable)

Ensure that neither a voltage monitoring 2 reset nor a voltage monitoring 2 interrupt is generated during programming or erasure of the flash memory.

LVD2DFDIS Bit (Voltage Monitoring 2 Digital Filter Disable Mode Select)

Set the LOCOCR.LCSTP bit to 0 (the LOCO operates) if the LVD1DFDIS bit is 0 (enabling the digital filter circuit). Set the LVD2DFDIS bit to 1 (digital filter circuit disabled) when using voltage monitoring 2 circuit in software standby mode.

LVD2FSAMP[1:0] Bits (Sampling Clock Select)

The LVD2FSAMP[1:0] bits can be modified only when the LVD2DFDIS bit is 1 (digital filter circuit disabled). The LVD2FSAMP[1:0] bits should not be modified when the LVD2DFDIS bit is 0 (digital filter circuit enabled).

LVD2RN Bit (Voltage Monitoring 2 Reset Negate Select)

If the LVD2RN bit is to be set to 1 (negation follows a stabilization time after assertion of the LVD2 reset signal), set the LOCOCR.LCSTP bit to 0 (the LOCO operates). Furthermore, if a transition to software standby is to be made, the only possible value for the LVD2RN bit is 0 (negation follows a stabilization time after VCC > Vdet2 is detected). Do not set the LVD2RN bit to 1 (negation follows a stabilization time after assertion of the LVD2 reset signal) when this is the case.

8.3 VCC Input Voltage Monitor

8.3.1 Monitoring Vdet0

Monitoring Vdet0 is not possible.

8.3.2 Monitoring Vdet1

Table 8.2 lists the procedures for setting up monitoring against Vdet1. After the settings are completed, results of comparison by voltage monitoring 1 can be monitored by using the LVD1SR.LVD1MON flag.

Table 8.2 Procedures for Setting up Monitoring against Vdet1

Step		Monitoring the Results of Comparison by Voltage Monitoring 1
Setting the voltage detection 1 circuit	1	Select the detection voltage by setting the LVDLVLR.LVD1LVL[3:0] bits.
	2	Set LVCMPCR.LVD1E = 1 (enabling the voltage detection 1 circuit).
	3	Wait for at least td(E-A) (LVD operation stabilization time after LVD is enabled).*1
Setting the digital filter *2	4	Select the sampling clock for the digital filter by setting the LVD1CR0.LVD1FSAMP[1:0] bits.
	5	Set LVD1CR0.LVD1DFDIS = 0 (enabling the digital filter).
	6	Wait for at least 2n + 3 cycles of the LOCO (where n = 2, 4, 8, 16, and the sampling clock for the digital filter is the LOCO frequency-divided by n).
Enabling output	7	Set LVD1CR0.LVD1CMPE = 1 (enabling output of the results of comparison by voltage monitoring 1).

Note 1. Steps 4 to 6 can be performed during the waiting time of step 3. For details of td(E-A), see section 49, Electrical Characteristics.
Note 2. Steps 4 to 6 are not required if the digital filter is not in use.

8.3.3 Monitoring Vdet2

Table 8.3 lists the procedures for setting up monitoring against Vdet2. After the settings are completed, results of comparison by voltage monitoring 2 can be monitored by using the LVD2SR.LVD2MON flag.

Table 8.3 Procedures for Setting up Monitoring against Vdet2

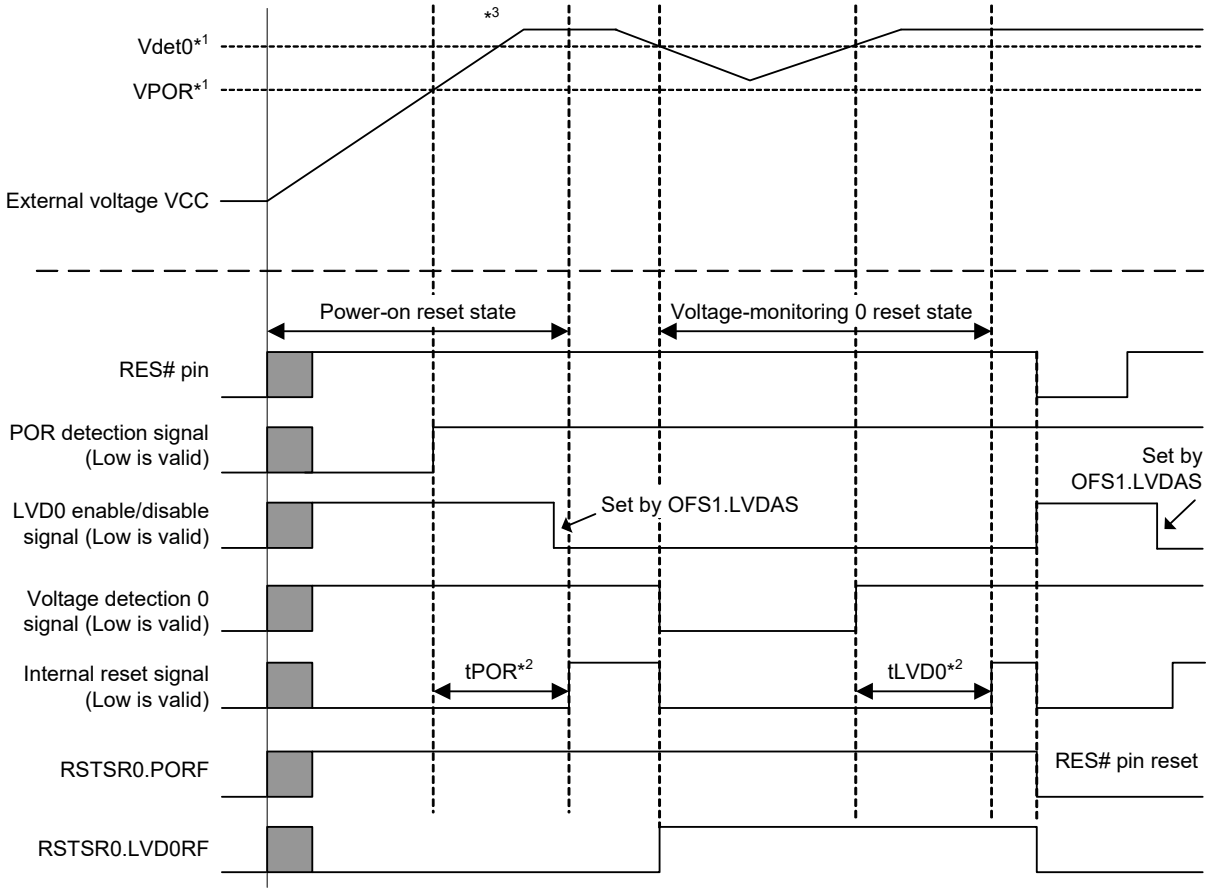
Step		Monitoring the Results of Comparison by Voltage Monitoring 2
Setting the voltage detection 2 circuit	1	Select the detection voltage by setting the LVDLVLR.LVD2LVL[3:0] bits.
	2	Set LVCMPCR.LVD2E = 1 (enabling the voltage detection 2 circuit).
	3	Wait for at least td(E-A) (LVD operation stabilization time after LVD is enabled).*1
Setting the digital filter *2	4	Select the sampling clock for the digital filter by setting the LVD2CR0.LVD2FSAMP[1:0] bits.
	5	Set LVD2CR0.LVD2DFDIS = 0 (enabling the digital filter).
	6	Wait for at least 2n + 3 cycles of the LOCO (where n = 2, 4, 8, 16, and the sampling clock for the digital filter is the LOCO frequency-divided by n).
Enabling output	7	Set LVD2CR0.LVD2CMPE = 1 (enabling output of the results of comparison by voltage monitoring 2).

Note 1. Steps 4 to 6 can be performed during the waiting time of step 3. For details of td(E-A), see section 49, Electrical Characteristics.
Note 2. Steps 4 to 6 are not required if the digital filter is not in use.

8.4 Reset from Voltage Monitor 0

When using the reset from voltage monitor 0, clear the OFS1.LVDAS bit to 0 (enabling the voltage monitor 0 reset after a reset).

Figure 8.4 shows an example of operations for a voltage monitoring 0 reset.



Note: For details of the electrical characteristics, see section 49, Electrical Characteristics.
Note 1. VPOR indicates the detection level for a power-on reset and Vdet0 indicates the detection level for a voltage monitor 0 reset.
Note 2. tPOR indicates the period of a power-on reset and tLVD0 indicates the period of a voltage monitor 0 reset.
Note 3. At the time the power-supply voltage rises, VCC must rise to at least the minimum guaranteed voltage before release from the POR reset state.

Figure 8.4 Example of Voltage Monitoring 0 Reset Operation

8.5 Interrupt and Reset from Voltage Monitor 1

An interrupt or reset can be generated in response to the results of comparison by the voltage detection 1 circuit. Table 8.4 lists the procedures for setting bits related to the voltage monitor 1 interrupt and voltage monitor 1 reset so that voltage monitoring operates. Table 8.5 shows the procedure for setting bits related to the voltage monitor 1 interrupt and voltage monitor 1 reset so that voltage monitoring stops. Figure 8.5 shows an example of operations for a voltage monitor 1 interrupt. For the operation of the voltage monitor 1 reset, see Figure 6.2 in section 6, Resets. Furthermore, if you intend to use the voltage monitoring 1 circuit in software standby mode, make settings for the voltage monitoring 1 circuit according to the following procedures.

- Disable the digital filter (LVD1DFDIS = 1).
- After VCC > Vdet1 is detected, negate the voltage monitoring 1 reset signal (LVD1RN = 0) following a stabilization time.

Table 8.4 Procedures for Setting Bits Related to the Voltage Monitoring 1 Interrupt and Voltage Monitoring 1 Reset so that Voltage Monitoring Operates

Step		Voltage Monitoring 1 Interrupt (Voltage Monitoring 1 ELC Event Output)	Voltage Monitoring 1 Reset
Setting the voltage detection 1 circuit	1	Select the detection voltage by setting the LVDLVLR.LVD1LVL[3:0] bits.	
	2	Set LVCMPCR.LVD1E = 1 (enabling the voltage detection 1 circuit).	
	3	Wait for at least td(E-A) (LVD operation stabilization time after LVD is enabled).*1	
Setting the digital filter *2	4	Select the sampling clock for the digital filter by setting the LVD1CR0.LVD1FSAMP[1:0] bits.	
	5	Set LVD1CR0.LVD1DFDIS = 0 (enabling the digital filter).	
	6	Wait for at least 2n + 3 cycles of the LOCO (where n = 2, 4, 8, 16, and the sampling clock for the digital filter is the LOCO frequency-divided by n).	
Setting the voltage monitoring 1 interrupt or reset	7	Set LVD1CR0.LVD1RI = 0 (selecting the voltage monitoring 1 interrupt).	• Set LVD1CR0.LVD1RI = 1 (selecting the voltage monitoring 1 reset). • Select the type of the reset negation by setting the LVD1CR0.LVD1RN bit.
	8	• Select the timing of interrupt requests by setting the LVD1CR1.LVD1IDTSEL[1:0] bits. • Select the type of interrupt by setting the LVD1CR1.LVD1IRQSEL bit.	—
Enabling output	9	Set LVD1SR.LVD1DET = 0.	
	10	Set LVD1CR0.LVD1RIE = 1 (enabling the voltage monitoring 1 interrupt or reset).*3	
	11	Set LVD1CR0.LVD1CMPE = 1 (enabling output of the results of comparison by voltage monitoring 1).	

Note 1. Steps 4 to 10 can be performed during the waiting time of step 3. For details of td(E-A), see section 49, Electrical Characteristics.
Note 2. Steps 4 to 6 are not required if the digital filter is not in use.
Note 3. Step 10 is not required if only the ELC event signal is to be output.

Table 8.5 Procedures for Setting Bits Related to the Voltage Monitoring 1 Interrupt and Voltage Monitoring 1 Reset so that Voltage Monitoring Stops

Step		Voltage Monitoring 1 Interrupt (Voltage Monitoring 1 ELC Event Output), Voltage Monitoring 1 Reset
Settings to stop enabling of output	1	Set LVD1CR0.LVD1CMPE = 0 (disabling output of the results of comparison by voltage monitoring 1).
	2	Wait for at least 2n + 3 cycles of the LOCO (where n = 2, 4, 8, 16, and the sampling clock for the digital filter is the LOCO frequency-divided by n).*1
	3	Set LVD1CR0.LVD1RIE = 0 (disabling the voltage monitoring 1 interrupt or reset).*2
Stopping the digital filter	4	Set LVD1CR0.LVD1DFDIS = 1 (disabling the digital filter).*1, *3
Stopping the voltage detection 1 circuit	5	Set LVCMPCR.LVD1E = 0 (disabling the voltage detection 1 circuit).

Note 1. Steps 2 and 4 are not required if the digital filter is not in use.
Note 2. Step 3 is not required if only the ELC event signal is to be output.
Note 3. To disable the digital filter from its enabled state and then re-enable it, disable it and wait for at least two cycles of the LOCO before re-enabling it.

If the voltage monitoring 1 interrupt or voltage monitoring 1 reset setting is to be made again after it has been used and stopped once, the following steps in the procedures for stopping and making the setting can be omitted according to the condition.

- Setting or stopping the voltage detection 1 circuit is not required if the setting for the voltage detection 1 circuit is not to be changed.
- Setting or stopping the digital filter is not required if the setting for the digital filter is not to be changed.
- Setting the voltage monitoring 1 interrupt or reset is not required if the setting for the voltage monitoring 1 interrupt or voltage monitoring 1 reset is not to be changed.

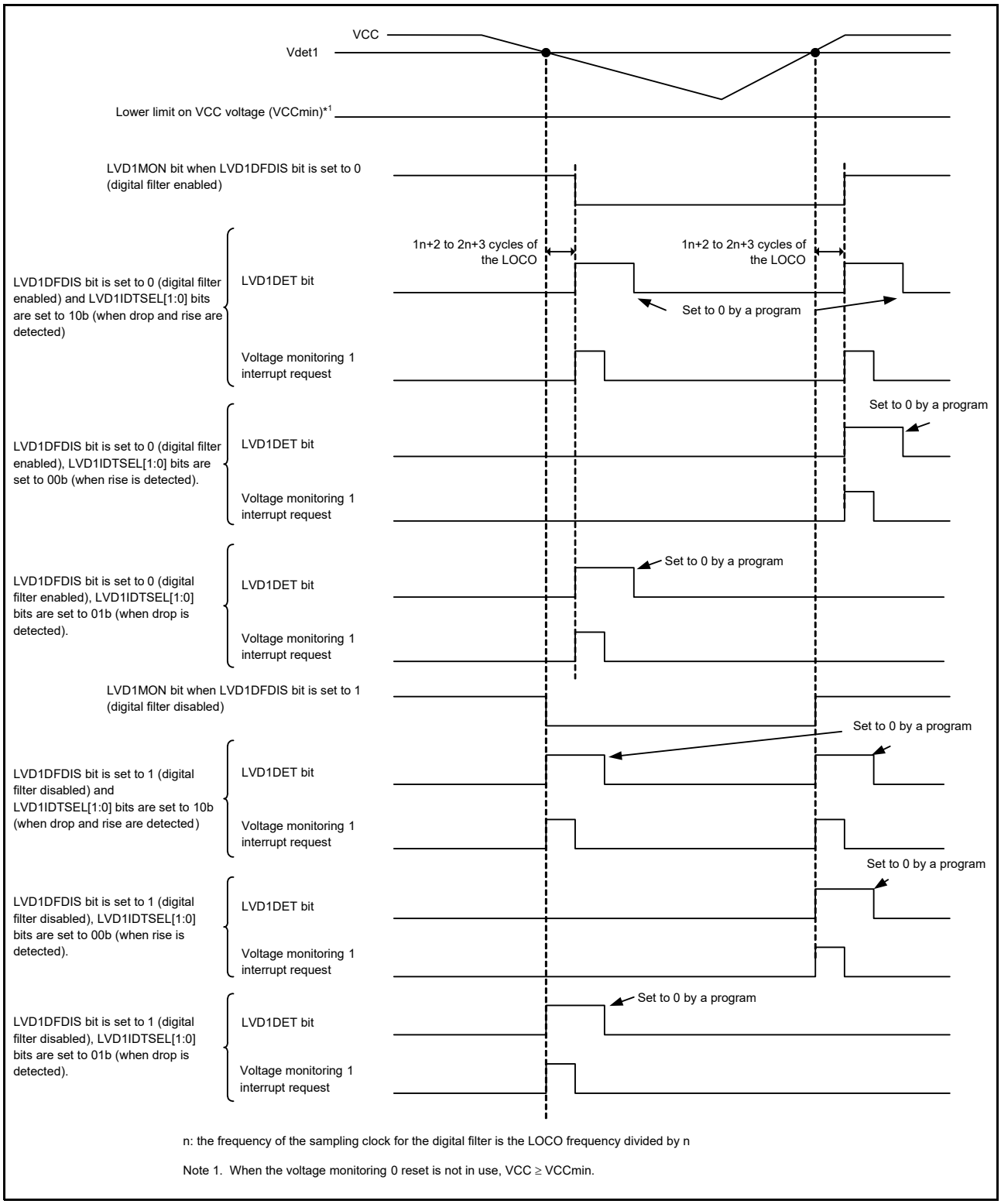


Figure 8.5 Example of Voltage Monitoring 1 Interrupt Operation

8.6 Interrupt and Reset from Voltage Monitor 2

An interrupt or reset can be generated in response to the results of comparison by the voltage detection 2 circuit. Table 8.6 shows the procedures for setting bits related to the voltage monitor 2 interrupt and voltage monitor 2 reset so that voltage monitoring operates.Table 8.7 shows the procedure for setting bits related to the voltage monitor 2 interrupt and voltage monitor 2 reset so that voltage monitoring stops. Figure 8.6 shows an example of operations for a voltage monitor 2 interrupt. For the operation of the voltage monitor 2 reset, see Figure 6.2 in section 6, Resets.
Furthermore, if you intend to use the voltage monitoring 2 circuit in software standby mode, make settings for the voltage monitoring 2 circuit according to the following procedures.

- Disable the digital filter (LVD2DFDIS = 1).
- After VCC > Vdet2 is detected, negate the voltage monitoring 2 reset signal (LVD2RN = 0) following a stabilization time.

Table 8.6 Procedures for Setting Bits Related to the Voltage Monitoring 2 Interrupt and Voltage Monitoring 2 Reset so that Voltage Monitoring Operates

Step		Voltage Monitoring 2 Interrupt (Voltage Monitoring 2 ELC Event Output)	Voltage Monitoring 2 Reset
Setting the voltage detection 2 circuit	1	Select the detection voltage by setting the LVDLVLR.LVD2LVL[3:0] bits.	
	2	Set LVCMPCR.LVD2E = 1 (enabling the voltage detection 2 circuit).	
	3	Wait for at least td(E-A) (LVD operation stabilization time after LVD is enabled).*1	
Setting the digital filter *2	4	Select the sampling clock for the digital filter by setting the LVD2CR0.LVD2FSAMP[1:0] bits.	
	5	Set LVD2CR0.LVD2DFDIS = 0 (enabling the digital filter).	
	6	Wait for at least 2n + 3 cycles of the LOCO (where n = 2, 4, 8, 16, and the sampling clock for the digital filter is the LOCO frequency-divided by n).	
Setting the voltage monitoring 2 interrupt or reset	7	Set LVD2CR0.LVD2RI = 0 (selecting the voltage monitoring 2 interrupt).	• Set LVD2CR0.LVD2RI = 1 (selecting the voltage monitoring 2 reset). • Select the type of the reset negation by setting the LVD2CR0.LVD2RN bit.
	8	• Select the timing of interrupt requests by setting the LVD2CR1.LVD2IDTSEL[1:0] bits. • Select the type of interrupt by setting the LVD2CR1.LVD2IRQSEL bit.	—
Enabling output	9	Set LVD2SR.LVD2DET = 0.	
	10	Set LVD2CR0.LVD2RIE = 1 (enabling the voltage monitoring 2 interrupt or reset).*3	
	11	Set LVD2CR0.LVD2CMPE = 1 (enabling output of the results of comparison by voltage monitoring 2).	

Note 1. Steps 4 to 10 can be performed during the waiting time of step 3. For details of td(E-A), see section 49, Electrical Characteristics.
Note 2. Steps 4 to 6 are not required if the digital filter is not in use.
Note 3. Step 10 is not required if only the ELC event signal is to be output.

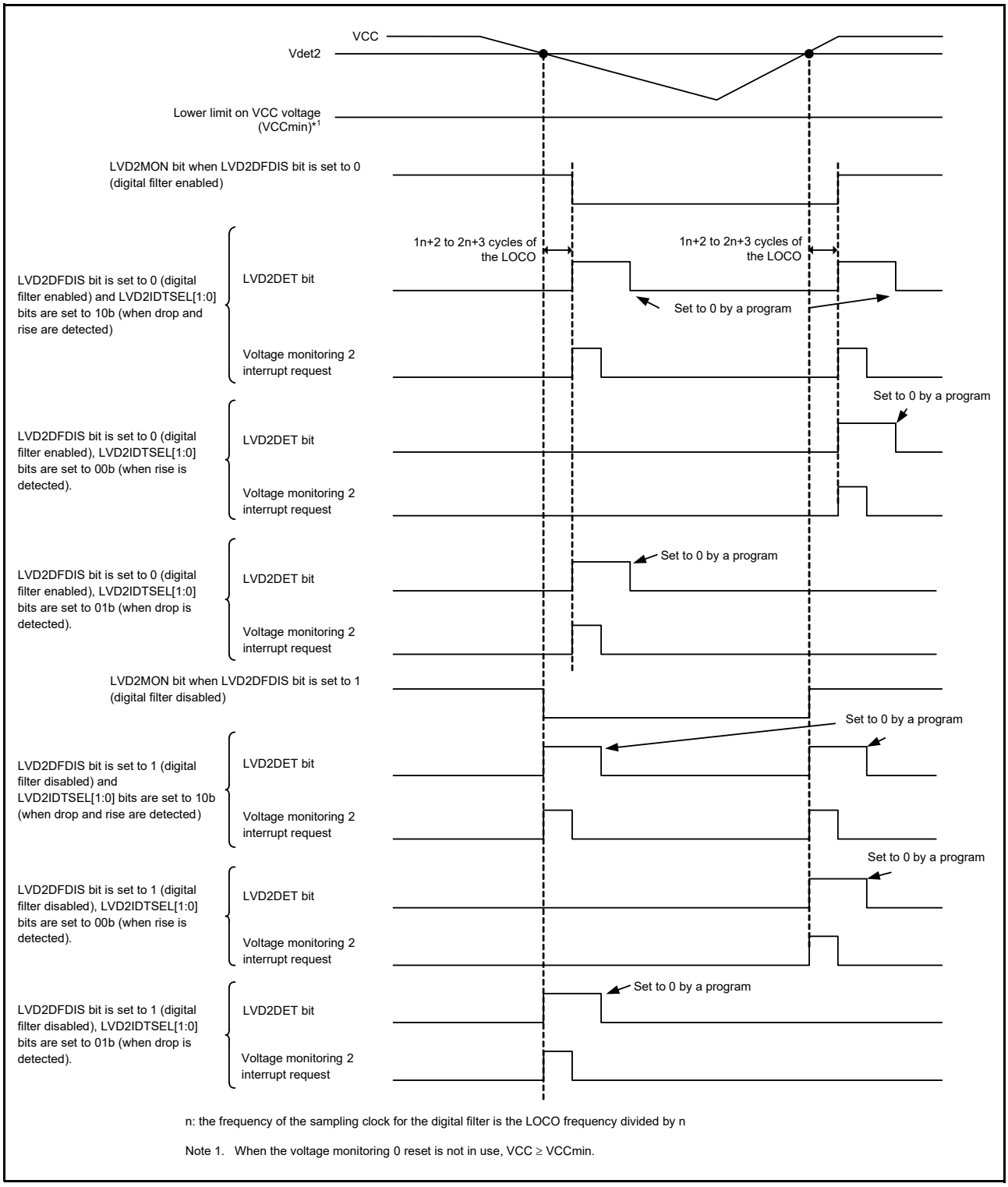
Table 8.7 Procedures for Setting Bits Related to the Voltage Monitoring 2 Interrupt and Voltage Monitoring 2 Reset so that Voltage Monitoring Stops

Step		Voltage Monitoring 2 Interrupt (Voltage Monitoring 2 ELC Event Output), Voltage Monitoring 2 Reset
Settings to stop enabling of output	1	Set LVD2CR0.LVD2CMPE = 0 (disabling output of the results of comparison by voltage monitoring 2).
	2	Wait for at least 2n + 3 cycles of the LOCO (where n = 2, 4, 8, 16, and the sampling clock for the digital filter is the LOCO frequency-divided by n).*1
	3	Set LVD2CR0.LVD2RIE = 0 (disabling the voltage monitoring 2 interrupt or reset).*2
Stopping the digital filter	4	Set LVD2CR0.LVD2DFDIS = 1 (disabling the digital filter).*1, *3
Stopping the voltage detection 1 circuit	5	Set LVCMPCR.LVD2E = 0 (disabling the voltage detection 2 circuit).

Note 1. Steps 2 and 4 are not required if the digital filter is not in use.
Note 2. Step 3 is not required if only the ELC event signal is to be output.
Note 3. To disable the digital filter from its enabled state and then re-enable it, disable it and wait for at least two cycles of the LOCO before re-enabling it.

If the voltage monitoring 2 interrupt or voltage monitoring 2 reset setting is to be made again after it has been used and stopped once, the following steps in the procedures for stopping and making the setting can be omitted according to the condition.

- Setting or stopping the voltage detection 2 circuit is not required if the setting for the voltage detection 2 circuit is not to be changed.
- Setting or stopping the digital filter is not required if the setting for the digital filter is not to be changed.
- Setting the voltage monitoring 2 interrupt or reset is not required if the setting for the voltage monitoring 2 interrupt or voltage monitoring 2 reset is not to be changed.



n: the frequency of the sampling clock for the digital filter is the LOCO frequency divided by n

Note 1. When the voltage monitoring 0 reset is not in use, VCC ≥ VCCmin.

Figure 8.6 Example of Voltage Monitoring 2 Interrupt Operation

8.7 Event Link Output

The LVD can output the event signals to the event link controller (ELC).

(1) Vdet1 passage detection event

The LVD outputs the event signal when it is detected that the voltage has passed the Vdet1 voltage while both the voltage detection 1 circuit and the voltage monitoring l circuit comparison result output are enabled.

(2) Vdet2 passage detection event

The LVD outputs the event signal when it is detected that the voltage has passed the Vdet2 voltage while both the voltage detection 2 circuit and the voltage monitoring 2 circuit comparison result output are enabled.

When enabling the LVD's event link output function, be sure to make settings for enabling the LVD before enabling the LVD event link function of the ELC. To stop the LVD's event link output function, be sure to make settings for stopping the LVD after disabling the LVD event link function of the ELC.

8.7.1 Interrupt Handling and Event Linking

The LVD has the bits to separately enable or disable the voltage monitoring l and 2 interrupts. When an interrupt source is generated and the interrupt is enabled by the interrupt enable bit, the interrupt signal (LVD1RIE and LVD2RIE) is output to the CPU.
On the contrary, as soon as an interrupt source is generated, the event link signal is output as the event signal to the other module via the ELC regardless of the state of the interrupt enable bit.
When the event Vdet1/Vdet2 are detected in software standby mode, no event signals are generated for the ELC because no clock is presented in software standby mode. Since Vdet1/Vdet2 passage detection flags are preserved, however, when the supply of the clock is resumed after restoring from software standby mode, the event signals for the ELC are output according to the state of the Vdet1/Vdet2 passage detection flags.

9. Clock Generation Circuit

9.1 Overview

This MCU incorporates a clock generation circuit.
Table 9.1 lists the specifications of the clock generation circuit. Figure 9.1 shows a block diagram of the clock generation circuit.

Table 9.1 Specifications of Clock Generation Circuit

Item	Specification
Use	• Generates the system clock (ICLK) to be supplied to the CPU, TFU, DMAC, DTC, code flash memory, and RAM. • Generates the peripheral module clock (PCLKA) to be supplied to the RSPI, RSPIA, RSCI, RI3C, CANFD, MTU (internal peripheral buses), GPTW (internal peripheral buses), and HRPWM (internal peripheral buses). • Generates the peripheral module clock (PCLKB) to be supplied to peripheral modules. • Generates the counter reference clock for the peripheral module to be supplied to the MTU and GPTW and the reference clock (PCLKC) for the HRPWM. • Generates the peripheral module clocks (for analog conversion) (PCLKD) to be supplied to S12AD. • Generates the flash-IF clock (FCLK) to be supplied to the flash interface. • Generates the CAC clock (CACCLK) to be supplied to the CAC. • Generates the CANFD clock (CANFDCLK) to be supplied to the CANFD. • Generates the CANFD main clock (CANFDMCLK) to be supplied to the CANFD. • Generates the IWDT-dedicated clock (IWDTCLK) to be supplied to the IWDT.
Operating frequency*1	• ICLK: 120 MHz (max) • PCLKA: 120 MHz (max) • PCLKB: 60 MHz (max) • PCLKC: 120 MHz (max) • PCLKD: 8 MHz to 60 MHz (for conversion with 12-bit A/D converter) • FCLK: 4 MHz to 60 MHz (for programming and erasing the code flash memory and data flash memory) 60 MHz (max) (for reading from the data flash memory) • CACCLK: Same as the clock from respective oscillators. • CANFDCLK: 60 MHz (max) • CANFDMCLK: 24 MHz (max) • IWDTCLK: 120 kHz
Main clock oscillator	• Resonator frequency: 8 MHz to 24 MHz • External clock input frequency: 24 MHz (max) • Connectable resonator or additional circuit: ceramic resonator, crystal resonator • Connection pin: EXTAL, XTAL • Oscillation stop detection function: When an oscillation stop is detected with the main clock, the system clock source is switched to LOCO, and MTU and GPTW output can be forcedly driven to the high-impedance.
PLL frequency synthesizer	• Input clock source: Main clock, HOCO*2 • Input pulse frequency division ratio: Selectable from 1, 2, and 3 • Input frequency: 8 MHz to 24 MHz • Frequency multiplication ratio: Selectable from 10 to 30 • Output clock frequency of the PLL frequency synthesizer: 120 MHz to 240 MHz
High-speed on-chip oscillator (HOCO)	• Selectable from 16 MHz, 18 MHz, and 20 MHz • HOCO power supply control
Low-speed on-chip oscillator (LOCO)	Oscillation frequency: 240 kHz
IWDT-dedicated on-chip oscillator	Oscillation frequency: 120 kHz
Event linking (output)	Detection of stopping of the main clock oscillator
Event linking (input)	Switching of the clock source to the low-speed on-chip oscillator

Note 1. Restrictions on setting clock frequency: PCLKC ≥ PCLKA ≥ PCLKB
Restrictions on clock frequency ratio: (N: integer)
ICLK:FCLK = N:1 or 1:N; ICLK:PCLKA = N:1 or 1:N; ICLK:PCLKB = N:1 or 1:N;
ICLK:PCLKC = N:1 or 1:N; ICLK:PCLKD = N:1 or 1:N;
PCLKA:PCLKC = 1:1 or 1:2, PCLKB:PCLKD = 1:1, 2:1, 4:1 or 1:2
Restrictions on clock-frequency settings when the CAN FD module is to be used: PCLKA:PCLKB = 2:1, PCLKB ≥ CANFDCLK, PCLKB ≥ CANFDMCLK

Note 2. When using the HOCO as the input clock source for the PLL, select the multiplication ratio of the PLL so that the HOCO clock oscillation frequency (min./max.) is in the range of 120 to 240 MHz.

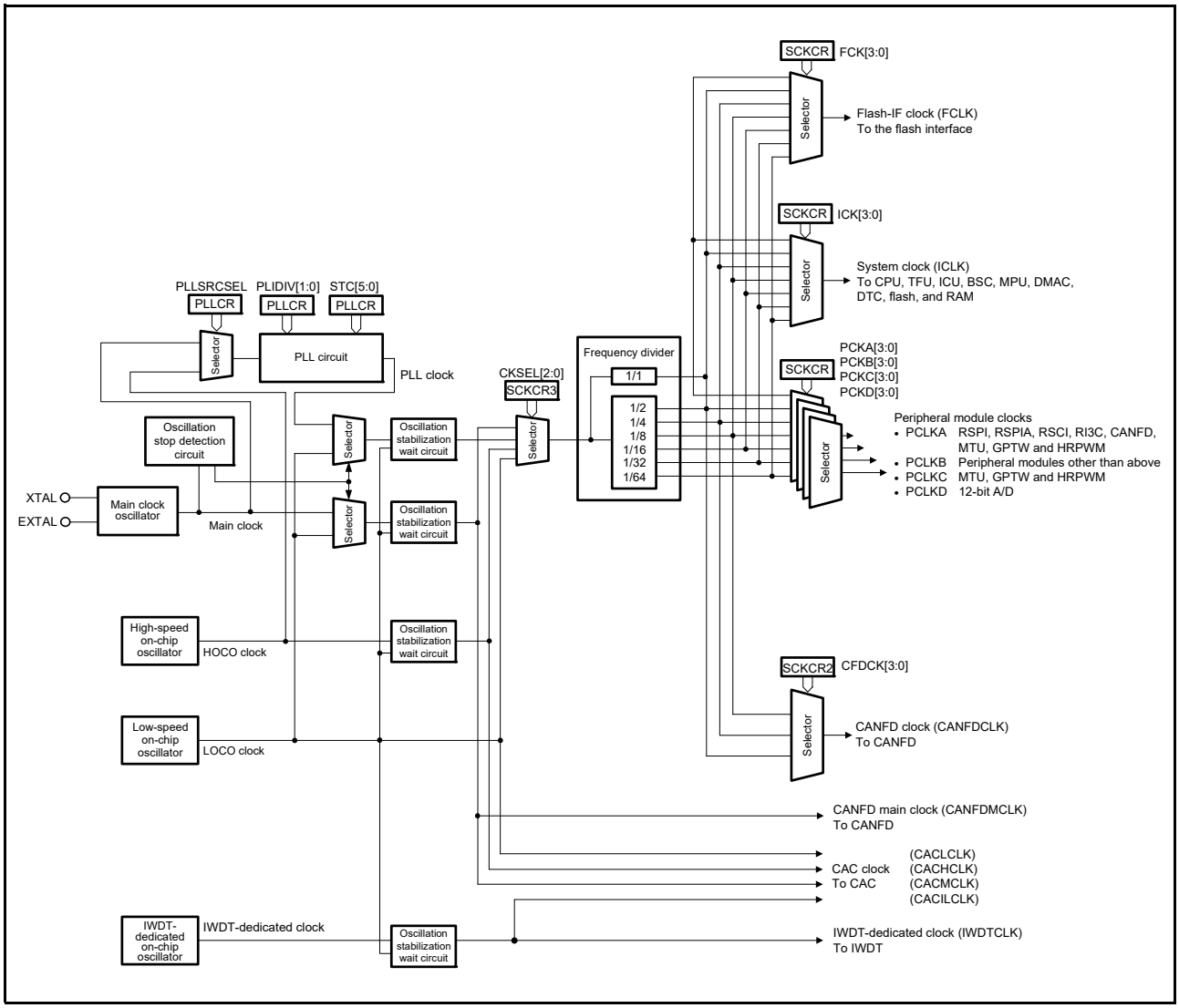


Figure 9.1 Block Diagram of Clock Generation Circuit

Table 9.2 lists the input/output pins of the clock generation circuit.

Table 9.2 Input/Output Pins of Clock Generation Circuit

Pin Name	I/O	Description
XTAL	Output	These pins are used to connect a crystal resonator. The EXTAL pin can also be used to input an external clock. For details, section 9.3.2, External Clock Input.
EXTAL	Input	

9.2 Register Descriptions

9.2.1 System Clock Control Register (SCKCR)

Address(es): 0008 0020h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	FCK[3:0]				ICK[3:0]				—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PCKA[3:0]				PCKB[3:0]				PCKC[3:0]				PCKD[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	PCKD[3:0]	Peripheral Module Clock D (PCLKD) Select	b3 b0 0 0 0 0: ×1/1 0 0 0 1: ×1/2 0 0 1 0: ×1/4 0 0 1 1: ×1/8 0 1 0 0: ×1/16 0 1 0 1: ×1/32 0 1 1 0: ×1/64 Settings other than above are prohibited.	R/W
b7 to b4	PCKC[3:0]	Peripheral Module Clock C (PCLKC) Select	b7 b4 0 0 0 0: ×1/1 0 0 0 1: ×1/2 0 0 1 0: ×1/4 0 0 1 1: ×1/8 0 1 0 0: ×1/16 0 1 0 1: ×1/32 0 1 1 0: ×1/64 Settings other than above are prohibited.	R/W
b11 to b8	PCKB[3:0]	Peripheral Module Clock B (PCLKB) Select	b11 b8 0 0 0 0: ×1/1 0 0 0 1: ×1/2 0 0 1 0: ×1/4 0 0 1 1: ×1/8 0 1 0 0: ×1/16 0 1 0 1: ×1/32 0 1 1 0: ×1/64 Settings other than above are prohibited.	R/W
b15 to b12	PCKA[3:0]	Peripheral Module Clock A (PCLKA) Select	b15 b12 0 0 0 0: ×1/1 0 0 0 1: ×1/2 0 0 1 0: ×1/4 0 0 1 1: ×1/8 0 1 0 0: ×1/16 0 1 0 1: ×1/32 0 1 1 0: ×1/64 Settings other than above are prohibited.	R/W
b19 to b16	—	Reserved	Set the same value as the ICK[3:0] bits.	R/W
b23 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit Name	Description	R/W
b27 to b24	ICK[3:0]	System Clock (ICLK) Select	b27 b24 0 0 0 0: ×1/1 0 0 0 1: ×1/2 0 0 1 0: ×1/4 0 0 1 1: ×1/8 0 1 0 0: ×1/16 0 1 0 1: ×1/32 0 1 1 0: ×1/64 Settings other than above are prohibited.	R/W
b31 to b28	FCK[3:0]	Flash-IF Clock (FCLK) Select	b31 b28 0 0 0 0: ×1/1 0 0 0 1: ×1/2 0 0 1 0: ×1/4 0 0 1 1: ×1/8 0 1 0 0: ×1/16 0 1 0 1: ×1/32 0 1 1 0: ×1/64 Settings other than above are prohibited.	R/W

SCKCR should not be modified in the following cases:

- The code flash memory P/E mode entry bit or the data flash memory P/E mode entry bit in the flash P/E mode entry register (FENTRYR.FENTRYC or FENTRYR.FENTRYD) is 1 (P/E mode)
- Time period from WAIT instruction execution for a transition to sleep mode, to return from sleep mode to normal operating mode

9.2.2 System Clock Control Register 2 (SCKCR2)

Address(es): 0008 0024h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CFDCK[3:0]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b11 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b12	CFDCK[3:0]	CANFD Clock (CANFDCLK) Select	b15 b12 0 0 0 1: ×1/2 0 0 1 0: ×1/4 0 0 1 1: ×1/8 Settings other than above are prohibited.	R/W

SCKCR2 should not be modified in the following cases:

- The code flash memory P/E mode entry bit or the data flash memory P/E mode entry bit in the flash P/E mode entry register (FENTRYR.FENTRYC or FENTRYR.FENTRYD) is 1 (P/E mode)
- Time period from WAIT instruction execution for a transition to sleep mode, to return from sleep mode to normal operating mode

9.2.3 System Clock Control Register 3 (SCKCR3)

Address(es): 0008 0026h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	CKSEL[2:0]			—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b10 to b8	CKSEL[2:0]	Clock Source Select	b10 b8 0 0 0: LOCO 0 0 1: HOCO 0 1 0: Main clock oscillator 1 0 0: PLL circuit Settings other than above are prohibited.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SCKCR3 should not be modified in the following cases:

- The code flash memory P/E mode entry bit or the data flash memory P/E mode entry bit in the flash P/E mode entry register (FENTRYR.FENTRYC or FENTRYR.FENTRYD) is 1 (P/E mode)
- Time period from WAIT instruction execution for a transition to sleep mode, to return from sleep mode to normal operating mode

CKSEL[2:0] Bits (Clock Source Select)

These bits select the source of the system clock (ICLK), peripheral module clocks (PCLKA, PCLKB, PCLKC, and PCLKD), flash-IF clock (FCLK), and CANFD clock (CANFDCLK) from low-speed on-chip oscillator (LOCO), high-speed on-chip oscillator (HOCO), the main clock oscillator, and the PLL circuit.
Transitions to clock sources which are not in operation are prohibited.

9.2.4 PLL Control Register (PLLCR)

Address(es): 0008 0028h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	STC[5:0]						—	—	—	PLLSRCSEL	—	—	PLIDIV[1:0]	
Value after reset:	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	PLIDIV[1:0]	PLL Input Frequency Division Ratio Select	b1 b0 0 0: ×1 0 1: ×1/2 1 0: ×1/3 1 1: Setting prohibited	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	PLLSRCSEL	PLL Clock Source Select	0: Main clock oscillator 1: HOCO	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13 to b8	STC[5:0]	Frequency Multiplication Factor Select	b13 b8 0 1 0 0 1 1: ×10.0 0 1 0 1 0 0: ×10.5 0 1 0 1 0 1: ×11.0 0 1 0 1 1 0: ×11.5 0 1 0 1 1 1: ×12.0 0 1 1 0 0 0: ×12.5 0 1 1 0 0 1: ×13.0 0 1 1 0 1 0: ×13.5 0 1 1 0 1 1: ×14.0 0 1 1 1 0 0: ×14.5 0 1 1 1 0 1: ×15.0 0 1 1 1 1 0: ×15.5 0 1 1 1 1 1: ×16.0 1 0 0 0 0 0: ×16.5 b13 b8 1 0 0 0 0 1: ×17.0 1 0 0 0 1 0: ×17.5 1 0 0 0 1 1: ×18.0 1 0 0 1 0 0: ×18.5 1 0 0 1 0 1: ×19.0 1 0 0 1 1 0: ×19.5 1 0 0 1 1 1: ×20.0 1 0 1 0 0 0: ×20.5 1 0 1 0 0 1: ×21.0 1 0 1 0 1 0: ×21.5 1 0 1 0 1 1: ×22.0 1 0 1 1 0 0: ×22.5 1 0 1 1 0 1: ×23.0 1 0 1 1 1 0: ×23.5 b13 b8 1 0 1 1 1 1: ×24.0 1 1 0 0 0 0: ×24.5 1 1 0 0 0 1: ×25.0 1 1 0 0 1 0: ×25.5 1 1 0 0 1 1: ×26.0 1 1 0 1 0 0: ×26.5 1 1 0 1 0 1: ×27.0 1 1 0 1 1 0: ×27.5 1 1 0 1 1 1: ×28.0 1 1 1 0 0 0: ×28.5 1 1 1 0 0 1: ×29.0 1 1 1 0 1 0: ×29.5 1 1 1 0 1 1: ×30.0 Settings other than above are prohibited.	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Writing to the PLLCR is prohibited when the PLLCR2.PLLEN bit is 0 (the PLL operates).

PLIDIV[1:0] Bits (PLL Input Frequency Division Ratio Select)

These bits select the frequency division ratio of the PLL clock source.
Set these bits so that the frequency of PLL input signal is within the range of 8 MHz to 24 MHz.

PLLSRCSEL Bit (PLL Clock Source Select)

This bit selects the clock source for the PLL.

STC[5:0] Bits (Frequency Multiplication Factor Select)

These bits select the frequency multiplication factor of the PLL circuit.
Set these bits so that the output frequency is within the range of the output clock frequency of the PLL circuit (120 MHz to 240 MHz).

9.2.5 PLL Control Register 2 (PLLCR2)

Address(es): 0008 002Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	PLLEN
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	PLLEN	PLL Stop Control	0: PLL is operating. 1: PLL is stopped.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

PLLEN Bit (PLL Stop Control)

This bit runs or stops the PLL circuit.

The PLL clock source is selectable as the main clock oscillator and HOCO.

Selecting the main clock oscillator as the PLL clock source with the PLLCR.PLLSRCSEL bit requires setting the main clock oscillator wait control register (MOSCWTCR).

After the setting of the PLLEN bit has been changed to make the PLL run, only start using the PLL clock after confirming that the OSCOVFSR.PLOVF flag has been set to 1.

That is, a fixed time for stabilization is required after the setting for PLL operation. A fixed time is also required for oscillation to stop after the setting to stop PLL operation. Accordingly, take note of the following limitations when starting and stopping PLL operation by the PLLEN bit. The following notes apply when selecting the main clock oscillator as the PLL clock source.

- Setting PLL operation with the PLLEN bit is possible regardless of the setting of the OSCOVFSR.PLOVF flag. However, the time until deactivation of the PLL is completed (the time until the PLOVF flag is set to 0 after the setting to stop PLL operation) means that writing to the PLLCR2 register takes longer than the setting to operate the PLL.
- The PLL can be stopped by the PLLEN bit regardless of the setting of the OSCOVFSR.PLOVF flag. However, the wait for oscillation to become stable, i.e. the oscillation settling time (the time until the OSCOVFSR.PLOVF flag is set to 1 after the setting to operate the PLL), means that writing to the PLLCR2 register takes longer than the setting to stop PLL operation.
- Regardless of whether or not the PLL clock is selected as the system clock, confirm that the OSCOVFSR.PLOVF flag has been set to 1 before executing a WAIT instruction to place the chip to software standby after the setting to operate the PLL.
- When a transition to software standby is to follow the setting to stop the PLL, confirm that the OSCOVFSR.PLOVF flag has been set to 0 before executing the WAIT instruction.

Writing of 1 to the PLLEN bit (stopping the PLL) is prohibited while the PLL clock is selected by the clock source select bits in system clock control register 3 (SCKCR3.CKSEL[2:0]).

9.2.6 Main Clock Oscillator Control Register (MOSCCR)

Address(es): 0008 0032h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	MOSTP
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	MOSTP	Main Clock Oscillator Stop	0: Main clock oscillator is operating. 1: Main clock oscillator is stopped.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

MOSTP Bit (Main Clock Oscillator Stop)

This bit runs or stops the main clock oscillator.

The main clock oscillator is operated or stopped by the MOSTP bit. The main clock oscillator can be started by setting the MOSTP bit to operating.

To make the main clock run, the main clock oscillator wait control register (MOSCWTCR) must be set. In this case, after the setting of the MOSCCR.MOSTP bit has been changed to make the main clock run, only start using the main clock after confirming that the OSCOVFSR.MOOVF flag has been set to 1.

For the main clock oscillator, a fixed time is required for oscillation to become stable after the settings for operation have been made. Furthermore, a fixed time is required for oscillation to actually stop after the settings to stop oscillation have been made. Accordingly, take note of the following limitations when starting and stopping operation.

- Main-clock operation can be selected with the MOSTP bit regardless of the setting of the OSCOVFSR.MOOVF flag. However, the time until deactivation of the main clock is completed (the time until the OSCOVFSR.MOOVF flag is set to 0 after the setting to stop operation) means that writing to the MOSCCR register takes longer than the setting to operate the main clock.
- The main clock can be stopped by the MOSTP bit regardless of the setting of the OSCOVFSR.MOOVF flag. However, the wait for oscillation to become stable, i.e. the oscillation settling time (the time until the OSCOVFSR.MOOVF flag is set to 1 after the setting to operate the main clock), means that writing to the MOSCCR register takes longer than the setting to stop operation.
- Regardless of whether or not it is selected as the system clock, confirm that the OSCOVFSR.MOOVF flag has been set to 1 before executing a WAIT instruction to place the chip to software standby after the setting to operate the main clock oscillator by the MOSTP bit.
- When a transition to software standby is to follow the setting to stop the main clock oscillator, confirm that the OSCOVFSR.MOOVF flag has been set to 0 before executing the WAIT instruction.

Writing of 1 to the MOSTP bit (stopping the main clock oscillator) is prohibited in either of the following cases:

- The main clock oscillator is selected by the clock source select bits in system clock control register 3 (SCKCR3.CKSEL[2:0]).
- 0 (PLL operation) is selected by the PLL stop control bit in PLL control register 2 (PLLCR2.PLLEN), and the main clock oscillator is selected by the PLL clock source select bit in the PLL control register (PLLCR.PLLSRCSEL).

9.2.7 Low-Speed On-Chip Oscillator Control Register (LOCOCR)

Address(es): 0008 0034h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	LCSTP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	LCSTP	LOCO Stop	0: LOCO is operating. 1: LOCO is stopped.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

LCSTP Bit (LOCO Stop)

This bit runs or stops the LOCO.

After the setting of the LCSTP bit has been changed so that the LOCO operates, only start using the LOCO after the LOCO clock oscillation stabilization time (tLOCOWT) has elapsed.

That is, a fixed time for stabilization of oscillation is required for oscillation to become stable after setting LOCO operation with the LCSTP bit. A fixed time is also required for oscillation to stop after the setting to stop the oscillator. Accordingly, take note of the following limitations when starting and stopping the oscillator.

- When restarting the LOCO after it has been stopped, allow at least five cycles of the LOCO as an interval over which it is still stopped.
- Ensure that oscillation by the LOCO is stable when making the setting to stop the LOCO.
- Regardless of whether or not it is selected as the system clock, ensure that oscillation by the LOCO is stable before executing a WAIT instruction to place the chip on software standby.
- When a transition to software standby is to follow the setting to stop the LOCO, wait for at least three cycles of the LOCO after the setting to stop the LOCO and before executing the WAIT instruction.

Writing of 1 to the LCSTP bit (stopping the LOCO) is prohibited while the LOCO is selected by the clock source select bits in system clock control register 3 (SCKCR3.CKSEL[2:0]).

Writing of 1 to the LCSTP bit (stopping the LOCO) is prohibited if detection of oscillation stopping is enabled by the oscillation stop detection enable bit in the oscillation stop detection control register (OSTDCR.OSTDE).

Since the LOCO clock is used to measure the waiting time for other oscillators, the LOCO clock oscillates while the waiting time for other oscillators is being measured, regardless of the setting of LCSTP bit. Therefore, the LOCO clock may be unintentionally supplied even if the LCSTP bit is set to be stopped.

9.2.8 IWDT-Dedicated On-Chip Oscillator Control Register (ILOCOCR)

Address(es): 0008 0035h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	ILCSTP
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	ILCSTP	IWDT-Dedicated On-Chip Oscillator Stop	0: IWDT-dedicated on-chip oscillator is operating. 1: IWDT-dedicated on-chip oscillator is stopped.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

When the IWDT start mode select bit in the option function select register 0 (OFS0.IWDTSTRT) is 0 (IWDT operating), the setting of this register is invalid; it is valid only when the OFS0.IWDTSTRT bit is set to 1 (IWDT stopped). The ILCSTP bit cannot be changed from 0 (IWDT-dedicated on-chip oscillator operating) to 1 (IWDT-dedicated on-chip oscillator stopped) while ILOCOCR is valid.

ILCSTP Bit (IWDT-Dedicated On-Chip Oscillator Stop)

This bit runs or stops the IWDT-dedicated on-chip oscillator.

After the setting of the ILCSTP bit has been changed to make the IWDT-dedicated on-chip oscillator run, confirm that the OSCOVFSR.ILCOVF flag has been set to 1 before starting to use the oscillator.

When a transition to software standby mode is to follow the setting to start the IWDT-dedicated on-chip oscillator, confirm that the OSCOVFSR.ILCOVF flag has been set to 1 before executing the WAIT instruction.

9.2.9 High-Speed On-Chip Oscillator Control Register (HOCOCR)

Address(es): 0008 0036h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	HCSTP
Value after reset:	0	0	0	0	0	0	0	0/1*1

Bit	Symbol	Bit Name	Description	R/W
b0	HCSTP	HOCO Stop	0: HOCO is operating. 1: HOCO is stopped.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. The HCSTP bit value after a reset is 0 when the OFS1.HOCOEN bit is 0. It is 1 when the OFS1.HOCOEN bit is 1.

HCSTP Bit (HOCO Stop)

This bit runs or stops the HOCO.

After the setting of the HCSTP bit has been changed to make the HOCO run, confirm that the OSCOVFSR.HCOVF flag has been set to 1 before starting to use the oscillator.

A fixed time for stabilization is required for oscillation to become stable after setting HOCO operation with the HCSTP bit. Furthermore, a fixed time is required for oscillation to actually stop after the setting to stop operation. Accordingly, take note of the following when starting and stopping operation with the HCSTP bit.

- Setting HOCO operation with the HCSTP bit is possible regardless of the setting of the OSCOVFSR.HCOVF flag. However, the time until deactivation of the HOCO is completed (the time until the OSCOVFSR.HCOVF flag is set to 0 after the setting to stop HOCO operation) means that writing to the HOCOCR register takes longer than the setting to operate the HOCO.
- The HOCO can be stopped by the HCSTP bit regardless of the setting of the OSCOVFSR.HCOVF flag. However, the wait for oscillation to become stable, i.e. the oscillation settling time (the time until the OSCOVFSR.HCOVF flag is set to 1 after the setting to operate the HOCO), means that writing to the HOCOCR register takes longer than the setting to stop HOCO operation.
- Regardless of whether or not it is selected as the system clock, confirm that the OSCOVFSR.HCOVF flag has been set to 1 before executing a WAIT instruction to place the chip on software standby after selecting HOCO operation with the HCSTP bit.
- When a transition to software standby is to follow the setting to stop the HOCO, confirm that the OSCOVFSR.HCOVF flag has been set to 0 after the setting to stop the HOCO and before executing the WAIT instruction.

Writing of 1 to the HCSTP bit (stopping the HOCO) is prohibited while the HOCO is selected by the clock source select bits in system clock control register 3 (SCKCR3.CKSEL[2:0]), or the HOCO is selected as the clock source for the PLL by the PLLCR.PLLSRCSEL bit, and the PLL is selected by the SCKCR3.CKSEL[2:0] bits.

9.2.10 High-Speed On-Chip Oscillator Control Register 2 (HOCOCR2)

Address(es): 0008 0037h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	HCFRQ[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	HCFRQ[1:0]	HOCO Frequency Setting	b1 b0 0 0: 16 MHz 0 1: 18 MHz 1 0: 20 MHz Settings other than above are prohibited.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Writing to the HOCOCR2 register is prohibited when the HOCOCR.HCSTP bit is 0 (making the HOCO run).

9.2.11 Oscillation Stabilization Flag Register (OSCOVFSR)

Address(es): 0008 003Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	ILCOVF	HCOVF	PLOVF	—	MOOVF
Value after reset:	0	0	0	0/1*1	0/1*2	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MOOVF	Main Clock Oscillation Stabilization Flag	0: MOSTP = 1 (stopping the main clock oscillator) or oscillation of the main clock has not yet become stable.*3 1: Oscillation of the main clock is stable so the clock is available for use as the system clock.	R
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	PLOVF	PLL Clock Oscillation Stabilization Flag	0: The PLL clock is stopped or oscillation of the PLL clock has not yet become stable. 1: Oscillation of the PLL clock is stable so the clock is available for use as the system clock.	R
b3	HCOVF*2	HOCO Clock Oscillation Stabilization Flag	0: The HOCO clock is stopped or oscillation of the HOCO clock has not yet become stable. 1: Oscillation of the HOCO clock is stable so the clock is available for use as the system clock.	R
b4	ILCOVF*1	IWDT-Dedicated Clock Oscillation Stabilization Flag	0: The IWDT-dedicated on-chip oscillator is stopped or oscillation of the IWDT-dedicated on-chip oscillator has not yet become stable. 1: Oscillation of the IWDT-dedicated on-chip oscillator is stable so the clock is available for use as the IWDT-dedicated clock.	R
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. The ILCOVF flag value after a reset is 1 when the OFS0.IWDTSTRT bit is 0. It is 0 when the OFS0.IWDTSTRT bit is 1.
Note 2. The HCOVF flag value after a reset is 1 when the OFS1.HOCOEN bit is 0. It is 0 when the OFS1.HOCOEN bit is 1.
Note 3. If the value set in the wait control register of the main clock oscillator is not sufficient for the given oscillation stabilization time, the oscillation stabilization flag is set to 1 and supply of the clock signal to the internal circuits starts before oscillation is stable. This may cause malfunction of this MCU, so ensure that the setting of the wait control register is at least the oscillation settling time for the oscillator considering the maximum frequency of the LOCO clock.

OSCOVFSR contains flags to indicate the states of operation of the counters within the oscillation stabilization wait circuits for the individual oscillators.
The counters measure the waiting times until each oscillator output clock is supplied to the internal circuits after oscillation starts, and an overflow of a counter indicates the start of clock supply from the corresponding oscillator to the internal circuits.

MOOVF Flag (Main Clock Oscillation Stabilization Flag)

This flag indicates the state of operation of the counter that measures the waiting time for the main clock oscillator.
[Setting condition]

- After the main clock oscillator has stopped and the MOSCCR.MOSTP bit is set to 0, the number of LOCO clock cycles corresponding to the setting of the MOSCWTCR register being counted and supply of the main clock within the MCU starting.

[Clearing condition]

- After the main clock oscillator has started to operate and the MOSCCR.MOSTP bit has been set to 1, deactivation of the main clock oscillator being completed.

PLOVF Flag (PLL Clock Oscillation Stabilization Flag)

This flag indicates the state of operation of the counter that measures the waiting time for the PLL.
[Setting condition]

- After the PLL has stopped and the PLLCR2.PLLEN bit is set to 0, 62 cycles of the LOCO clock being counted and supply of the PLL clock within the MCU starting.

If oscillation by the PLL clock source selected by the PLLCR.PLLSRCSEL bit is not stable when the PLLEN bit is set to 0, counting of LOCO clock cycles proceeds after the oscillation of the PLL clock source has been stabilized.

[Clearing condition]

- After the PLL has started to operate and the PLLCR2.PLLEN bit has been set to 1, deactivation of the PLL being completed.

HCOVF Flag (HOCO Clock Oscillation Stabilization Flag)

This flag indicates the state of operation of the counter that measures the waiting time for the high-speed on-chip oscillator.
[Setting condition]

- After the high-speed on-chip oscillator has stopped and the HOCOCR.HCSTP bit is set to 0, 25 cycles of the LOCO clock being counted and supply of the HOCO clock within the MCU starting.

[Clearing condition]

- After the high-speed on-chip oscillator has started to operate and the HOCOCR.HCSTP bit has been set to 1, deactivation of the high-speed on-chip oscillator being completed.

ILCOVF Flag (IWDT-Dedicated Clock Oscillation Stabilization Flag)

This flag indicates the state of operation of the counter that measures the waiting time for the IWDT-dedicated on-chip oscillator.
[Setting condition]

- After the IWDT-dedicated on-chip oscillator has stopped and the ILOCOCR.ILCSTP bit is set to 0, 34 cycles of the LOCO clock being counted and supply of the IWDT-dedicated clock within the MCU starting.

9.2.12 Oscillation Stop Detection Control Register (OSTDCR)

Address(es): 0008 0040h

	b7	b6	b5	b4	b3	b2	b1	b0
	OSTDE	—	—	—	—	—	—	OSTDIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OSTDIE	Oscillation Stop Detection Interrupt Enable	0: The oscillation stop detection interrupt is disabled. Oscillation stop detection is not notified to the POE/POEG. 1: The oscillation stop detection interrupt is enabled. Oscillation stop detection is notified to the POE/POEG.	R/W
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	OSTDE	Oscillation Stop Detection Function Enable	0: Oscillation stop detection function is disabled. 1: Oscillation stop detection function is enabled.	R/W

OSTDCR register is used to enable the oscillation stop detection function for the main clock oscillator and conveying of interrupts in response.

OSTDIE Bit (Oscillation Stop Detection Interrupt Enable)

If the oscillation-stop detection flag in the oscillation-stop detection status register (OSTDSR.OSTDF) requires clearing, do this after clearing the OSTDIE bit to 0. Wait for at least two cycles of PCLKB before again setting the OSTDIE bit to 1. According to the number of cycles for access to read out a given I/O register, waiting time longer than two cycles of PCLKB may have to be secured.

OSTDE Bit (Oscillation Stop Detection Function Enable)

This bit enables or disables the oscillation stop detection function.

When the OSTDE bit is 1 (oscillation stop detection function enabled), the LOCO stop bit (LOCOCR.LCSTP) is set to 0 and the LOCO operation is started. The LOCO cannot be stopped while the oscillation stop detection function is enabled; writing 1 to the LOCOCR.LCSTP bit (LOCO stopped) is invalid.

When the oscillation stop detection flag in the oscillation stop detection status register (OSTDSR.OSTDF) is 1 (main clock oscillation stop detected), writing 0 to the OSTDE bit is invalid.

When the OSTDE bit is 1, a transition cannot be made to software standby mode. To make a transition to software standby mode, execute the WAIT instruction with the OSTDE bit being 0.

To check the oscillation stop detection flag in the oscillation stop detection status register (OSTDSR.OSTDF), wait for at least three cycles of ICLK after the OSTDE bit has been set to 1 (oscillation stop detection enabled).

9.2.13 Oscillation Stop Detection Status Register (OSTDSR)

Address(es): 0008 0041h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	OSTDF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OSTDF	Oscillation Stop Detection Flag	0: Stopping of the main clock oscillator has not been detected. 1: Stopping of the main clock oscillator has been detected.	R(/W) *1
b7 to b1	—	Reserved	These bits are read as 0 and cannot be modified.	R

Note 1. This bit can only be set to 0.

The OSTDSR register indicates the detection of stopping of the main clock oscillator.

OSTDF Flag (Oscillation Stop Detection Flag)

This bit is a flag to indicate the main clock status. When the OSTDF flag is 1, it indicates that the main clock oscillation stop has been detected.

Once the main clock oscillation stop is detected, the OSTDF flag is not set to 0 even though the main clock oscillation is restarted. The OSTDF flag is set to 0 by reading 1 from the bit and then writing 0. At least 3 ICLK cycles of wait time is necessary between writing 0 to OSTDF and reading OSTDF as 0. If the OSTDF flag is set to 0 from 1 while the main clock oscillation is stopped, the OSTDF flag becomes 0 and then returns to 1.

The OSTDF flag cannot be modified to 0 while the main clock oscillator or PLL is selected by the clock source select bits in system clock control register 3 (SCKCR3.CKSEL[2:0]) (010b or 100b). The OSTDF flag should be set to 0 after switching the clock source to other sources than the main clock oscillator and PLL. When a resonator is selected as the source for the main clock oscillator, apply a reset to clear the OSTDF flag.

[Setting condition]

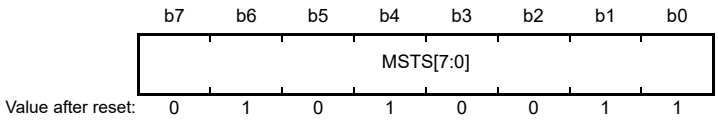
- The main clock oscillation is stopped with the OSTDCR.OSTDE being 1 (oscillation stop detection function enabled).

[Clearing condition]

- 1 is read and then 0 is written when the SCKCR3.CKSEL[2:0] bits are neither 010b nor 100b.

9.2.14 Main Clock Oscillator Wait Control Register (MOSCWTCR)

Address(es): 0008 00A2h



MOSCWTCR is used to control the waiting time until output of the signal from the main clock oscillator to the internal circuits starts. The oscillation stabilization wait circuit for the main clock oscillator measures the waiting time by counting the number of LOCO clock cycles corresponding to the setting of the MOSCWTCR register.
The oscillation stabilization wait circuit measures the waiting time and controls the clock supply within the MCU. When the main clock oscillator starts by setting the MOSCCR.MOSTP bit, the oscillation stabilization wait circuit starts counting the waiting time with the LOCO clock. The clock supply within the MCU is disabled over the period until counting of the set number of cycles is completed. After counting is completed, supply of the clock signal within the MCU starts and the OSCOVFSR.MOOVF flag is set to 1.
Counting of LOCO clock cycles by the oscillation stabilization wait circuit proceeds regardless of the setting of the LOCOCR.LCSTP bit. Hardware automatically controls running and stopping of the LOCO clock for measurement of the waiting time.
Values can only be written to MOSCWTCR while the MOSCCR.MOSTP bit is 1 or the OSCOVFSR.MOOVF flag is 1; do not attempt writing to MOSCWTCR if neither is the case.
The waiting time is not required when an external clock signal is input for the main clock oscillator. Set the MSTS[7:0] bits to 00h.

The value of the MSTS[7:0] bits required for correspondence with the waiting time required to secure stable oscillation by the main clock oscillator is obtained by using the maximum frequency for fLOCO in the formula below.

MSTS[7:0] > (tMAINOSC × (fLOCO_max) + 16)/32
(tMAINOSC: main clock oscillation stabilization time; fLOCO_max: maximum frequency for fLOCO)

If tMAINOSC is 1 ms and fLOCO_max is 264 kHz (the period is 1/3.78 µs), the formula gives MSTS[7:0] > (1 ms × (264 kHz) + 16)/32 = 8.75, so set the MSTS[7:0] bits to 9.

Waiting time:
When LOCO is at its highest frequency: (9 × 32 – 16) × (1/264 kHz = 3.78 µs) = 1.028 ms
When LOCO is at its normal frequency: (9 × 32 + 3) × (1/240 kHz = 4.18 µs) = 1.216 ms
When LOCO is at its lowest frequency: (9 × 32 + 10) × (1/216 kHz = 4.63 µs) = 1.380 ms

9.2.15 Main Clock Oscillator Function Control Register (MOFCR)

Address(es): 0008 C293h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	MOSEL	MODRV2[1:0]		—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5, b4	MODRV2[1:0]	Main Clock Oscillator Driving Ability 2 Switching	b5 b4 0 0: 20.1 to 24 MHz 0 1: 16.1 to 20 MHz 1 0: 8.1 to 16 MHz 1 1: 8 MHz	R/W
b6	MOSEL	Main Clock Oscillator Switching	0: Resonator 1: External clock input	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

MOFCR selects the driving ability of the main clock oscillator, and also selects the oscillator or an external clock signal.

MODRV2[1:0] Bits (Main Clock Oscillator Driving Ability 2 Switching)

These bits switch the driving ability of the main clock oscillator.
Specify the driving ability according to the frequency of a crystal connected to the main clock oscillator.
The frequency ranges specified in the bit description of the MODRV2[1:0] bits are the reference values of the crystal with capacitive load of 8 pF. A setting value may not fit within the frequency range depending on a crystal. Use values recommended by the resonator manufacturer.
In case of a ceramic resonator, it may be better to select lower frequency range than the frequency of the resonator (for example, specify 10b instead of 01b when a ceramic resonator with the frequency range from 16.1 to 20 MHz is used). Use values recommended by the resonator manufacturer.

MOSEL Bit (Main Clock Oscillator Switching)

This bit switches the source for the main clock oscillator.

9.2.16 High-Speed On-Chip Oscillator Power Supply Control Register (HOCOPCR)

Address(es): 0008 C294h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	HOCOPCNT
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	HOCOPCNT	High-Speed On-Chip Oscillator Power Supply Control	0: Turns the power supply of the HOCO on. 1: Turns the power supply of the HOCO off.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

HOCOPCNT Bit (High-Speed On-Chip Oscillator Power Supply Control)

This bit controls the power supply for the HOCO.

When this bit is set to 0, the power supply of the HOCO is turned on, enabling oscillation.

When this bit is set to 1, the power supply of the HOCO is turned off, reducing power consumption.

When setting the HOCOPCNT bit to 1, set the HOCO stop bit in the high-speed on-chip oscillator control register (HOCOCR.HCSTP) to 1 (HOCO stopped) beforehand.

After the HOCOPCNT bit is changed from 1 to 0, oscillation settling time is required before the HOCOCR.HCSTP bit is set to 0. For details, see section 49, Electrical Characteristics.

Do not change the value of the HOCOPCNT bit when the HOCO is selected as the clock source by the clock source select bits in system clock control register 3 (SCKCR3.CKSEL[2:0]).

9.3 Main Clock Oscillator

There are two ways of supplying the clock signal to the main clock oscillator: connecting an oscillator or the input of an external clock signal.

9.3.1 Connecting a Crystal Resonator

Figure 9.2 shows an example of connecting a crystal.
Connect capacitors referring to the capacitive load of the crystal to be used. In addition, a damping resistor Rd should be added, if necessary. The values of capacitors and resistor vary depending on the resonator and the oscillator driving ability. Use values recommended by the resonator manufacturer. If use of an external feedback resistor (Rf) is directed by the resonator manufacturer, insert an Rf between EXTAL and XTAL by following the instruction.
When connecting a resonator to supply the clock, the frequency of the crystal must be in the frequency range of the resonator for the main clock oscillator described in Table 9.1.
When a resonator is connected, setting the MOFCR.MODRV2[1:0] bits (Main Clock Oscillator Driving Ability 2 Switching) is required.
The frequency ranges that are specified in the bit description of the MODRV2[1:0] bits are the reference values of the crystal with capacitive load of C_L = 8 pF. The setting value may not fit within the frequency range depending on a crystal. Use values recommended by the resonator manufacturer.
In case of a ceramic resonator, it may be better to select lower frequency range than the frequency of the resonator. (For example, specify 10b instead of 01b when a ceramic resonator with the frequency range of 16.1 to 20 MHz is used). Use values recommended by the resonator manufacturer.

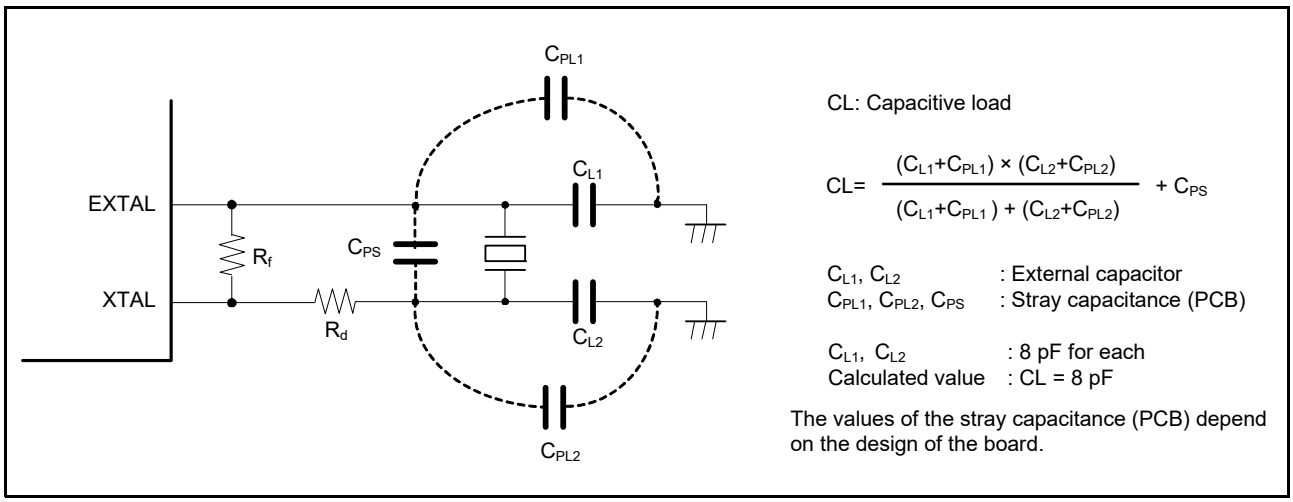


Figure 9.2 Example of Crystal Connection

Table 9.3 Damping Resistance (Reference Values)

Frequency (MHz)	8	12	16	20	24
Rd (Ω)	0	0	0	0	0

Figure 9.3 shows an equivalent circuit of the crystal resonator. Use a crystal resonator that has the characteristics shown in Table 9.4.

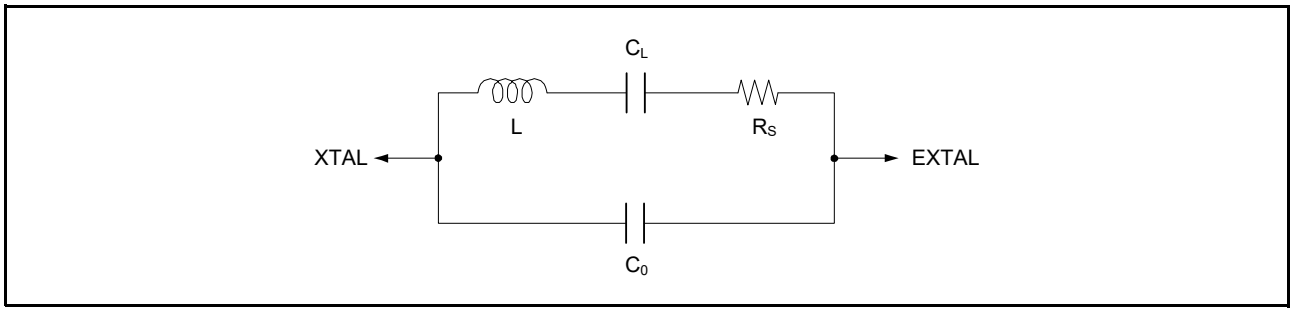


Figure 9.3 Equivalent Circuit of Crystal Resonator

Table 9.4 Crystal Characteristics (Reference Values)

Frequency (MHz)	8	12	16	20	24
R_S max (Ω)	300	100	80	50	50

9.3.2 External Clock Input

Figure 9.4 shows examples of connection of external clock input. Set the MOFCR.MOSEL bit to 1 and open the XTAL pin to operate the oscillator by inputting an external clock signal.

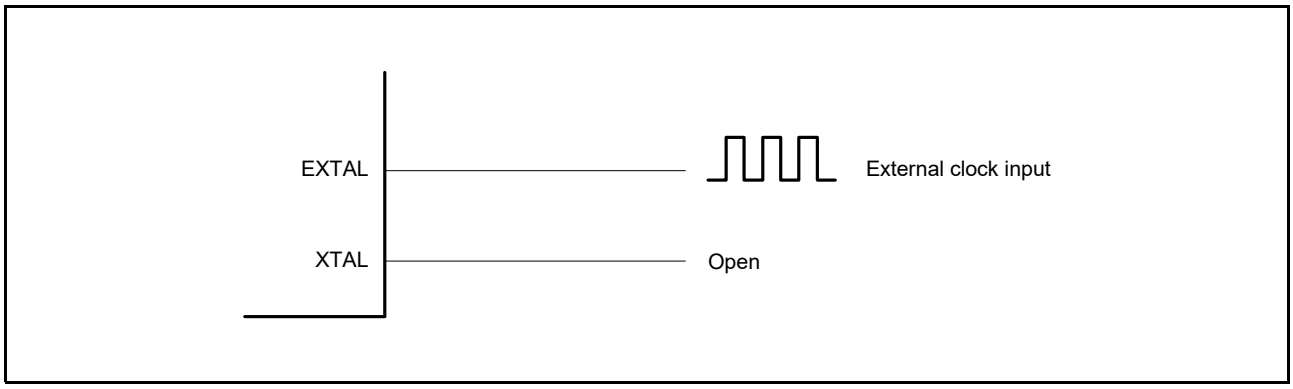


Figure 9.4 Example of Connection of External Clock

9.3.3 Notes on the External Clock Input

The frequency of the external clock input can only be changed while the main clock oscillator is stopped. Do not change the frequency of the external clock input while the setting of the main clock oscillator stop bit (MOSCCR.MOSTP) is 0 (making the main clock oscillator run).

9.4 Oscillation Stop Detection Function

9.4.1 Oscillation Stop Detection and Operation after Detection

The oscillation stop detection function is used to detect the main clock oscillator stop and to supply LOCO clock pulses from the low-speed on-chip oscillator as the system clock source instead of the main clock or PLL clock. When the HOCO is selected as the clock source for the PLL and the PLL clock is selected as the system clock, the system clock is not switched to the LOCO even if stopping of the main clock is detected.

An oscillation stop detection interrupt request can be generated when an oscillation stop is detected. In addition, the MTU and GPTW output can be forcedly driven to the high-impedance on the detection. For details, see section 22, Multi-Function Timer Pulse Unit 3 (MTU3d), section 23, Port Output Enable 3 (POE3D), and section 26, GPTW Port Output Enable (POEG).

In the MCU, the input clock remaining at a given level over a certain period due, for example, to a malfunction of the main clock oscillator, is the criterion to detect stopping of the main clock. For details of the detection period, refer to Table 49.52, Oscillation Stop Detection Circuit Characteristics.

When an oscillation stop is detected, the main clock or PLL clock selected by the clock source select bits (SCKCR3.CKSEL[2:0]) is switched to the LOCO clock by the corresponding selectors in the former stage.

Therefore, on detection of oscillation stopping while the PLL clock or system clock is selected as the source of the main clock, the system clock source is switched to the LOCO clock without a change of CKSEL[2:0].

Switching between the main clock and LOCO clock or between the PLL clock and LOCO clock is controlled by the oscillation stop detection flag (OSTDSR.OSTDF). Switching to the LOCO clock is carried out by setting the OSTDF flag.

After a reset is released, activate the main clock oscillator, and then, set the SCKCR3.CKSEL[2:0] bits to the main clock or PLL clock after the specified oscillation settling time has elapsed. If a resonator is selected as the source of the main clock oscillator, apply a reset to clear the OSTDF flag.

When the external clock input is selected as the source for the main clock oscillator, clearing the OSTDF flag by the software switches the clock source back to the main clock or PLL clock.

However, the OSTDF flag cannot be cleared if the main clock oscillator is selected with the SCKCR3.CKSEL [2:0] bits, or if the PLL clock is selected in a state that the main clock oscillator is set for a clock source. After detecting the oscillation stop, in order to set the clock source back to the main clock or the PLL clock, change the setting of the SCKCR3.CKSEL[2:0] bits to the setting other than the main clock oscillator and the PLL clock with the main clock oscillator as the clock source, and then, clear the OSTDF flag by the software. Subsequently, confirm that the OSTDF flag is not set again, and then, set the SCKCR3.CKSEL [2:0] bits to the main clock or PLL clock after the specified oscillation settling time has elapsed.

After a reset is released, the main clock oscillator is stopped and the oscillation stop detection function is disabled. To enable the oscillation stop detection function, activate the main clock oscillator and write 1 to the oscillation stop detection function enable bit (OSTDCR.OSTDE) after confirming that the OSCOVFSR.MOOVF flag or OSCOVFSR.PLOVF flag have been set to 1.

The oscillation stop detection function is provided against the main clock stop by an external cause. Therefore, the oscillation stop detection function should be disabled before the main clock oscillator is stopped by the software or a transition is made to software standby mode.

The clocks that are switched to the LOCO clock by the oscillation stop detection are: the main clock, PLL clock, CAC main clock (CACMCLK), and CANFD main clock (CANFDMCLK), which are provided as the system clock sources. Note that the frequencies of the derived clock signals after switching to the LOCO depends on the settings of the system clock control registers (SCKCR, SCKCR2, or SCKCR3).

Figure 9.5 shows an example of a flowchart for initialization of the oscillation stop detection function.

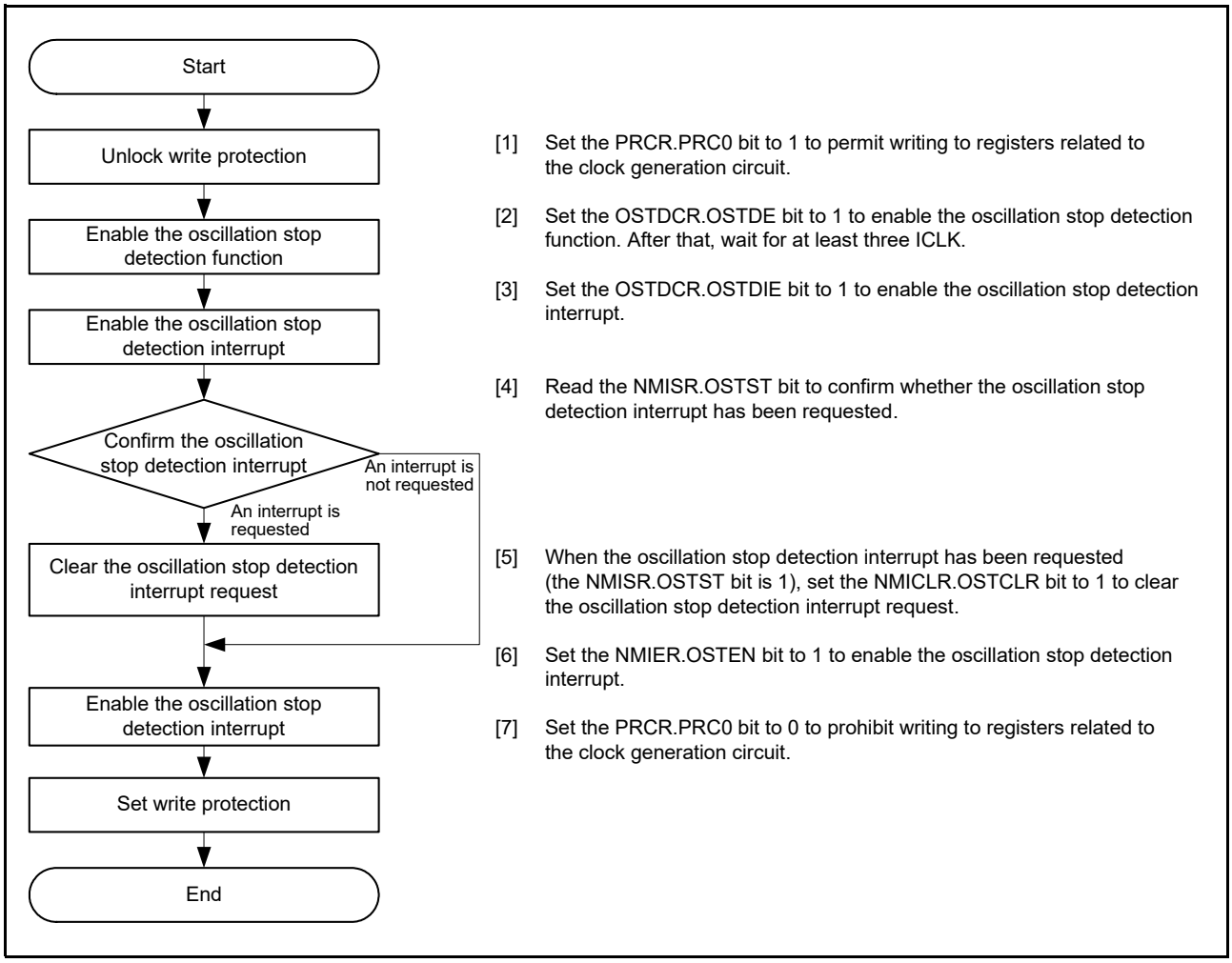
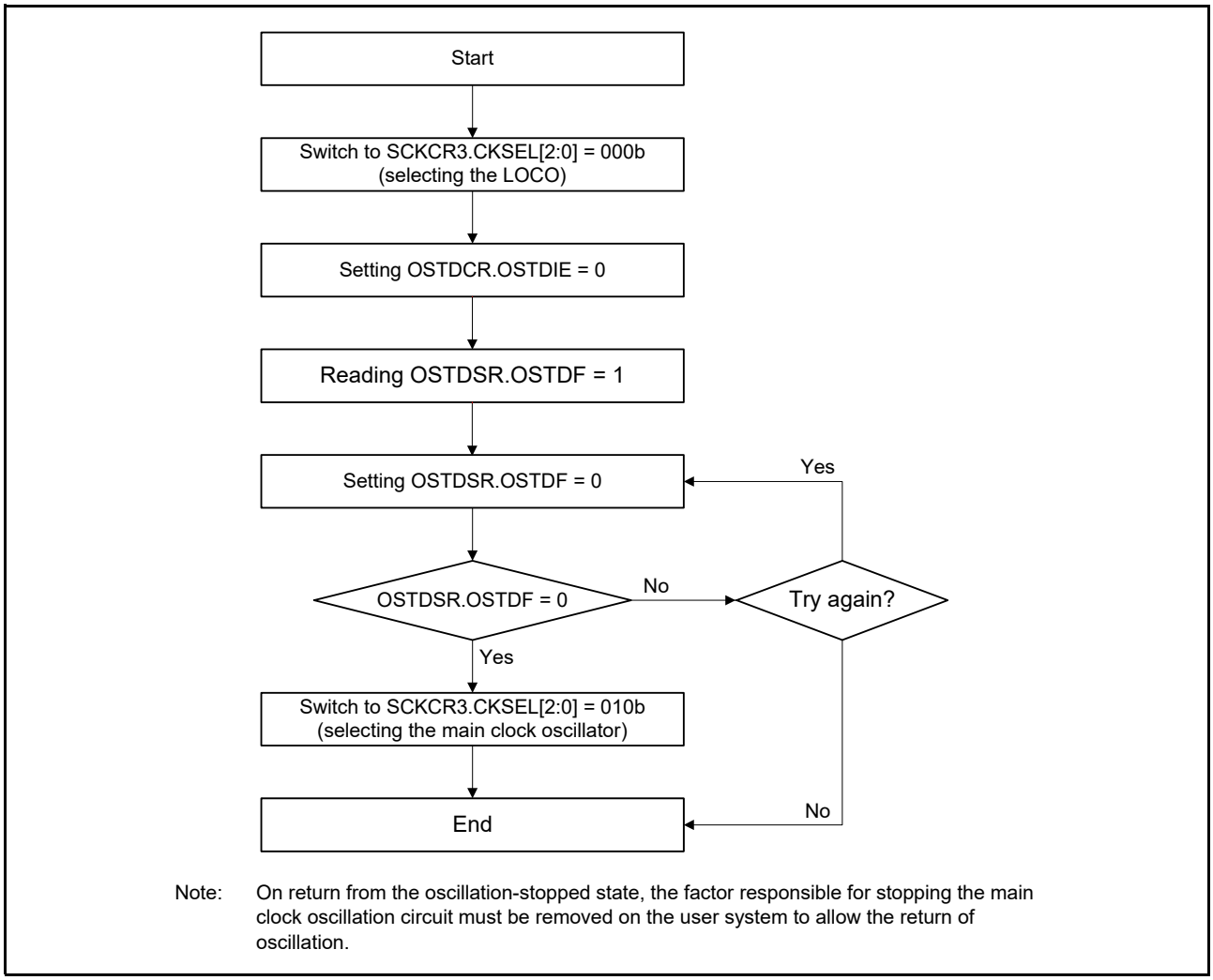


Figure 9.5 Flowchart Example for Initialization of Oscillation Stop Detection Function



Note: On return from the oscillation-stopped state, the factor responsible for stopping the main clock oscillation circuit must be removed on the user system to allow the return of oscillation.

Figure 9.6 Flowchart Example for Recovery from Detection of Oscillator Stop

9.4.2 Oscillation Stop Detection Interrupts

An oscillation-stop detection interrupt (OSTDI) will be generated if the oscillation-stop detection flag (OSTDSR.OSTDF) becomes 1 while the oscillation-stop detection interrupt enable bit in the oscillation stop detection control register (OSTDCR.OSTDIE) is 1 (enabling interrupt generation on oscillation stop detection). At this time, the main clock oscillator stop is notified to the port output enable 3 (POE3) and GPTW port output enable (POEG). On accepting the notification of the oscillation stop, the POE3 sets the OSTST high-impedance flag in input level control/ status register 6 (ICSR6.OSTSTF) to 1. After the oscillation stop is detected, wait for at least 10 cycles of PCLKB before writing to this ICSR6.OSTSTF flag. When the OSTDSR.OSTDF flag requires clearing, do so after clearing the oscillation-stop detection interrupt enable bit in the oscillation stop detection control register (OSTDCR.OSTDIE). Wait for at least two cycles of PCLKB clock before again setting the OSTDCR.OSTDIE bit to 1. According to the number of cycles for access to read out a given I/O register, waiting time longer than two cycles of PCLKB may have to be secured. While at the same time, on accepting the notification of the oscillation stop, the POEG sets the oscillation stop detection flag (POEGGn.OSTPF (n = A to D)) in the POEG group n setting register to 1.

If the oscillation stop detection interrupt is to be used as a non-maskable interrupt, since non-maskable interrupts are disabled in the initial state after a reset release, set the corresponding bit in the NMIER register to 1 by software to enable non-maskable interrupts. If it is to be used as a maskable interrupt, do not change the value of the NMIER register from the value after a reset. For details, see section 14, Interrupt Controller (ICUG).

9.5 PLL Circuit

The PLL circuit has a function to multiply the frequency from the oscillator.

9.6 Internal Clock

Clock sources of internal clock signals are the main clock, HOCO clock, LOCO clock, PLL clock, and dedicated clock for the IWDT. The internal clocks listed in the table below are produced from these sources.
Frequencies of the internal clocks are set by the combination of the divisors selected by the FCK[3:0], ICK[3:0], PCKA[3:0], PCKB[3:0], PCKC[3:0], and PCKD[3:0] bits in SCKCR, the CFDCK[3:0] bits in SCKCR2, the clock source selected by the CKSEL[2:0] bits in SCKCR3, the bits that select the frequency of the PLL circuit (STC[5:0] and PLIDIV[1:0] in PLLCR), and the HCFRQ[1:0] bits in HOCOCR2. If the value of any of these bits is changed, subsequent operation will be at the frequency determined by the new value.

Table 9.5 Internal Clocks and Supply Destination Modules

	Type of Internal Clock	Clock Name	Supply Destination Module
1	System clock	ICLK	CPU, TFU, code flash memory, RAM, ICU, BSC, DMAC, DTC, MPU
2	Peripheral module clocks	PCLKA	RSPI, RSPIA, RSCI, RI3C, CANFD, MTU (internal peripheral buses), GPTW (internal peripheral buses), and HRPWM (internal peripheral buses)
		PCLKB	TMR, CMT, CMTW, WDT, IWDT, POE3, SCIk, SCIh, RSCI, RIIC, CANFD, S12AD, R12DA, CMPC, CRC, DOC, CAC, TSIP-Lite, I/O ports, MPC, ICU, POEG, ELC, temperature sensor
		PCLKC	MTU (counter reference clock), GPTW (counter reference clock), HRPWM (reference clock)
		PCLKD	S12AD
3	Flash-IF clock	FCLK	Data flash memory, code flash memory (P/E)
4	CANFD clock	CANFDCLK	CANFD
5	CANFD main clock	CANFDMCLK	CANFD
6	CAC clocks	CACMCLK (Main clock)	CAC
		CACHCLK (HOCO clock)	
		CACLCLK (LOCO clock)	
		CACILCLK (IWDT-dedicated clock)	
7	IWDT-dedicated clock	IWDTCLK	IWDT

9.6.1 System Clock

The system clock (ICLK) is used as the operating clock of the CPU, ICU, BSC, MPU, DMAC, DTC, TFU, code flash memory, and RAM.
The ICLK frequency is specified by the ICK[3:0] bits in SCKCR, the CKSEL[2:0] bits in SCKCR3, the STC[5:0], and PLIDIV[1:0] bits in PLLCR, and the HCFRQ[1:0] bits in HOCOCR2.

9.6.2 Peripheral Module Clock

The peripheral module clocks (PCLKA, PCLKB, PCLKC, and PCLKD) are the operating clocks for use by peripheral modules.
The frequency of the given clock is specified by the PCKA[3:0], PCKB[3:0], PCKC[3:0], and PCKD[3:0] bits in SCKCR, the CKSEL[2:0] bits in SCKCR3, the STC[5:0], and PLIDIV[1:0] bits in PLLCR, and the HCFRQ[1:0] bits in HOCOCR2. The peripheral module clocks can be set to frequencies above that of the system clock.

9.6.3 Flash-IF Clock

The flash-interface clock (FCLK) is used as the operating clock for the flash-memory interfaces. That is, FCLK is used for programming and erasure of the code flash memory and data flash memory, and reading from the data flash memory.
The FCLK frequency is specified by the FCK[3:0] bits in SCKCR, the CKSEL[2:0] bits in SCKCR3, and the STC[5:0] and PLIDIV[1:0] bits in PLLCR, and the HCFRQ[1:0] bits in HOCOCR2.

9.6.4 CANFD Clock (CANFDCLK)

The CANFD clock (CANFDCLK) is used as the operating clock for the CANFD.
Use the SCKCR2.CFDCK[3:0], SCKCR3.CKSEL[2:0], PLLCR.STC[5:0], PLLCR.PLIDIV[1:0], or HOCOCR2.HCFRQ[1:0] bits to set the CANFDCLK frequency.

9.6.5 CANFD Main Clock (CANFDMCLK)

The CANFD main clock (CANFDMCLK) is an operating clock for the CANFD.
CANFDMCLK is generated by the main clock oscillator.

9.6.6 CAC Clock (CACCLK)

The CAC clock (CACCLK) is an operating clock for the CAC.
CACCLK includes CACMCLK generated by the main clock oscillator, CACHCLK generated by the high-speed on-chip oscillator, CACLCLK generated by the low-speed on-chip oscillator, CACILCLK generated by the IWDT-dedicated on-chip oscillator, and PCLKB supplied to peripheral modules.

9.6.7 IWDT-Dedicated Clock

The IWDT-dedicated clock (IWDTCLK) is the operating clock for the IWDT.
IWDTCLK is internally generated by the IWDT-dedicated on-chip oscillator.

9.7 Clock Source Switching

In this MCU, the clock signal from the LOCO, which oscillates during release from the reset state, is used to start the fetching of CPU instructions after the internal reset time (tRESWT) has elapsed. After that, set up the clock to which the CPU will be switched while it is still driven by the LOCO, and read the oscillation stabilization flag register to confirm that oscillation of the given clock signal is stable before switching to the selected clock source.

(1) Example of procedure for settings to switch the system clock source from the LOCO to the PLL (clock source for the PLL: main clock) after release from an internal reset

1. After release from the internal reset state, set the driving ability by writing to the MODRV2[1:0] bits in the MOFCR register.
2. Set the oscillation settling time for the main clock oscillator by writing to the MSTS[7:0] bits in the MOSCWTCR register.
3. Make the main clock oscillator run by setting the MOSTP bit in the MOSCCR register.
4. Set the frequency multiplication factor in the PLLCR register (the initial setting for the PLL clock source selects the main clock oscillator).
5. Select PLL operation by writing to the PLL stop control bit in the PLLCR2 register.
6. Confirm that the PLL clock has become stable by reading the PLOVF flag in the OSCOVFSR register.
7. Set the division ratios after switching the clock sources by the SCKCR and SCKCR2 registers.
8. Change the clock signal from the LOCO clock to the PLL clock by writing to the CKSEL[2:0] bits in the SCKCR3 register.

(2) Example of procedure for settings to switch the system clock source from the LOCO to the PLL (clock source for the PLL: HOCO) after release from an internal reset

1. After release from the internal reset state, set the frequency by writing to the HCFRQ[1:0] bits in the HOCOCR2 register.
2. Make the HOCO clock run by setting the HCSTP bit in the HOCOCR register (the initial value depends on the value of the OFS1.HOCOEN bit; if the OFS1.HOCOEN bit is 0, an explicit setting to make the HOCO clock run is not required. The oscillation settling time for the HOCO is 25 cycles of the LOCO).
3. Set the frequency multiplication factor and set the HOCO clock as the PLL clock source by writing to the PLLCR register.
4. Select PLL operation by writing to the PLL stop control bit in the PLLCR2 register.
5. Confirm that the PLL clock has become stable by reading the PLOVF flag in the OSCOVFSR register.
6. Set the division ratios after switching the clock sources by the SCKCR and SCKCR2 registers.
7. Change the clock signal from the LOCO clock to the PLL clock by writing to the CKSEL[2:0] bits in the SCKCR3 register.

(3) Example of procedure for settings to switch the system clock source from the LOCO to the main clock after release from an internal reset

1. After release from the internal reset state, set the driving ability by writing to the MODRV2[1:0] bits in the MOFCR register.
2. Set the oscillation settling time for the main clock oscillator by writing to the MSTS[7:0] bits in the MOSCWTCR register.
3. Make the main clock oscillator run by setting the MOSTP bit in the MOSCCR register.
4. Confirm that the main clock has become stable by reading the MOOVF flag in the OSCOVFSR register.
5. Set the division ratios after switching the clock sources by the SCKCR and SCKCR2 registers.
6. Change the clock signal from the LOCO clock to the main clock by writing to the CKSEL[2:0] bits in the SCKCR3 register.

(4) Example of procedure for settings to switch the system clock source from the LOCO to the HOCO after release from an internal reset

1. After release from the internal reset state, set the frequency by writing to the HCFRQ[1:0] bits in the HOCOCR2 register.
2. Make the HOCO clock run by setting the HCSTP bit in the HOCOCR register (the initial value depends on the value of the OFS1.HOCOEN bit; if the OFS1.HOCOEN bit is 0, an explicit setting to make the HOCO clock run is not required. The oscillation settling time for the HOCO is 25 cycles of the LOCO).
3. Confirm that the HOCO clock has become stable by reading the HCOVF flag in the OSCOVFSR register.
4. Set the division ratios after switching the clock sources by the SCKCR and SCKCR2 registers.
5. Change the clock signal from the LOCO clock to the HOCO clock by writing to the CKSEL[2:0] bits in the SCKCR3 register.

9.8 Operations Linked by the ELC

9.8.1 Event Signal Output to the ELC

The clock generation circuit is capable of operation linked with another module set in advance when its interrupt request signal is used as an even signal by the event link controller (ELC) on detection of stopping of the main clock oscillation. The clock generation circuit outputs the event signal regardless of the setting of the corresponding oscillation stop detection interrupt enable bit (OSTDCR.OSTDIE). For details, see section 19, Event Link Controller (ELC).

9.8.2 Clock Source Switching on Reception of the Event Signal from the ELC

The clock generation circuit is capable of switching the clock source to the low-speed on-chip oscillator in response to the event set in advance when the event specified in the ELSRn register of the ELC occurs.
While this function is in use, clock source switching on return from sleep mode cannot be used. For details, see section 11.2.6, Sleep Mode Return Clock Source Switching Register (RSTCKCR).

9.9 Usage Notes

9.9.1 Notes on Clock Generation Circuit

(1) The frequencies of the system clock (ICLK), peripheral module clock (PCLKA to PCLKD), and flash-IF clock (FCLK) supplied to each module change according to the settings of SCKCR. Each frequency should meet the following:
Select each frequency that is within the operation guaranteed range of clock cycle time (tcyc) specified in AC characteristics of electrical characteristics.
The frequencies must not exceed the ranges listed in Table 9.1.
The peripheral modules operate on the PCLKA, PCLKB, and PCLKC. Note therefore that the operating speed of modules such as the timer and SCI varies before and after the frequency is changed.

(2) The following relation is required between the frequencies of the peripheral module clocks.
PCLKC ≥ PCLKA ≥ PCLKB
PCLKA:PCLKC = 1:1 or 1:2
PCLKB:PCLKD = 1:1, 2:1, 4:1, or 1:2

(3) The following relations between frequencies must apply if the CAN FD module is to be used.
PCLKA:PCLKB = 2:1
PCLKB ≥ CANFDCLK
PCLKB ≥ CANFDMCLK

(4) If the clock frequency is to be changed by modifying the value of SCKCR, SCKCR2, or SCKCR3 register, wait for writing of the value to the register to be complete and the new frequency to be stable before starting subsequent processing. For the procedure to confirm the completion of writing to I/O registers, see (2) Notes on writing to I/O registers, in section 5, I/O Registers.

9.9.2 Note on Rewriting the SCKCR3 Register

When the SCKCR3.CKSEL[2:0] bits have been rewritten, the clock output is temporarily stopped to prevent the switch of the clock source from generating a clock pulse of short duration (glitch). The interrupt controller may not detect the following signals that were input during this period.

(1) An external pin interrupt or NMI pin interrupt with a pulse width shorter than 4 cycles of the PCLKB after the switch while the PCLKB frequency is the 1/1 of the clock source (the SCKCR.PCKB[3:0] bits are 0000b).

(2) An external pin interrupt or NMI pin interrupt with a pulse width shorter than 2.5 cycles of the PCLKB after the switch while the PCLKB frequency is the 1/2 of the clock source (the SCKCR.PCKB[3:0] bits are 0001b).

When the external pin interrupt or NMI pin interrupt is in use, input the signal with enough pulse width to exceed the time condition described in (1) and (2).

9.9.3 Notes on Resonator

Since various resonator characteristics relate closely to the user's board design, adequate evaluation is required on the user side before use, referencing the resonator connection example shown in this section. The circuit constants for the resonator depend on the resonator to be used and the stray capacitance of the mounting circuit. Therefore, the circuit constants should be determined in full consultation with the resonator manufacturer. The voltage to be applied between the resonator pins must be within the absolute maximum rating.

9.9.4 Notes on Board Design

When using a crystal resonator, place the resonator and its load capacitors as close to the XTAL and EXTAL pins as possible. Other signal lines should be routed away from the oscillation circuit as shown in Figure 9.7 to prevent electromagnetic induction from interfering with correct oscillation.

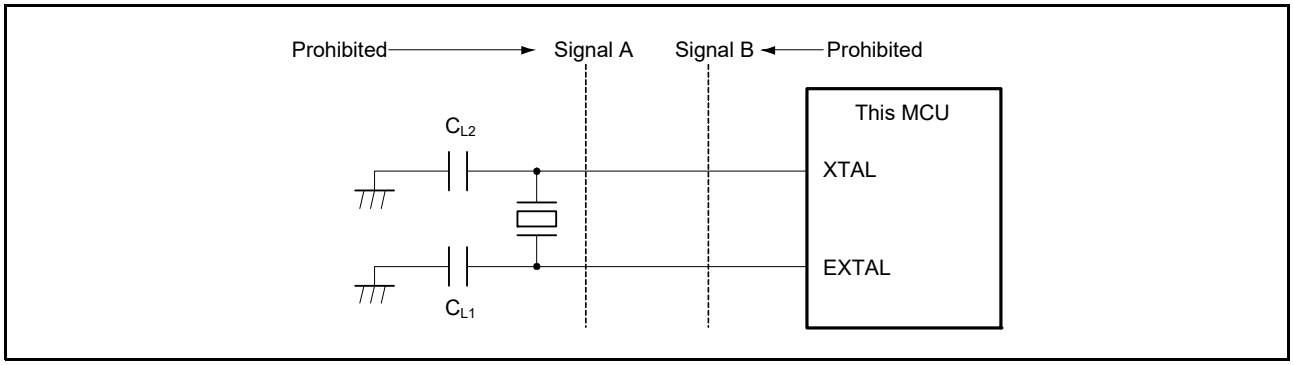


Figure 9.7 Notes on Board Design for Oscillation Circuit

9.9.5 Notes on Resonator Connect Pin

When the main clock is not used, the EXTAL and XTAL pins can be used as general ports P36 and P37. When they are used as the general ports, the main clock should be stopped (MOSCCR.MOSTP should be set to 1). However, with the system using the main clock, the EXTAL (P36) and XTAL (P37) pins should not be used as output ports.
For the values of registers related to port settings, refer to Table 21.25, Register Settings.

10. Clock Frequency Accuracy Measurement Circuit (CAC)

10.1 Overview

The clock frequency accuracy measurement circuit (CAC) counts pulses of the clock to be measured (measurement target clock) within the time generated by the clock to be used as a measurement reference (measurement reference clock), and determines the accuracy depending on whether the number of pulses is within the allowable range.
When measurement is completed or the number of pulses within the time generated by the measurement reference clock is not within the allowable range, an interrupt request is generated.
Table 10.1 lists the specifications of the CAC and Figure 10.1 shows a block diagram of the CAC.

Table 10.1 CAC Specifications

Item	Description
Measurement target clocks	The frequency of the following clocks can be measured. • Main clock • HOCO clock • LOCO clock • IWDT-dedicated clock (IWDTCLK) • Peripheral module clock B (PCLKB)
Measurement reference clocks	• External clock input to the CACREF pin • Main clock • HOCO clock • LOCO clock • IWDT-dedicated clock (IWDTCLK) • Peripheral module clock B (PCLKB)
Selectable function	Digital filter function
Interrupt sources	• Measurement end interrupt • Frequency error interrupt • Overflow interrupt
Low power consumption function	Module stop state can be set.

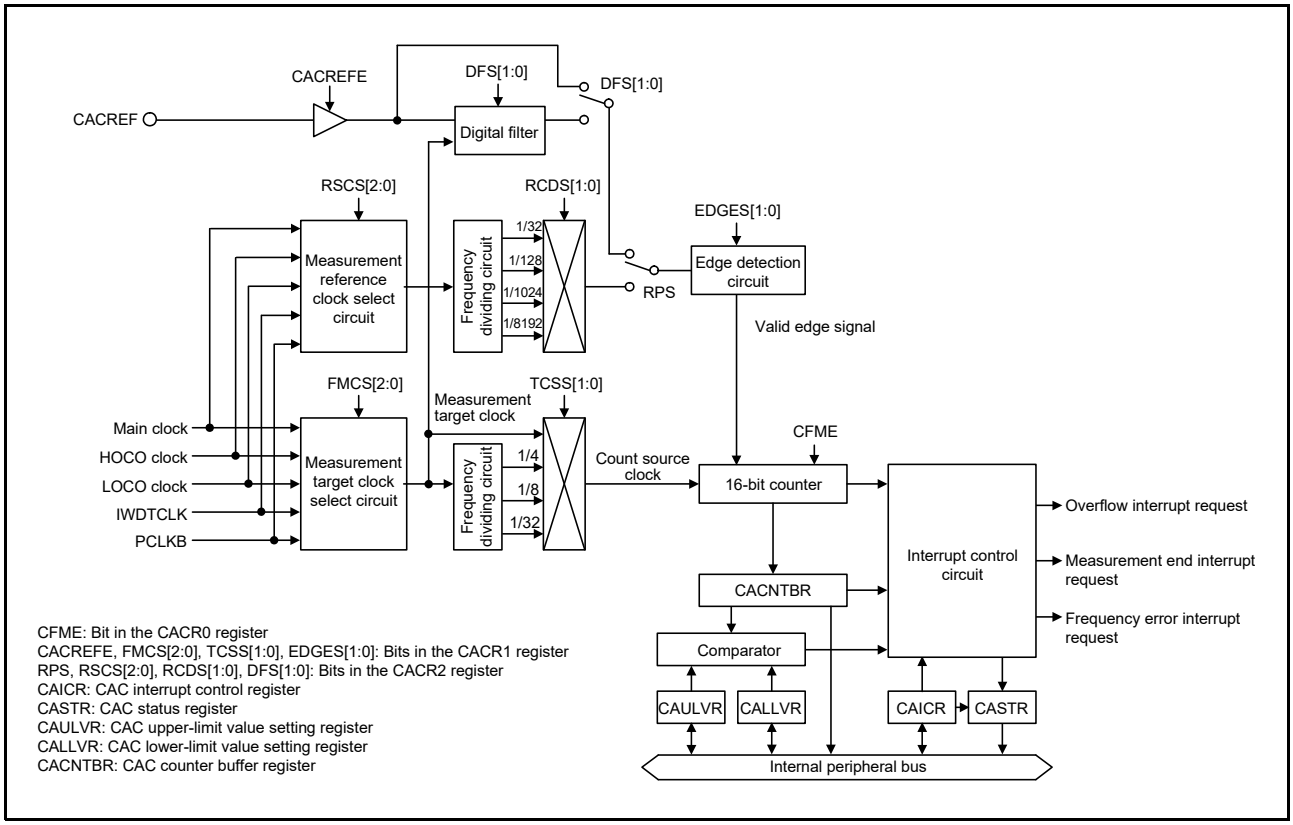


Figure 10.1 CAC Block Diagram

Table 10.2 shows the pin configuration of the CAC.

Table 10.2 Pin Configuration of CAC

Pin Name	I/O	Function
CACREF	Input	Measurement reference clock input pin

10.2 Register Descriptions

10.2.1 CAC Control Register 0 (CACR0)

Address(es): CAC.CACR0 0008 B000h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CFME
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CFME	Clock Frequency Measurement Enable	0: Clock frequency measurement is disabled. 1: Clock frequency measurement is enabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CFME Bit (Clock Frequency Measurement Enable)

This bit specifies whether clock frequency measurement is enabled or disabled.

When rewriting this bit, more time is required than other bits for the new value to be reflected in the register. Further write access to this bit are ignored until the current write access is reflected in the register. Read the bit to confirm that the rewrite has been reflected in the register.

10.2.2 CAC Control Register 1 (CACR1)

Address(es): CAC.CACR1 0008 B001h

	b7	b6	b5	b4	b3	b2	b1	b0
	EDGES[1:0]		TCSS[1:0]		FMCS[2:0]			CACREFE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CACREFE	CACREF Pin Input Enable	0: CACREF pin input is disabled. 1: CACREF pin input is enabled.	R/W
b3 to b1	FMCS[2:0]	Measurement Target Clock Select	b3 b1 0 0 0: Main clock 0 1 0: HOCO clock 0 1 1: LOCO clock 1 0 0: IWDT-dedicated clock (IWDTCLK) 1 0 1: Peripheral module clock B (PCLKB) Settings other than above are prohibited.	R/W
b5, b4	TCSS[1:0]	Timer Count Clock Source Select	b5 b4 0 0: No division 0 1: ×1/4 clock 1 0: ×1/8 clock 1 1: ×1/32 clock	R/W
b7, b6	EDGES[1:0]	Valid Edge Select	b7 b6 0 0: Rising edge 0 1: Falling edge 1 0: Both rising and falling edges 1 1: Setting prohibited	R/W

Note: Set the CACR1 register when the CACR0.CFME bit is 0.

CACREFE Bit (CACREF Pin Input Enable)

This bit specifies whether the CACREF pin input is enabled or disabled.

FMCS[2:0]Bits (Measurement Target Clock Select)

These bits select the measurement target clock whose frequency is to be measured.

TCSS[1:0] Bits (Timer Count Clock Source Select)

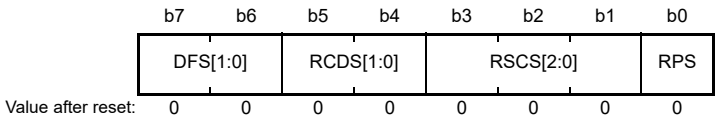
These bits select the count clock source for the clock frequency accuracy measurement circuit.

EDGES[1:0]Bits (Valid Edge Select)

These bits select the valid edge for the reference signal.

10.2.3 CAC Control Register 2 (CACR2)

Address(es): CAC.CACR2 0008 B002h



Bit	Symbol	Bit Name	Description	R/W
b0	RPS	Reference Signal Select	0: CACREF pin input 1: Internal clock (internally generated signal)	R/W
b3 to b1	RSCS[2:0]	Measurement Reference Clock Select	b3 b1 0 0 0: Main clock 0 1 0: HOCO clock 0 1 1: LOCO clock 1 0 0: IWDT-dedicated clock (IWDTCLK) 1 0 1: Peripheral module clock B (PCLKB) Settings other than above are prohibited.	R/W
b5, b4	RCDS[1:0]	Measurement Reference Clock Frequency Division Ration Select	b5 b4 0 0: ×1/32 clock 0 1: ×1/128 clock 1 0: ×1/1024 clock 1 1: ×1/8192 clock	R/W
b7, b6	DFS[1:0]	Digital Filter Select	b7 b6 0 0: Digital filtering is disabled. 0 1: The sampling clock for the digital filter is the measurement target clock. 1 0: The sampling clock for the digital filter is the measurement target clock divided by 4. 1 1: The sampling clock for the digital filter is the measurement target clock divided by 16.	R/W

Note: Set the CACR2 register when the CACR0.CFME bit is 0.

RPS Bit (Reference Signal Select)

This bit selects whether to use the CACREF pin input or an internal clock (internally generated signal) as the reference signal.

RSCS[2:0]Bits (Measurement Reference Clock Select)

These bits select the clock source for generating the measurement reference clock.

RCDS[1:0]Bits (Measurement Reference Clock Frequency Division Ration Select)

These bits select the frequency division ratio of the measurement reference clock.

DFS[1:0]Bits (Digital Filter Select)

The setting of these bits enables or disables the digital filter and selects its sampling clock.

10.2.4 CAC Interrupt Request Enable Register (CAICR)

Address(es): CAC.CAICR 0008 B003h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	OVFFCL	MENDFCL	FERRFCL	—	OVFIE	MENDIE	FERRIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	FERRIE	Frequency Error Interrupt Request Enable	0: Frequency error interrupt request is disabled. 1: Frequency error interrupt request is enabled.	R/W
b1	MENDIE	Measurement End Interrupt Request Enable	0: Measurement end interrupt request is disabled. 1: Measurement end interrupt request is enabled.	R/W
b2	OVFIE	Overflow Interrupt Request Enable	0: Overflow interrupt request is disabled. 1: Overflow interrupt request is enabled.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	FERRFCL	FERRF Clear	When 1 is written to this bit, the CASTR.FERRF flag is cleared. This bit is read as 0.	R/W
b5	MENDFCL	MENDF Clear	When 1 is written to this bit, the CASTR.MENDF flag is cleared. This bit is read as 0.	R/W
b6	OVFFCL	OVFF Clear	When 1 is written to this bit, the CASTR.OVFF flag is cleared. This bit is read as 0.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

FERRIE Bit (Frequency Error Interrupt Request Enable)

This bit specifies whether the frequency error interrupt request is enabled or disabled.

MENDIE Bit (Measurement End Interrupt Request Enable)

This bit specifies whether the measurement end interrupt request is enabled or disabled.

OVFIE Bit (Overflow Interrupt Request Enable)

This bit specifies whether the overflow interrupt request is enabled or disabled.

FERRFCL Bit (FERRF Clear)

Setting this bit to 1 clears the CASTR.FERRF flag.

MENDFCL Bit (MENDF Clear)

Setting this bit to 1 clears the CASTR.MENDF flag.

OVFFCL Bit (OVFF Clear)

Setting this bit to 1 clears the CASTR.OVFF flag.

10.2.5 CAC Status Register (CASTR)

Address(es): CAC.CASTR 0008 B004h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	OVFF	MENDF	FERRF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	FERRF	Frequency Error Flag	0: The clock frequency is within the range corresponding to the settings. 1: The clock frequency has deviated beyond the range corresponding to the settings (frequency error).	R
b1	MENDF	Measurement End Flag	0: Measurement is in progress. 1: Measurement has ended.	R
b2	OVFF	Overflow Flag	0: The counter has not overflowed. 1: The counter has overflowed.	R
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

FERRF Flag (Frequency Error Flag)

This flag indicates deviation of the clock frequency from the set value (frequency error).

[Setting condition]

- The clock frequency is outside of the setting range.

[Clearing condition]

- 1 is written to the CAICR.FERRFCL bit.

MENDF Flag (Measurement End Flag)

This flag indicates the end of measurement.

[Setting condition]

- Measurement has finished.

[Clearing condition]

- 1 is written to the CAICR.MENDFCL bit.

OVFF Flag (Overflow Flag)

This flag indicates that the counter has overflowed.

[Setting condition]

- The counter has overflowed.

[Clearing condition]

- 1 is written to the CAICR.OVFFCL bit.

10.2.6 CAC Upper-Limit Value Setting Register (CAULVR)

Address(es): CAC.CAULVR 0008 B006h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The CAULVR register is a 16-bit readable/writable register that specifies the upper-limit value of the counter used for measuring the frequency. When the frequency rises above the value specified in this register, a frequency error is detected.

Write to this register when the CACR0.CFME bit is 0.

The counter value held in the CACNTBR register can vary with the difference between the phases of the digital filter and edge-detection circuit on the one hand and the signal on the CACREF pin on the other, so ensure that this setting allows an adequate margin.

10.2.7 CAC Lower-Limit Value Setting Register (CALLVR)

Address(es): CAC.CALLVR 0008 B008h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The CALLVR register is a 16-bit readable/writable register that specifies the lower-limit value of the counter used for measuring the frequency. When the frequency falls below the value specified in this register, a frequency error is detected.

Write to this register when the CACR0.CFME bit is 0.

The counter value held in the CACNTBR register can vary with the difference between the phases of the digital filter and edge-detection circuit on the one hand and the signal on the CACREF pin on the other, so ensure that this setting allows an adequate margin.

10.2.8 CAC Counter Buffer Register (CACNTBR)

Address(es): CAC.CACNTBR 0008 B00Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The CACNTBR register is a 16-bit read-only register that retains the counter value at the time a valid reference signal edge is input.

10.3 Operation

10.3.1 Measuring Clock Frequency

The clock frequency accuracy measurement circuit measures the clock frequency using the CACREF pin input or the internal clock as a reference. Figure 10.2 shows an operating example of the clock frequency accuracy measurement circuit.

The clock frequency accuracy measurement circuit operates as shown below when measuring the clock frequency.

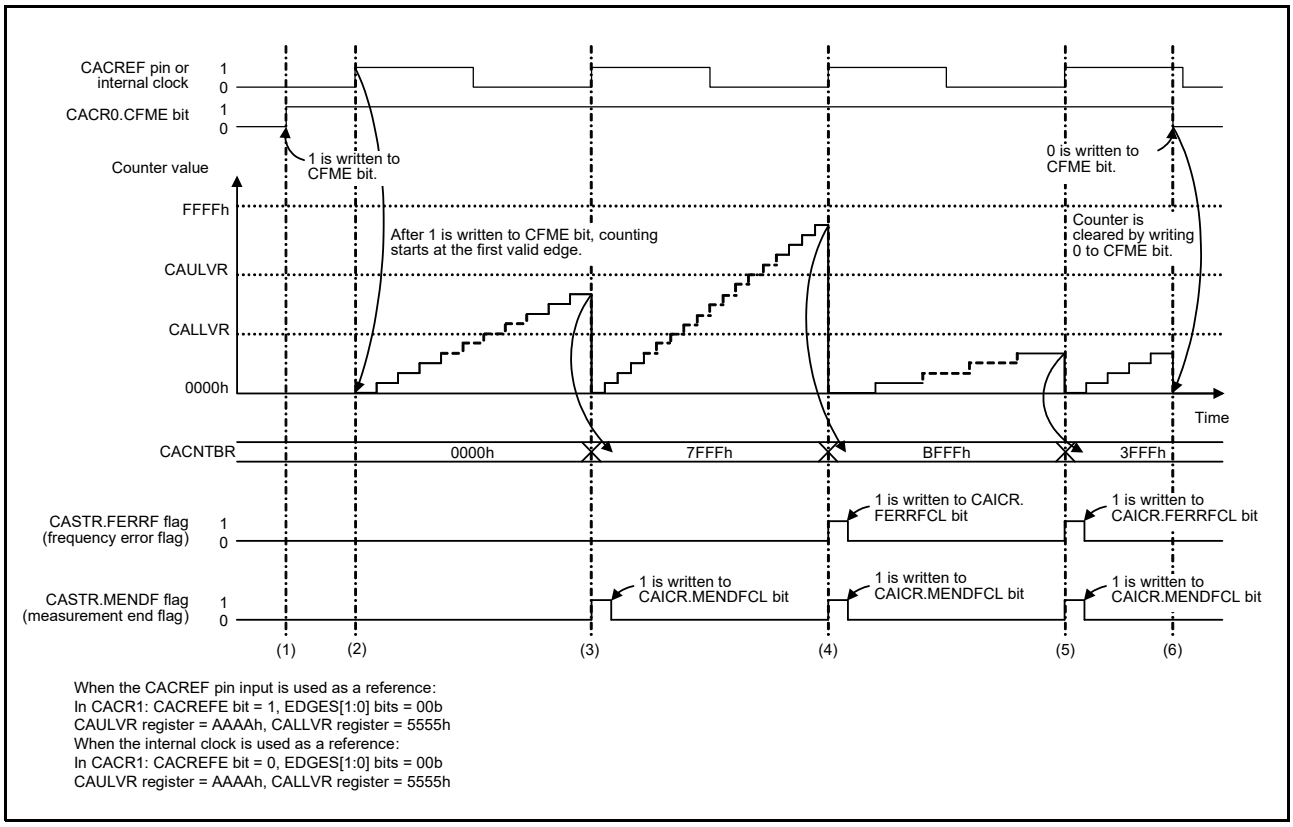


Figure 10.2 Operating Example of Clock Frequency Accuracy Measurement Circuit

(1) When the CACREF pin input is used as a reference (the CACR1.CACREFE bit = 1), clock frequency measurement is enabled by writing 1 to the CACR0.CFME bit while the CACR2.RPS bit is 0 and the CACR1.CACREFE bit is 1. On the other hand, when the internal clock is used as a reference (the CACR1.CACREFE bit = 0), clock frequency measurement is enabled by writing 1 to the CACR0.CFME bit while the CACR2.RPS bit is 1.

(2) When the CACREF pin input is used as a reference, the timer starts up-counting if the valid edge selected by the CACR1.EDGES[1:0] bits is input from the CACREF pin after 1 is written to the CFME bit. The valid edge is a rising edge (the CACR1.EDGES[1:0] bits = 00b) in Figure 10.2.
When the internal clock is used as a reference, the timer starts up-counting if the valid edge selected by the CACR1.EDGES[1:0] bits is input based on the clock source selected by the CACR2.RSCS[2:0] bits after 1 is written to the CFME bit. The valid edge is a rising edge (the CACR1.EDGES[1:0] bits = 00b) in Figure 10.2.

(3) When the next valid edge is input, the counter value is transferred in the CACNTBR register and compared with the values of the CAULVR and CALLVR registers. If the formula CALLVR ≤ CACNTBR ≤ CAULVR is satisfied, only the CASTR.MENDF flag is set to 1 because the clock frequency is correct. If the CAICR.MENDIE bit is 1, a measurement end interrupt is generated.

(4) When the next valid edge is input, the counter value is transferred in the CACNTBR register and compared with the values of the CAULVR and CALLVR registers. In the case of CACNTBR > CAULVR, the CASTR.FERRF flag is set to 1 because the clock frequency is erroneous. If the CAICR.FERRIE bit is 1, a frequency error interrupt is

generated. Also, the CASTR.MENDF flag is set to 1. If the CAICR.MENDIE bit is 1, a measurement end interrupt is generated.

(5) When the next valid edge is input, the counter value is transferred in the CACNTBR register and compared with the values of the CAULVR and CALLVR registers. In the case of CACNTBR < CALLVR, the CASTR.FERRF flag is set to 1 because the clock frequency is erroneous. If the CAICR.FERRIE bit is 1, a frequency error interrupt is generated. Also, the CASTR.MENDF flag is set to 1. If the CAICR.MENDIE bit is 1, a measurement end interrupt is generated.

(6) While the CACR0.CFME bit is 1, the counter value is transferred in the CACNTBR register and compared with the values of the CAULVR and CALLVR registers every time a valid edge is input. Writing 0 to the CACR0.CFME bit clears the counter and stops up-counting.

10.3.2 Digital Filtering of Signals on the CACREF Pin

The CACREF pin has a digital filter. Levels on the target pin for sampling are conveyed to the internal circuitry after matching three consecutive times at the selected sampling interval and the same level continues to be conveyed internally until the level on the pin again matches three consecutive times.

Enabling and disabling of the digital filter and its sampling clock are selectable.

The counter value transferred in the CACNTBR register may be in error by up to one cycle of the sampling clock due to the difference between the phases of the digital filter and the signal input to the CACREF pin.

When a frequency dividing clock is selected as a count source clock, the counter value error is obtained by the following formula:

Counter value error = (One cycle of the count source clock) / (One cycle of the sampling clock)

10.4 Interrupt Requests

The CAC generates three types of interrupt request: frequency error interrupt, measurement end interrupt, and overflow interrupt. When an interrupt source is generated, the corresponding status flag becomes 1. Table 10.3 lists details on the interrupt requests of the clock frequency accuracy measurement circuit.

Table 10.3 Interrupt Requests of Clock Frequency Accuracy Measurement Circuit

Interrupt Request	Interrupt Enable Bit	Status Flag	Interrupt Source
Frequency error interrupt	CAICR.FERRIE	CASTR.FERRF	The result of comparing CACNTBR to CAULVR and CALLVR is either CACNTBR > CAULVR or CACNTBR < CALLVR.
Measurement end interrupt	CAICR.MENDIE	CASTR.MENDF	A valid edge is input from the CACREF pin. Note however that a measurement end interrupt does not occur at the first valid edge after writing 1 to the CACR0.CFME bit.
Overflow interrupt	CAICR.OVFIE	CASTR.OVFF	The counter has overflowed.

10.5 Usage Notes

10.5.1 Module Stop Function Setting

CAC operation can be disabled or enabled using module stop control register C (MSTPCRC). The initial setting is for the CAC to be halted. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

11. Low Power Consumption

11.1 Overview

This MCU has several functions for reducing power consumption, including switching of clock signals to reduce power consumption, stopping modules, and transitions to low power consumption states.
Table 11.1 lists the specifications of low power consumption functions, and Table 11.2 lists the conditions to shift to low power consumption modes, states of the CPU and peripheral modules, and the method for release from each mode.
After a reset, this MCU enters the normal program execution state, but modules except for the DMAC, DTC, and RAM do not operate.

Table 11.1 Specifications of Low Power Consumption Functions

Item	Specification
Reducing power consumption by switching clock signals	The frequency division ratio is settable independently for the system clock (ICLK), peripheral module clock (PCLKA, PCLKB, PCLKC, PCLKD), and flash-IF clock (FCLK).*1
Module-stop function	Functions can be stopped independently for each peripheral module.
Function for transition to low power consumption mode	Transition to a low power consumption mode in which the CPU, peripheral modules, or oscillators are stopped is enabled.
Low power consumption modes	• Sleep mode • All-module clock stop mode • Software standby mode

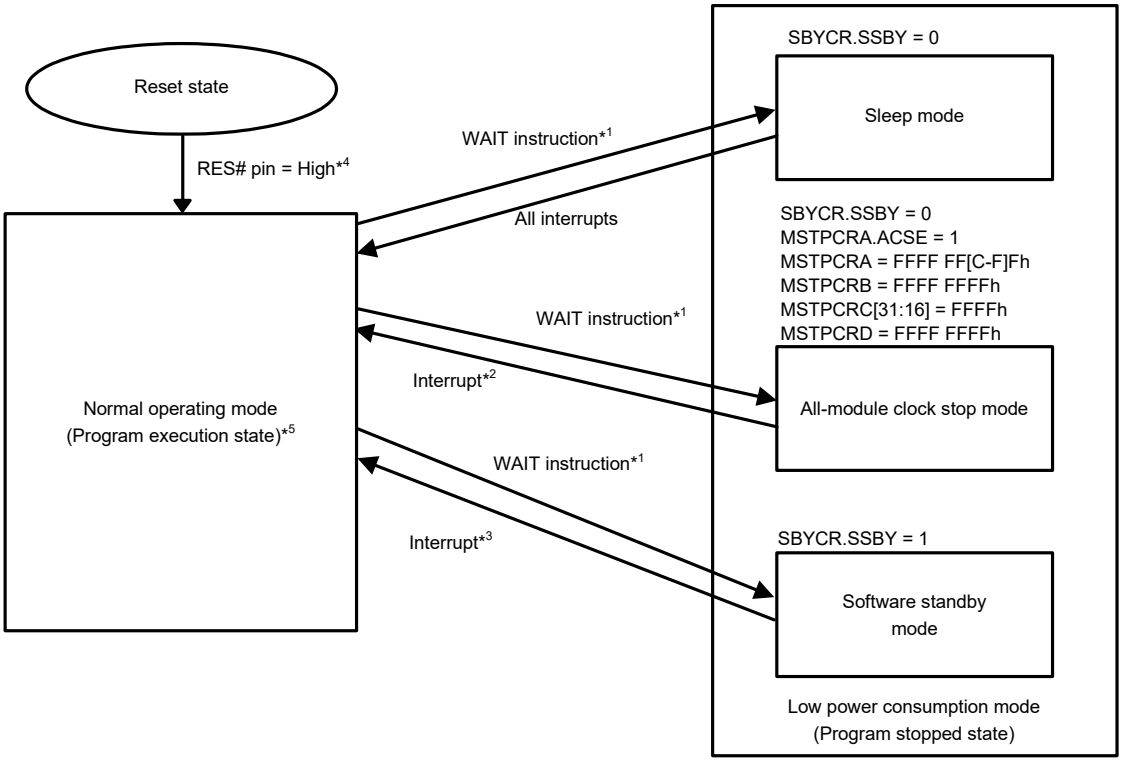
Note 1. For details, see section 9, Clock Generation Circuit.

Table 11.2 Entering and Exiting Low Power Consumption Modes and Operating States in Each Mode

Entering and Exiting Low Power Consumption Modes and Operating States	Sleep Mode	All-Module Clock Stop Mode	Software Standby Mode
Transition condition	Control register + instruction	Control register + instruction	Control register + instruction
Method of release other than reset	Interrupt	Interrupt*1	Interrupt*2
State after release*3	Program execution state (interrupt processing)	Program execution state (interrupt processing)	Program execution state (interrupt processing)
Main clock oscillator	Operating possible	Operating possible	Stopped
High-speed on-chip oscillator	Operating possible	Operating possible	Stopped
Low-speed on-chip oscillator	Operating possible	Operating possible	Stopped
IWDT-dedicated on-chip oscillator	Operating possible*4	Operating possible*4	Operating possible*4
PLL	Operating possible	Operating possible	Stopped
CPU	Stopped (Retained)	Stopped (Retained)	Stopped (Retained)
RAM	Operating possible (Retained)	Stopped (Retained)	Stopped (Retained)
Flash memory	Operating	Stopped (Retained)	Stopped (Retained)
Watchdog timer (WDT)	Stopped (Retained)	Stopped (Retained)	Stopped (Retained)
Independent watchdog timer (IWDT)	Operating possible*4	Operating possible*4	Operating possible*4
Port output enable (POE)	Operating possible	Operating possible*5	Stopped (Retained)
8-bit timer (unit 0, unit 1) (TMR)	Operating possible	Operating possible*6	Stopped (Retained)
Voltage detection circuit (LVD)	Operating possible	Operating possible	Operating possible
Power-on reset circuit	Operating	Operating	Operating
Peripheral modules	Operating possible	Stopped (Retained)	Stopped (Retained)
I/O ports	Operating	Retained*7	Retained

“Operating possible” means that operating or stopped can be controlled by the control register setting.
“Stopped (Retained)” means that internal register values are retained and internal operations are suspended.
“Stopped (Undefined)” means that internal register values are undefined and power is not supplied to the internal circuit.

Note 1. “Interrupts” here indicates an external pin interrupt (the NMI or IRQ0 to IRQ15) or any of peripheral interrupts (the 8-bit timer, IWDT, voltage monitoring 1, voltage monitoring 2, and main-clock oscillation stop detection).
Note 2. “Interrupts” here indicates an external pin interrupt (the NMI or IRQ0 to IRQ15) or any of peripheral interrupts (the IWDT, voltage monitoring 1, and voltage monitoring 2 interrupts).
Note 3. This does not include release initiated by the RES# pin reset, power-on reset, voltage monitoring reset, or independent watchdog-timer reset. The transition is to the reset state when release is initiated by one of these reset sources.
Note 4. Operation or stopping is selected by the setting of the IWDT sleep mode count stop control bit (IWDTSLCSTP) in the option function select register 0 (OFS0) in IWDT auto start mode. In any mode other than IWDT auto start mode, operation or stopping is selected by the setting of the sleep mode counter stop control bit (SLCSTP) in the IWDT counter stop control register (IWDTCSTPR).
Note 5. When a source condition for POE interrupts is satisfied while POE interrupts are enabled and the chip is in all-module clock stop mode, the flag for the source condition is retained but return from all-module clock stop mode does not proceed. If a different source initiates return from all-module clock stop mode in this situation, the POE interrupt is generated after that.
Note 6. Stopping or operation is controlled by the module-stop setting bits (MSTPA4 and MSTPA5, respectively) in module-stop control register A (MSTPCRA) for 8-bit timers 0 and 1 (unit 0) and 2 and 3 (unit 1).
Note 7. While the 8-bit timer is operated, the related pins continue operation.



Note 1. When an interrupt that acts as a trigger for release is received during the transition to the program-stopped state after the execution of a WAIT instruction, the transition to the program stopped state does not proceed and interrupt exception processing is executed instead.

Note 2. “Interrupts” here indicates an external pin interrupt (the NMI or IRQ0 to IRQ15) or any of peripheral interrupts (the 8-bit timer, IWDT, voltage-monitoring 1, voltage-monitoring 2, and oscillator-stopped detection interrupts). However, a, 8-bit timer interrupt is only effective if the value of the corresponding module-stop setting bit (MSTPA4 or MSTPA5, respectively) in module-stop control register A (MSTPCRA) for 8-bit timer 0 and 1 (unit 0) or 2 and 3 (unit 1) is 0.

Note 3. “Interrupts” here indicates an external pin interrupt (the NMI or IRQ0 to IRQ15) or any of peripheral interrupts (the IWDT, voltage-monitoring 1, and voltage-monitoring 2 interrupts).

Note 4. The LOCO is the source of the operating clock following a transition from the reset state to normal operating mode.

Note 5. The transition to normal operating mode is made due to an interrupt from sleep mode, all-module clock-stop mode, or software standby mode. In the case of recovery from sleep mode, the clock source for subsequent use is selectable. For details, refer to section 11.2.6, Sleep Mode Return Clock Source Switching Register (RSTCKCR). Also in the case of recovery from software standby mode, the clock source will be the same before and at the time of the transition.

When the RES# pin reset, power-on reset, voltage-monitoring 0 reset, voltage-monitoring 1 reset, voltage-monitoring 2 reset, watchdog timer reset, independent watchdog timer reset, or software reset is generated in any state, a transition to the reset state is made.

Figure 11.1 Mode Transitions

11.2 Register Descriptions

11.2.1 Standby Control Register (SBYCR)

Address(es): 0008 000Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SSBY	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b13 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b15	SSBY	Software Standby	0: Shifts to sleep mode or all-module clock stop mode after the WAIT instruction is executed 1: Shifts to software standby mode after the WAIT instruction is executed	R/W

SSBY Bit (Software Standby)

The SSBY bit specifies the transition destination after the WAIT instruction is executed.

When the SSBY bit is set to 1, the MCU enters software standby mode after execution of the WAIT instruction. When the MCU returns to normal operating mode after an interrupt has initiated release from software standby mode, the SSBY bit remains 1. Write 0 to this bit to clear it.

When the oscillation stop detection function enable bit in the oscillation stop detection control register (OSTDCR.OSTDE) is 1, the setting of the SSBY bit is ineffective. Even if the SSBY bit is 1, the MCU will enter sleep mode or all module clock stop mode on execution of the WAIT instruction.

When the code flash memory P/E mode entry bit in the flash P/E mode entry register (FENTRYR.FENTRYC) is 1 or the data flash memory P/E mode entry bit (FENTRYR.FENTRYD) is 1, the setting of this bit is ineffective. Sleep mode is entered on execution of the WAIT instruction even if this bit has been set to 1.

11.2.2 Module Stop Control Register A (MSTPCRA)

Address(es): 0008 0010h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ACSE	—	MSTPA 29	MSTPA 28	MSTPA 27	—	—	MSTPA 24	MSTPA 23	—	—	—	MSTPA 19	—	MSTPA 17	MSTPA 16
Value after reset:	0	1	0	0	0	1	1	0	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MSTPA 15	MSTPA 14	—	—	—	—	MSTPA 9	—	MSTPA 7	—	MSTPA 5	MSTPA 4	MSTPA 3	MSTPA 2	MSTPA 1	MSTPA 0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	MSTPA0	Compare Match Timer W (Unit 1) Module Stop	Target module: CMTW1 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b1	MSTPA1	Compare Match Timer W (Unit 0) Module Stop	Target module: CMTW0 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b2	MSTPA2	8-Bit Timer 7/6 (Unit 3) Module Stop	Target module: TMR7/TMR6 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b3	MSTPA3	8-Bit Timer 5/4 (Unit 2) Module Stop	Target module: TMR5/TMR4 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b4	MSTPA4	8-Bit Timer 3/2 (Unit 1) Module Stop	Target module: TMR3/TMR2 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b5	MSTPA5	8-Bit Timer 1/0 (Unit 0) Module Stop	Target module: TMR1/TMR0 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b6	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b7	MSTPA7	General PWM Timer/ High Resolution PWM/ GPTW Port Output Enable Module Stop	Target module: GPTW, HRPWM, POEG 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b8	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b9	MSTPA9	Multifunction Timer Pulse Unit 3 Module Stop	Target module: MTU 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b13 to b10	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b14	MSTPA14	Compare Match Timer (Unit 1) Module Stop	Target module: CMT unit 1 (CMT2, CMT3) 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b15	MSTPA15	Compare Match Timer (Unit 0) Module Stop	Target module: CMT unit 0 (CMT0, CMT1) 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b16	MSTPA16	12-bit A/D Converter (Unit 1) Module Stop	Target module: S12AD unit 1 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b17	MSTPA17	12-bit A/D Converter (Unit 0) Module Stop	Target module: S12AD unit 0 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b18	—	Reserved	This bit is read as 1. The write value should be 1.	R/W

Bit	Symbol	Bit Name	Description	R/W
b19	MSTPA19	12-bit D/A Converter Module Stop	Target module: 12-bit D/A 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b22 to b20	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b23	MSTPA23	12-bit A/D Converter (Unit 2) Module Stop	Target module: S12AD unit 2 (temperature sensor)*1 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b24	MSTPA24	Module Stop A24	Writing to and reading from this bit is enabled. When a transition to all-module clock stop mode is made, be sure that 1 has been written to this bit.	R/W
b26, b25	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b27	MSTPA27	Module Stop A27	Writing to and reading from this bit is enabled. When a transition to all-module clock stop mode is made, be sure that 1 has been written to this bit.	R/W
b28	MSTPA28	DMA Controller/Data Transfer Controller Module Stop	Target module: DMAC/DTC 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b29	MSTPA29	Module Stop A29	Writing to and reading from this bit is enabled. When a transition to all-module clock stop mode is made, be sure that 1 has been written to this bit.	R/W
b30	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b31	ACSE	All-Module Clock Stop Mode Enable	0: All-module clock stop mode is disabled 1: All-module clock stop mode is enabled	R/W

Note 1. The temperature sensor is controlled by a register in the S12AD unit 2.

ACSE Bit (All-Module Clock Stop Mode Enable)

The ACSE bit enables or disables a transition to all-module clock stop mode.

With the ACSE bit set to 1, when the CPU executes the WAIT instruction with the SBYCR.SSBY bit, MSTPCRA, MSTPCRB, MSTPCRC, and MSTPCRD satisfying specified conditions, the MCU enters all-module clock stop mode. For details, see section 11.5.2, All-Module Clock Stop Mode.

Whether to stop the 8-bit timers or not can be selected by the MSTPA5 and MSTPA4 bits.

When the MSTPCRA.ACSE bit = 0 while the SBYCR.SSBY = 0, a transition to sleep mode is made after the WAIT instruction is executed.

When the code flash memory P/E mode entry bit in the flash P/E mode entry register (FENTRYR.FENTRYC) is 1 or the data flash memory P/E mode entry bit (FENTRYR.FENTRYD) is 1, the setting of this bit is ineffective. Sleep mode is entered on execution of the WAIT instruction if this bit has been set to 1.

11.2.3 Module Stop Control Register B (MSTPCRB)

Address(es): 0008 0014h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	MSTPB30	—	—	—	MSTPB26	MSTPB25	—	MSTPB23	—	MSTPB21	—	—	—	MSTPB17	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	MSTPB10	MSTPB9	—	—	MSTPB6	—	MSTPB4	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b4	MSTPB4	Serial Communication Interface 12 Module Stop	Target module: SCI12 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b5	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b6	MSTPB6	Data Operation Circuit Module Stop	Target module: DOC 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b8, b7	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b9	MSTPB9	Event Link Controller Module Stop	Target module: ELC 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b10	MSTPB10	Comparator C Module Stop	Target module: CMPC 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b16 to b11	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b17	MSTPB17	Serial Peripheral Interface 0 Module Stop	Target module: RSPI0 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b20 to b18	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b21	MSTPB21	I^2C Bus Interface 0 Module Stop	Target module: RIIC0 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b22	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b23	MSTPB23	CRC Calculator Module Stop	Target module: CRC 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b24	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b25	MSTPB25	Serial Communication Interface 6 Module Stop	Target module: SCI6 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b26	MSTPB26	Serial Communication Interface 5 Module Stop	Target module: SCI5 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b29 to b27	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b30	MSTPB30	Serial Communication Interface 1 Module Stop	Target module: SCI1 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b31	—	Reserved	This bit is read as 1. The write value should be 1.	R/W

11.2.4 Module Stop Control Register C (MSTPCRC)

Address(es): 0008 0018h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	MSTPC27	MSTPC26	—	MSTPC24	—	—	—	—	MSTPC19	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MSTPC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MSTPC0	RAM Module Stop*1	Target module: RAM (0000 0000h to 0000 FFFFh) 0: RAM operating 1: RAM stopped	R/W
b15 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b18 to b16	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b19	MSTPC19 *2	CAC Module Stop	Target module: CAC 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b23 to b20	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b24	MSTPC24	Serial Communications Interface 11 Module Stop	Target module: RSCI11 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b25	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b26	MSTPC26	Serial Communications Interface 9 Module Stop	Target module: RSCI9 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b27	MSTPC27	Serial Communications Interface 8 Module Stop	Target module: RSCI8 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b31 to b28	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

Note 1. The MSTPC0 bit should not be set to 1 during access to the RAM. The RAM should not be accessed while the MSTPC0 bit is set to 1.

Note 2. The MSTPC19 bit should be rewritten while the oscillation of the clock to be controlled by this bit is stable. For entering software standby mode after writing a new value to this bit, wait for two cycles of the slowest clock among the clocks output by the oscillators actually oscillating, and then execute a WAIT instruction.

11.2.5 Module Stop Control Register D (MSTPCRD)

Address(es): 0008 001Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	MSTPD27	MSTPD26	—	—	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	MSTPD10	—	—	—	—	MSTPD5	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b5	MSTPD5	I3C Bus Interface 0 Module Stop	Target module: RI3C0 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b9 to b6	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b10	MSTPD10	CANFD Module Stop*1	Target module: CANFD 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b25 to b11	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b26	MSTPD26	Serial Peripheral Interface 0 Module Stop	Target module: RSPIA0 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b27	MSTPD27	Trusted Secure IP-Lite Module Stop	Target module: Trusted Secure IP-Lite 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b31 to b28	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

Note 1. The MSTPD10 bit should be rewritten while the oscillation of the clock to be controlled by this bit is stable. For entering software standby mode after writing a new value to this bit, wait for two cycles of the CANFD clock (CANFDCLK) and CANFD main clock (CANFDMCLK), and then execute a WAIT instruction.

11.2.6 Sleep Mode Return Clock Source Switching Register (RSTCKCR)

Address(es): 0008 00A1h

	b7	b6	b5	b4	b3	b2	b1	b0
	RSTCKEN	—	—	—	—	RSTCKSEL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	RSTCKSEL [2:0]	Sleep Mode Return Clock Source Select	b2 b0 0 0 1: HOCO is selected 0 1 0: Main clock oscillator is selected Settings other than above are prohibited while the RSTCKEN bit is 1.	R/W
b6 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	RSTCKEN	Sleep Mode Return Clock Source Switching Enable	0: Clock source switching on release from sleep mode is disabled 1: Clock source switching on release from sleep mode is enabled	R/W

The RSTCKCR register is used to control clock source switching at the time of release from sleep mode.
When operation is restored from sleep mode by setting RSTCKCR, the main clock oscillator stop bit in the main clock oscillator control register (MOSCCR.MOSTP) and HOCO stop bit in the high-speed on-chip oscillator control register (HOCOCR.HCSTP) corresponding to the clock source to be used on restoration are automatically modified to the operating state. The value of RSTCKSEL[2:0] bits is automatically reloaded to the clock source select bits in the system clock control register 3 (SCKCR3.CKSEL[2:0]).
The sleep mode return clock source switching function and clock source switching function by the ELC cannot be used at the same time. To enable the sleep mode return clock source switching function, write 1 to the RSTCKCR.RSTCKEN bit with the ELC clock source switching function disabled. The ELC clock source switching function should be enabled with the RSTCKCR.RSTCKEN bit being 0.
When the setting of register RSTCKCR is for the HOCO to be used in recovery from sleep mode, the power supply for the HOCO is not automatically switched on. If the HOCO to be used in recovery from sleep mode, the power supply for the HOCO must be on when the transition to sleep mode takes place.

RSTCKSEL[2:0] Bits (Sleep Mode Return Clock Source Select)

The RSTCKSEL[2:0] bits select the clock source to be used at the time of release from sleep mode.
The clock source selected by the RSTCKSEL[2:0] bits is enabled only when the RSTCKEN bit is 1.

RSTCKEN Bit (Sleep Mode Return Clock Source Switching Enable)

The RSTCKEN bit enables or disables clock source switching at the time of release from sleep mode.
On release from sleep mode, the clock source should be switched only when LOCO is selected as a clock for a transition to sleep mode. To make a transition to sleep mode with HOCO, main clock, or PLL selected as the clock source, the RSTCKEN bit should not be set to 1.

11.3 Reducing Power Consumption by Switching Clock Signals

When the SCKCR.FCK[3:0], ICK[3:0], PCKA[3:0], PCKB[3:0], PCKC[3:0], and PCKD[3:0] bits are set, the clock frequency changes. The CPU, DMAC, DTC, code flash memory, and RAM operate on the operating clock specified by the ICK[3:0] bits.
Peripheral modules operate on the operating clock specified by the PCKA[3:0], PCKB[3:0], PCKC[3:0], and PCKD[3:0] bits.
The data flash memory operates on the operating clock specified by the FCK[3:0] bits.
For details, see section 9, Clock Generation Circuit.

11.4 Module-Stop Function

The module-stop function can be set for each on-chip peripheral module.
When the MSTPmi bit (m = A to D, i = 31 to 0) in registers MSTPCRA to MSTPCRD is set to 1, the specified module stops operating and enters the module-stop state, but the CPU continues to operate independently. When the corresponding MSTPmi bit is set to 0, the module is released from the module-stop state and restarts operating at the end of the bus cycle. The internal states of modules are retained in the module-stop state.
After release from a reset, all modules except for the DMAC, DTC, and RAM are placed in the module-stop state.
Though read/write access cannot be made to the registers of the module that are in the module-stop state, some registers may be written to directly after the setting to the module-stop state. Therefore, care should be paid.

11.5 Low Power Consumption Modes

11.5.1 Sleep Mode

11.5.1.1 Transition to Sleep Mode

When the WAIT instruction is executed while the SBYCR.SSBY bit is 0, the CPU enters sleep mode. In sleep mode, the CPU stops operating but the contents of its internal registers are retained. Other peripheral functions do not stop.
When the WDT is used, the WDT stops counting when sleep mode is entered.
Counting by the IWDT stops if a transition to sleep mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 1 (stopping counting by the IWDT at transitions to low-power-consumption modes). In the same way, counting by the IWDT stops if a transition to sleep mode is made while the IWDT is being used in register start mode and the SLCSTP bit in IWDTCSTPR is 1.
Furthermore, counting by the IWDT continues if a transition to sleep mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 0 (counting by the IWDT continues through transitions to low-power-consumption modes). In the same way, counting by the IWDT continues if a transition to sleep mode is made while the IWDT is being used in register start mode and the SLCSTP bit in IWDTCSTPR is 0.
To use sleep mode, make the following settings and then execute a WAIT instruction.

(1) Clear the PSW.I bit*1 of the CPU to 0.
(2) Set the interrupt request destination*2 to be used for recovery from sleep mode to the CPU.
(3) Set the priority*3 of the interrupt to be used for recovery from sleep mode to a level higher than the setting of the PSW.IPL[3:0] bits*1 of the CPU.
(4) Set the IERm.IENj bit*3 for that interrupt to 1.
(5) For the last I/O register to which writing proceeded, read the register to confirm that the value written has been reflected.
(6) Execute the WAIT instruction (this automatically sets the I bit*1 in the PSW of the CPU to 1).

Note 1. For details, see section 2, CPU.
Note 2. For details, see section 14.7.3, Selecting Interrupt Request Destination.
Note 3. For details, see section 14, Interrupt Controller (ICUG).

11.5.1.2 Release from Sleep Mode

Release from sleep mode is initiated by a non-maskable interrupt, an interrupt, the RES# pin reset, a power-on reset, a voltage monitoring reset, or a reset caused by an IWDT underflow.

- Release triggered by an interrupt signal
  Generation of an interrupt triggers release from sleep mode and the interrupt exception processing starts. If a maskable interrupt has been masked by the CPU (the priority level*1 of the interrupt has been set to a value or lower than that of the PSW.IPL[3:0] bits*2 of the CPU), release from sleep mode does not proceed.
- Release due to a reset on the RES# pin
  When the RES# pin is driven low, the MCU enters the reset state. When the RES# pin is driven high after the reset signal is input for a predetermined time period, the CPU starts the reset exception processing.
- Release due to a power-on reset
  Release from sleep mode is initiated by a power-on reset.
- Release due to a voltage monitoring reset
  Release from sleep mode is initiated by a voltage monitoring reset from the voltage detection circuit.
- Release due to the independent watchdog timer reset
  Release from sleep mode is initiated by an internal reset generated by an IWDT underflow. However, when such conditions are set that stop IWDT counting in sleep mode (OFS0.IWDTSTRT = 0 and OFS0.IWDTSLCSTP = 1, or OFS0.IWDTSTRT = 1 and IWDTCSTPR.SLCSTP = 1), the IWDT is stopped and release from sleep mode is not initiated by the independent watchdog timer reset.

Note 1. For details, see section 14, Interrupt Controller (ICUG).
Note 2. For details, see section 2, CPU.

11.5.1.3 Sleep Mode Return Clock Source Switching Function

To switch the clock source used on return from sleep mode, the clock used after return needs to be set by the sleep mode return clock source switching register (RSTCKCR) and the wait control register needs to be set for each clock source. When the return interrupt is generated, after oscillation settling of the oscillator specified as the return clock, the clock source is automatically switched, and then operation returns from sleep mode. At this time, the registers related to clock source switching are automatically rewritten.
For details, see section 11.2.6, Sleep Mode Return Clock Source Switching Register (RSTCKCR). For setting a waiting time for oscillation stabilization, see section 9.2.14, Main Clock Oscillator Wait Control Register (MOSCWTCR).

11.5.2 All-Module Clock Stop Mode

11.5.2.1 Transition to All-Module Clock Stop Mode

After setting the MSTPCRA.ACSE bit to 1 and placing modules controlled by MSTPCRA, MSTPCRB, MSTPCRC, and MSTPCRD registers in the module-stop state (MSTPCRA = FFFF FF[C-F]Fh, MSTPCRB = FFFF FFFFh, MSTPCRC[31:16] = FFFFh, MSTPCRD = FFFF FFFFh), executing a WAIT instruction while the SBYCR.SSBY bit is 0 stops the bus controller, I/O ports, and all modules except for the 8-bit timers*1, POE3*2, IWDT, power-on reset circuit, voltage detection circuit at the end of the current bus cycle, and the chip enters all-module clock stop mode*3.
When the WDT is used, the WDT stops counting when all-module clock stop mode is entered.
Counting by the IWDT stops if a transition to all-module clock-stop mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 1 (stopping counting by the IWDT at transitions to low-power-consumption modes). In the same way, counting by the IWDT stops if a transition to all-module clock-stop mode is made while the IWDT is being used in register start mode and the SLCSTP bit in IWDTCSTPR is 1.
Furthermore, counting by the IWDT continues if a transition to all-module clock-stop mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 0 (counting by the IWDT continues through transitions to low-power-consumption modes). In the same way, counting by the IWDT continues if a transition to all-module clock-stop mode is made while the IWDT is being used in register start mode and the SLCSTP bit in IWDTCSTPR is 0.
To use all-module clock-stop mode, make the following settings and then execute a WAIT instruction.

(1) Clear the PSW.I bit*4 of the CPU to 0.
(2) Set the interrupt request destination*5 to be used for recovery from all-module clock stop mode to the CPU.
(3) Set the priority*6 of the interrupt to be used for recovery from all-module clock stop mode to a level higher than the setting of the PSW.IPL[3:0] bits*4 of the CPU.
(4) Set the IERm.IENj bit*6 for the interrupt to be used for recovery from all-module clock stop mode to 1.
(5) Read the last I/O register to have been written and confirm that its value reflects the value written.
(6) Execute a WAIT instruction (executing a WAIT instruction causes automatic setting of the PSW.I bit*4 of the CPU to 1).

Note 1. The MSTPCRA.MSTPA4 and MSTPA5 bits select operation or stop of these modules.
Note 2. When a POE3 interrupt source condition is satisfied while the setting to enable POE3 interrupts is in place, recovery from all-module clock stop mode does not proceed but the flag to indicate satisfaction of the source condition is retained. If a different source leads to recovery from all-module clock stop mode in this situation, a POE3 interrupt is generated after recovery.
Note 3. Transitions to all-module clock stop mode are not to be made in some states of DTC or DMAC operations. Before setting the MSTPCRA.MSTPA28 bit to 1, clear the DMAST.DMST bit of the DMAC and the DTCST.DTCST bit of the DTC so that the DTC and DMAC are not activated.
Note 4. For details, see section 2, CPU.
Note 5. For details, see section 14.7.3, Selecting Interrupt Request Destination.
Note 6. For details, see section 14, Interrupt Controller (ICUG).

11.5.2.2 Release from All-Module Clock Stop Mode

Release from all-module clock-stop mode is initiated by an external pin interrupt (the NMI or IRQ0 to IRQ15), a peripheral interrupt (8-bit timer*1, IWDT*2, voltage monitoring 1, voltage monitoring 2, or oscillator-stopped detection interrupt), a RES# pin reset, a power-on reset, a voltage monitoring reset, or an independent watchdog timer reset, and the transition to the normal program execution state proceeds via exception processing for the given interrupt or reset. However, note that in cases where a maskable interrupt has been masked by the CPU (priority level*3 of the interrupt has been set to a value or lower than that of the PSW.IPL[3:0] bits*4 of the CPU) or a maskable interrupt has been set up as a trigger to start the DTC or DMA transfer, release from all-module clock stop mode will not proceed.

Note 1. The MSTPA4 and MSTPA5 bits of MSTPCRA select operation or stopping of these modules.
Note 2. If a condition for the independent watchdog timer to stop counting applied (OFS0.IWDTSTRT = 0 and OFS0.IWDTSLCSTP = 1, or OFS0.IWDTSTRT = 1 and IWDTCSTPR.SLCSTP = 1) at the time of a transition to all-module clock-stop mode, using a reset from the independent watchdog timer to release the chip from all-module clock-stop mode is impossible because the independent watchdog timer is stopped.
Note 3. For details, see section 14, Interrupt Controller (ICUG).
Note 4. For details, see section 2, CPU.

11.5.3 Software Standby Mode

11.5.3.1 Transition to Software Standby Mode

When a WAIT instruction is executed with the SBYCR.SSBY bit set to 1, a transition to software standby mode is made. In this mode, the CPU, on-chip peripheral functions, and the oscillator functions stop. However, the contents of the CPU internal registers, RAM data, on-chip peripheral functions, and the states of the I/O ports are retained. Software standby mode allows significant reduction in power consumption because the oscillator stops in this mode. For details, see Table 11.2, Entering and Exiting Low Power Consumption Modes and Operating States in Each Mode.
Set the DMAST.DMST bit of the DMAC and the DTCST.DTCST bit of the DTC to 0 before executing the WAIT instruction.
When the WDT is used, the WDT stops counting when software standby mode is entered because the oscillator stops. Counting by the IWDT stops if a transition to software standby mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 1 (stopping counting by the IWDT at transitions to low-power-consumption modes). In the same way, counting by the IWDT stops if a transition to software standby mode is made while the IWDT is being used in register start mode and the SLCSTP bit in IWDTCSTPR is 1.
Furthermore, counting by the IWDT continues if a transition to software standby mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 0 (counting by the IWDT continues through transitions to low-power-consumption modes). In the same way, counting by the IWDT continues if a transition to software standby mode is made while the IWDT is being used in register start mode and the SLCSTP bit in IWDTCSTPR is 0.
When the oscillation stop detection function is enabled (OSTDCR.OSTDE = 1), software standby mode cannot be entered. To make a transition to software standby mode, execute a WAIT instruction after disabling the oscillation stop detection function (OSTDCR.OSTDE = 0).
To use software standby mode, make the following settings and then execute a WAIT instruction.

(1) Set the PSW.I bit*1 of the CPU to 0.
(2) Set the interrupt request destination*2 to be used for recovery from software standby mode to the CPU.
(3) Set the priority*3 of the interrupt to be used for recovery from software standby mode to a level higher than the setting of the PSW.IPL[3:0] bits*1 of the CPU.
(4) Set the IERm.IENj bit*3 for the interrupt to be used for recovery from software standby mode to 1.
(5) For the last I/O register to which writing proceeded, read the register to confirm that the value written has been reflected.
(6) Execute a WAIT instruction (executing a WAIT instruction causes automatic setting of the PSW.I bit*1 of the CPU to 1).

Note 1. For details, see section 2, CPU.
Note 2. For details, see section 14.7.3, Selecting Interrupt Request Destination.
Note 3. For details, see section 14, Interrupt Controller (ICUG).

11.5.3.2 Release from Software Standby Mode

Release from software standby mode is initiated by an external pin interrupt (the NMI or IRQ0 to IRQ15), peripheral interrupts (the IWDT, voltage monitoring 1, and voltage monitoring 2 interrupts), an RES# pin reset, a power-on reset, a voltage monitoring reset, or an independent watchdog timer reset.
When an interrupt initiates release from software standby, the oscillators which were stopped by the transition to software standby are restarted. After the oscillation of all these oscillators has become stable, operation returns from software standby.

(1) Release due to an interrupt
When an interrupt request from the NMI, IRQ0 to IRQ15, IWDT, voltage monitoring 1, or voltage monitoring 2 interrupt is generated, each oscillator which was operating before a transition to software standby mode resumes oscillation. After the time for return from software standby has elapsed, the chip is released from software standby and starts interrupt exception processing.
The time for return after release from software standby is the oscillation stabilization waiting time plus the time required for operations by the software standby release sequencer.

tSBYi = tSBYOSCWT + tSBYSEQ

tSBYi (i = MC, EX, PC, PE, PH, HO, LO): Time for return after release from software standby
tSBYOSCWT: Oscillation stabilization waiting time
tSBYSEQ: Time required for operations by the software standby release sequencer

For the oscillation stabilization waiting time to be used in calculating the time for return after release from software standby, use the greatest value of the oscillation stabilization waiting time of the oscillators which are to be started. For the oscillation stabilization waiting times of the oscillators, see section 49, Electrical Characteristics.

(2) Release due to a reset on the RES# pin
Clock oscillation starts when the low level is applied to the RES# pin. Clock supply for the MCU starts at the same time. Keep the level on the RES# pin low over the time required for oscillation of the clocks to become stable. Reset exception processing starts when the high level is applied to the RES# pin.

(3) Release due to a power-on reset
Release from software standby mode proceeds after a fall in the power-supply voltage leads to the generation of a power-on reset.

(4) Release due to a voltage monitoring reset
Release from software standby mode proceeds after a fall in the power-supply voltage leads to the generation of a voltage monitoring reset.

(5) Release due to an independent watchdog timer reset
An internal reset due to an underflow of the IWDT leads to release from software standby mode.
However, if a condition for the independent watchdog timer to stop counting applied (OFS0.IWDTSTRT = 0 and OFS0.IWDTSLCSTP = 1, or OFS0.IWDTSTRT = 1 and IWDTCSTPR.SLCSTP = 1) at the time of a transition to software standby, using a reset from the independent watchdog timer to release the chip from software standby is impossible because the independent watchdog timer is stopped.

11.5.3.3 Example of Software Standby Mode Application

Figure 11.2 shows an example where a transition to software standby mode is made at the falling edge of the IRQn pin, and release from software standby mode is initiated at the rising edge of the IRQn pin.
In this example, an IRQn interrupt is accepted with the IRQCRi.IRQMD[1:0] bits of the ICU set to 01b (falling edge), and then the IRQCRi.IRQMD[1:0] bits are set to 10b (rising edge). After that, the SBYCR.SSBY bit is set to 1 and the WAIT instruction is executed. Thus a transition to software standby mode is made. After that, release from software standby mode is initiated at the rising edge of the IRQn pin.
To return from software standby mode, settings of the interrupt controller (ICU) are also necessary. For details, see section 14, Interrupt Controller (ICUG).

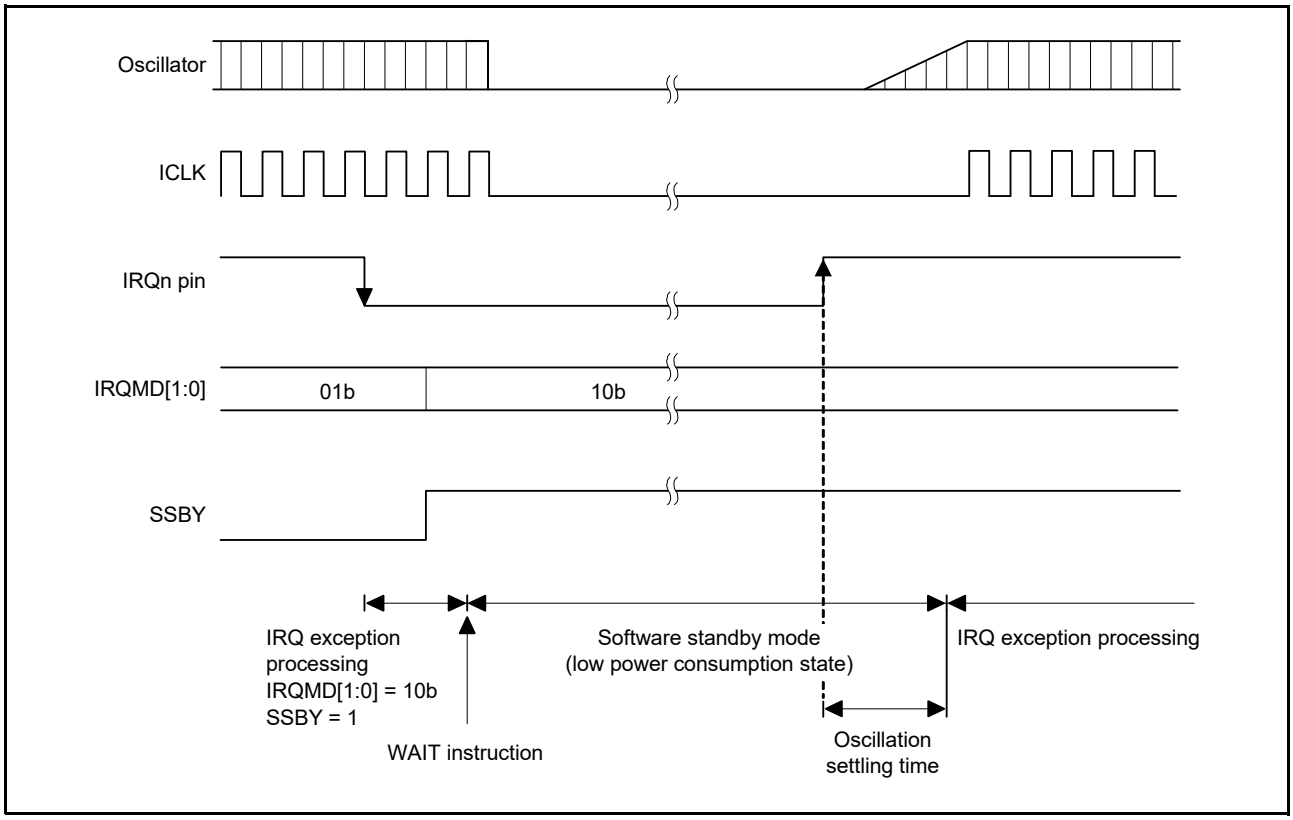


Figure 11.2 Example of Software Standby Mode Application

11.6 Usage Notes

11.6.1 I/O Port States

I/O port states are retained in software standby mode.

11.6.2 Module-Stop State of DMAC and DTC

Before setting the MSTPCRA.MSTPA28 bit to 1, set the DMAST.DMST bit of the DMAC and the DTCST.DTCST bit of the DTC to 0 so that the DMAC and DTC are not activated.
For details, see section 17, DMA Controller (DMACAa) and section 18, Data Transfer Controller (DTCb).

11.6.3 On-Chip Peripheral Module Interrupts

These interrupts do not operate in the module-stop state. Therefore, if the module-stop state is entered after an interrupt request is generated, a CPU interrupt source or a DMAC or DTC startup source cannot be cleared. For this reason, disable interrupts before entering the module-stop state.

11.6.4 Write Access to MSTPCRA, MSTPCRB, MSTPCRC, and MSTPCRD

Write accesses to registers MSTPCRA, MSTPCRB, MSTPCRC and MSTPCRD should be made only by the CPU.

11.6.5 Timing of WAIT Instruction

A WAIT instruction that follows register writing may be executed before the writing is completed. Accordingly, the WAIT instruction may be executed before the change to the setting of an I/O register is reflected, in which case operation may not be as intended. To avoid this, always execute the WAIT instruction after confirming that the last writing to the register has completed.

11.6.6 Rewriting the Register by DMAC and DTC in Sleep Mode

The WDT stops in sleep mode. Do not set up the DMAC and DTC to rewrite any registers related to the WDT while the chip is in sleep mode.
According to the settings of the OFS0.IWDTSLCSTP bit and IWDTCSTPR.SLCSTP bit, the IWDT may also stop in sleep mode. If that is the case, do not set up the DMAC and DTC to rewrite any registers related to the IWDT in sleep mode.
The RSTCKCR register can be set so that the clock source is switched on recovery from sleep mode. For this reason, rewriting the register while the chip is in sleep mode may lead to unintended operation, so do not allow rewriting of the RSTCKCR register in sleep mode.

12. Register Write Protection Function

The register write protection function protects important registers from being overwritten for in case a program runs out of control. The registers to be protected are set with the protect register (PRCR).

Table 12.1 lists the association between the PRCR bits and the registers to be protected.

Table 12.1 Association between PRCR Bits and Registers to be Protected

PRCR Bit	Register to be Protected
PRC0	• Registers related to the clock generation circuit: SCKCR, SCKCR2, SCKCR3, PLLCR, PLLCR2, MOSCCR, LOCOCR, ILOCOCR, HOCOCR, HOCOCR2, OSTDCR, OSTDSR
PRC1	• Registers related to the operating modes: SYSCR1, VOLSR • Registers related to the low power consumption functions: SBYCR, MSTPCRA, MSTPCRB, MSTPCRC, MSTPCRD, RSTCKCR • Registers related to clock generation circuit: MOSCWTCR, MOFCR, HOCOPCR • Software reset register: SWRR
PRC3	• Registers related to the LVD: LVCMPCR, LVDLVLR, LVD1CR0, LVD1CR1, LVD1SR, LVD2CR0, LVD2CR1, LVD2SR

12.1 Register Descriptions

12.1.1 Protect Register (PRCR)

Address(es): 0008 03FEh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PRKEY[7:0]								—	—	—	—	PRC3	—	PRC1	PRC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PRC0	Protect Bit 0	Enables writing to the registers related to the clock generation circuit. 0: Write disabled 1: Write enabled	R/W
b1	PRC1	Protect Bit 1	Enables writing to the registers related to operating modes, clock generation circuit, low power consumption, and software reset. 0: Write disabled 1: Write enabled	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	PRC3	Protect Bit 3	Enables writing to the registers related to the LVD. 0: Write disabled 1: Write enabled	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	PRKEY[7:0]	PRC Key Code	These bits control permission and prohibition of writing to the PRCR register. To modify the PRCR register, write A5h to the eight higher-order bits and the desired value to the eight lower-order bits as a 16-bit unit.	R/(W)*1

Note 1. Written values are not retained. These bits are read as 00h.

PRCi Bits (Protect Bit i) (i = 0, 1, 3)

These bits enable or disable writing to the corresponding registers to be protected.

Setting the PRCi bits to 1 and 0 enables and disables writing to the corresponding registers to be protected, respectively.

13. Exception Handling

13.1 Exception Events

During execution of a program by the CPU, the occurrence of a certain event may cause execution of that program to be suspended and execution of another program to be started. Such kinds of events are called exception events.
The RXv3 CPU supports eight types of exceptions. The types of exception events are shown in Figure 13.1.
The occurrence of an exception causes the processor mode to shift to supervisor mode.

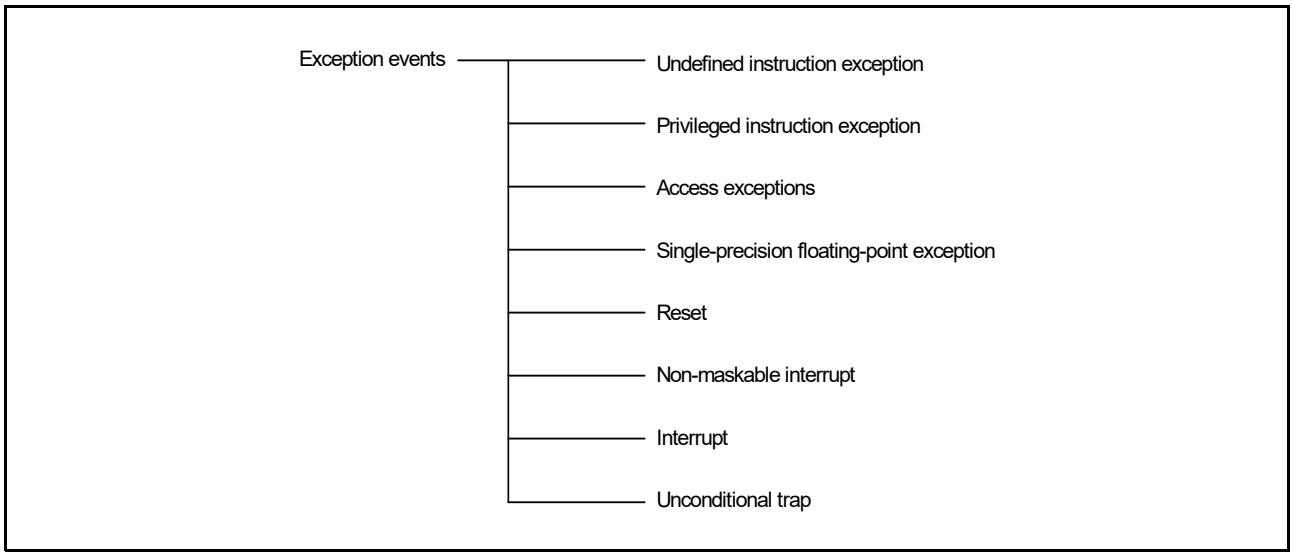


Figure 13.1 Types of Exception Events

13.1.1 Undefined Instruction Exception

An undefined instruction exception occurs when execution of an undefined instruction (an instruction not implemented) is detected.

13.1.2 Privileged Instruction Exception

A privileged instruction exception occurs when execution of a privileged instruction is detected in user mode. Privileged instructions can be executed only in supervisor mode.

13.1.3 Access Exceptions

An access exception occurs when an error is detected in access to memory by the CPU. If the memory-protection unit detects an instruction memory-protection error, an instruction-access exception occurs, and if the unit detects a data memory protection error, an operand-access exception occurs.

13.1.4 Single-Precision Floating-Point Exception

Single-precision floating-point exceptions are generated when any of the five exceptions specified in the IEEE 754 standard, namely overflow, underflow, inexact, division-by-zero, or invalid operation, or an attempt to use processing that is not implemented, is detected upon execution of a single-precision floating-point operation instruction. Exception handling by the CPU only proceeds when any among the EX, EU, EZ, EO, or EV bits in the FPSW, which corresponding to the five types of exception, is set to 1.

13.1.5 Reset

A reset is generated by input of a reset signal to the CPU. This has the highest priority of any exception and is always accepted.

13.1.6 Non-Maskable Interrupt

The non-maskable interrupt is generated by input of a non-maskable interrupt signal to the CPU and is only used when a fatal fault is considered to have occurred in the system. Never use the non-maskable interrupt with an attempt to return to the program that was being executed at the time of interrupt generation after the exception handling routine is ended.

13.1.7 Interrupt

Interrupts are generated by the input of interrupt signals to the CPU. A fast interrupt can be selected as the interrupt with the highest priority. In the case of the fast interrupt, hardware pre-processing and hardware post-processing are handled fast. The priority level of the fast interrupt is 15 (the highest). The exception handling of interrupts is masked when the I bit in PSW is 0.

13.1.8 Unconditional Trap

An unconditional trap is generated when the INT or BRK instruction is executed.

13.2 Exception Handling Procedure

In the exception handling, part of the processing is handled automatically by hardware and part of it is handled by a program (exception handling routine) that has been written by the user. Figure 13.2 shows the processing procedure when an exception other than a reset is accepted.

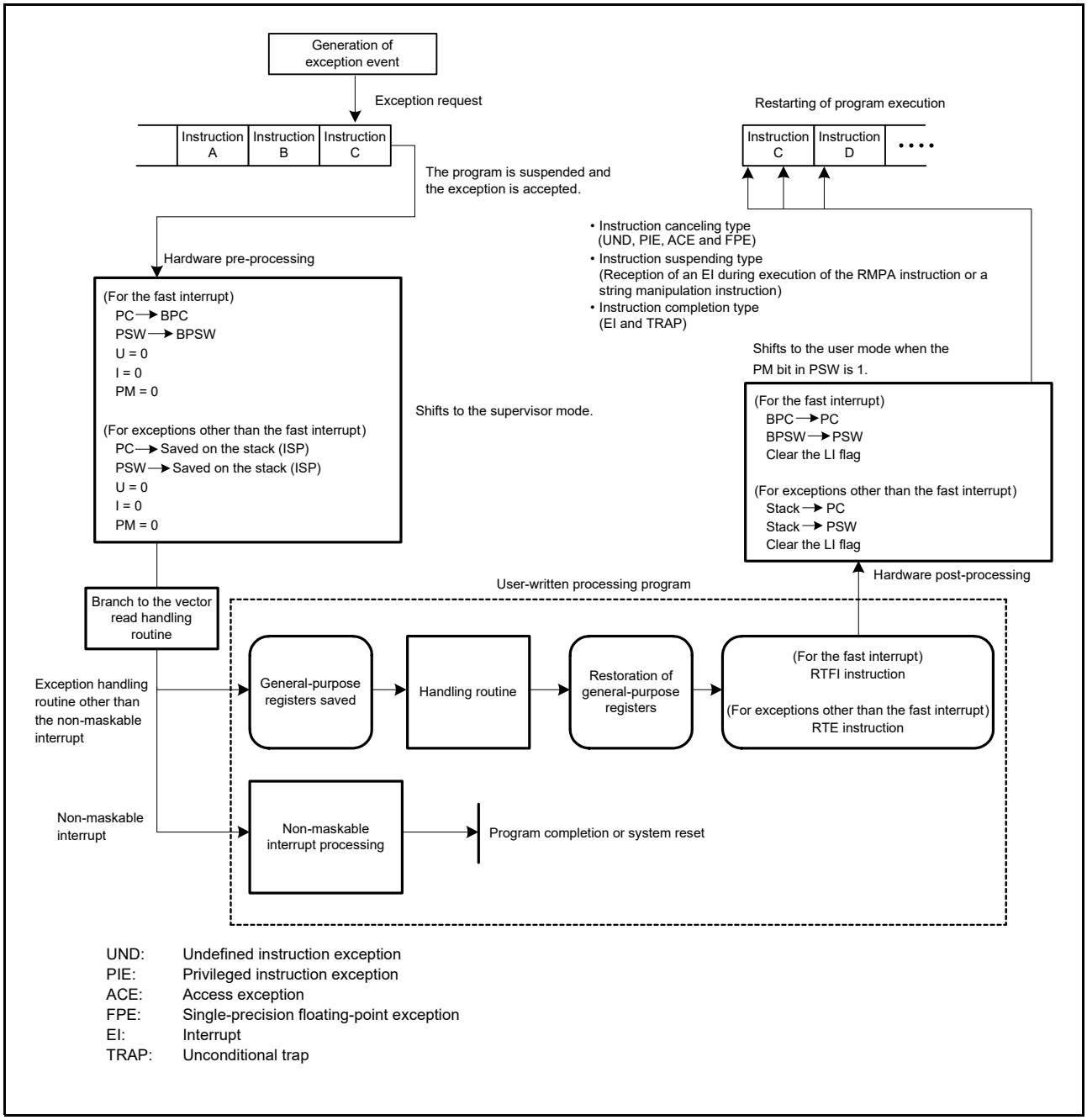


Figure 13.2 Outline of Exception Handling Procedure

When an exception is accepted, hardware processing by the RXv3 CPU is followed by access to the vector to acquire the address of the branch destination. In the vector, a vector address is allocated to each exception, and the branch destination address of the exception handling routine is written to each vector address.

Hardware pre-processing by the RXv3 CPU handles saving of the values of the program counter (PC) and processor status word (PSW). In the case of the fast interrupt, the values of the PC and PSW are saved in the backup PC (BPC) and the backup PSW (BPSW), respectively. In the case of exceptions other than the fast interrupt, the contents are saved on the stack. The values of general purpose registers and control registers other than the PC and PSW that are to be used within an exception handling routine must be saved by the user program at the start of the exception handling routine.

On completion of processing by an exception handling routine, saved registers are restored and the RTE instruction is executed to restore execution from the exception handling routine to the original program. For return from a fast interrupt, the RTFI instruction is used instead. In the case of a non-maskable interrupt, however, finish the program or reset the system without returning to the original program.

Hardware post-processing by the RXv3 CPU handles restoration of the contents of the PC and PSW. In the case of the fast interrupt, the values of the BPC and BPSW are restored to the PC and PSW, respectively. In the case of other exceptions, the values are restored from the stack to the PC and PSW.

The stack or the register-saving bank can be used to save and restore the general-purpose and other registers at the start and end of an exception handling routine.

Saving to and restoring from the register-saving bank is executed by using the SAVE and RSTR instructions. To save and restore a register that is not within the scope of saving and restoring by the SAVE and RSTR instructions, use the PUSH and POP instructions for saving to and restoring from the stack.

Using the register-saving bank is usually faster than using the stack, except when the number of registers that require saving and restoring in transitions to and from an exception-handling routine is extremely small.

13.3 Acceptance of Exception Events

When an exception occurs, the CPU suspends the execution of the program and processing branches to the exception handling routine.

13.3.1 Acceptance Timing and Saved PC Value

Table 13.1 lists the timing of acceptance and the program counter (PC) value to be saved for each exception event.

Table 13.1 Acceptance Timing and Saved PC Value

Exception Event		Type of Handling	Acceptance Timing	Value Saved in BPC or on the Stack
Undefined instruction exception		Instruction canceling type	During instruction execution	PC value of the instruction that generated the exception
Privileged instruction exception		Instruction canceling type	During instruction execution	PC value of the instruction that generated the exception
Access exception		Instruction canceling type	During instruction execution	PC value of the instruction that generated the exception
Single-precision floating-point exception		Instruction canceling type	During instruction execution	PC value of the instruction that generated the exception
Reset		Instruction abandonment type	Any machine cycle	None
Non-maskable interrupt	During execution of the RMPA, SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, and SWHILE instructions	Instruction suspending type	During instruction execution	PC value of the instruction being executed
	Other than above	Instruction completion type	At the next break between instructions	PC value of the next instruction
Interrupt	During execution of the RMPA, SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, and SWHILE instructions	Instruction suspending type	During instruction execution	PC value of the instruction being executed
	Other than above	Instruction completion type	At the next break between instructions	PC value of the next instruction
Unconditional trap		Instruction completion type	At the next break between instructions	PC value of the next instruction

13.3.2 Vector and Site for Saving the Values in the PC and PSW

The vector for each type of exception and the site for saving the values of the program counter (PC) and processor status word (PSW) are listed in Table 13.2. The addresses where the exception vector table and interrupt vector table start must be set. For details, see section 2.6, Vector Table.

Table 13.2 Vector and Site for Saving the Values in the PC and PSW

Exception		Vector	Site for Saving the Values in the PC and PSW
Undefined instruction exception		Exception vector table (EXTB)	Stack
Privileged instruction exception		Exception vector table (EXTB)	Stack
Access exception		Exception vector table (EXTB)	Stack
Single-precision floating-point exception		Exception vector table (EXTB)	Stack
Reset		Exception vector table (EXTB)	Nowhere
Non-maskable interrupt		Exception vector table (EXTB)	Stack
Interrupt	Fast interrupt	FINTV	BPC and BPSW
	Other than above	Interrupt vector table (INTB)	Stack
Unconditional trap		Interrupt vector table (INTB)	Stack

13.4 Hardware Processing for Accepting and Returning from Exceptions

This section describes the hardware processing for accepting and returning from exceptions other than a reset.

(1) Hardware Pre-Processing for Accepting an Exception

(a) Saving PSW

- For a fast interrupt
  PSW → BPSW
- For exceptions other than a fast interrupt
  PSW → Stack

Note: The FPSW is not saved by the hardware pre-processing. If single-precision floating-point operation instructions are used within an exception-handling routine, the user must save the FPSW on the stack within the exception-handling routine.

(b) Updating PM, U, and I Bits in PSW

I: Set to 0
U: Set to 0
PM: Set to 0

(c) Saving PC

- For a fast interrupt
  PC → BPC
- For exceptions other than a fast interrupt
  PC → Stack

(d) Setting Branch Destination Address of Exception Handling Routine in PC

Processing is shifted to the exception handling routine by acquiring the vector corresponding to the exception and then branching accordingly.

(2) Hardware Post-Processing for Execution of RTE and RTFI Instructions

(a) Restoring PSW

- For a fast interrupt
  BPSW → PSW
- For exceptions other than a fast interrupt
  Stack → PSW

(b) Restoring PC

- For a fast interrupt
  BPC → PC
- For exceptions other than a fast interrupt
  Stack → PC

(c) Clearing the LI flag

13.5 Hardware Pre-Processing

The hardware pre-processing from reception of each exception request to execution of the associated exception handling routine are explained below.

13.5.1 Undefined Instruction Exception

1. The value of the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are set to 0.
3. The value of the program counter (PC) is saved on the stack (ISP).
4. The vector is fetched from the value of EXTB + address 0000 005Ch.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

13.5.2 Privileged Instruction Exception

1. The value of the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are set to 0.
3. The value of the program counter (PC) is saved on the stack (ISP).
4. The vector is fetched from the value of EXTB + address 0000 0050h.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

13.5.3 Access Exceptions

1. The value of the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are set to 0.
3. The value of the program counter (PC) is saved on the stack (ISP).
4. The vector is fetched from the value of EXTB + address 0000 0054h.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

13.5.4 Single-Precision Floating-Point Exception

1. The value of the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are set to 0.
3. The value of the program counter (PC) is saved on the stack (ISP).
4. The vector is fetched from the value of EXTB + address 0000 0064h.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

13.5.5 Reset

1. The control registers are initialized.
2. The vector is fetched from address FFFF FFFCh.
3. The fetched vector is set to the PC.

13.5.6 Non-Maskable Interrupt

1. The value of the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are set to 0.
3. If the interrupt was generated during the execution of an RMPA, SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, or SWHILE instruction, the value of the program counter (PC) for that instruction is saved on the stack (ISP). For other instructions, the PC value of the next instruction is saved.
4. The processor interrupt priority level bits (IPL[3:0]) in PSW are set to Fh.
5. The vector is fetched from the value of EXTB + address 0000 0078h.
6. The fetched vector is set to the PC and processing branches to the exception handling routine.

13.5.7 Interrupt

1. The value of the processor status word (PSW) is saved on the stack (ISP) or, for the fast interrupt, in the backup PSW (BPSW).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are set to 0.
3. If the interrupt was generated during the execution of an RMPA, SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, or SWHILE instruction, the value of the program counter (PC) for that instruction is saved. For other instructions, the PC value of the next instruction is saved. Saving of the PC is in the backup PC (BPC) for fast interrupts.
4. The processor interrupt priority level bits (IPL[3:0]) in PSW indicate the interrupt priority level of the interrupt.
5. The vector for an interrupt source other than the fast interrupt is fetched from the interrupt vector table. For the fast interrupt, the address is fetched from the fast interrupt vector register (FINTV).
6. The fetched vector is set to the PC and processing branches to the exception handling routine.

13.5.8 Unconditional Trap

1. The value of the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are set to 0.
3. The value of the program counter (PC) for the next instruction is saved on the stack (ISP).
4. For the INT instruction, the value at the vector corresponding to the INT instruction number is fetched from the interrupt vector table.
   For the BRK instruction, the value at the vector from the start address is fetched from the interrupt vector table.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

13.6 Return from Exception Handling Routine

Executing the instruction listed in Table 13.3 at the end of the corresponding exception handling routine leads to restoration of the values of the program counter (PC) and processor status word (PSW) that were saved on the stack or in the control registers (BPC and BPSW) immediately before the exception handling sequence.

Table 13.3 Return from Exception Handling Routine

Exception		Instruction for Return
Undefined instruction exception		RTE
Privileged instruction exception		RTE
Access exception		RTE
Single-precision floating-point exception		RTE
Reset		Return is impossible
Non-maskable interrupt		Prohibited
Interrupt	Fast interrupt	RTFI
	Other than above	RTE
Unconditional trap		RTE

13.7 Priority of Exception Events

The priority of exception events is listed in Table 13.4. When multiple exceptions are generated at the same time, the exception with the highest priority is accepted first.

Table 13.4 Priority of Exception Events

Priority		Exception Event
High	1	Reset
↑	2	Non-maskable interrupt
	3	Interrupt
	4	Instruction access exception
	5	Undefined instruction exception Privileged instruction exception
	6	Unconditional trap
	7	Operand access exception
Low	8	Single-precision floating-point exception

14. Interrupt Controller (ICUG)

14.1 Overview

The interrupt controller (ICU) controls various interrupt requests from the peripheral modules and the IRQi pin (i = 0 to 15), and generates an interrupt request to the CPU and a transfer request to the DTC and DMAC.
Table 14.1 lists the ICU specifications, and Figure 14.1 shows a block diagram of the interrupt controller.

Table 14.1 ICU Specifications

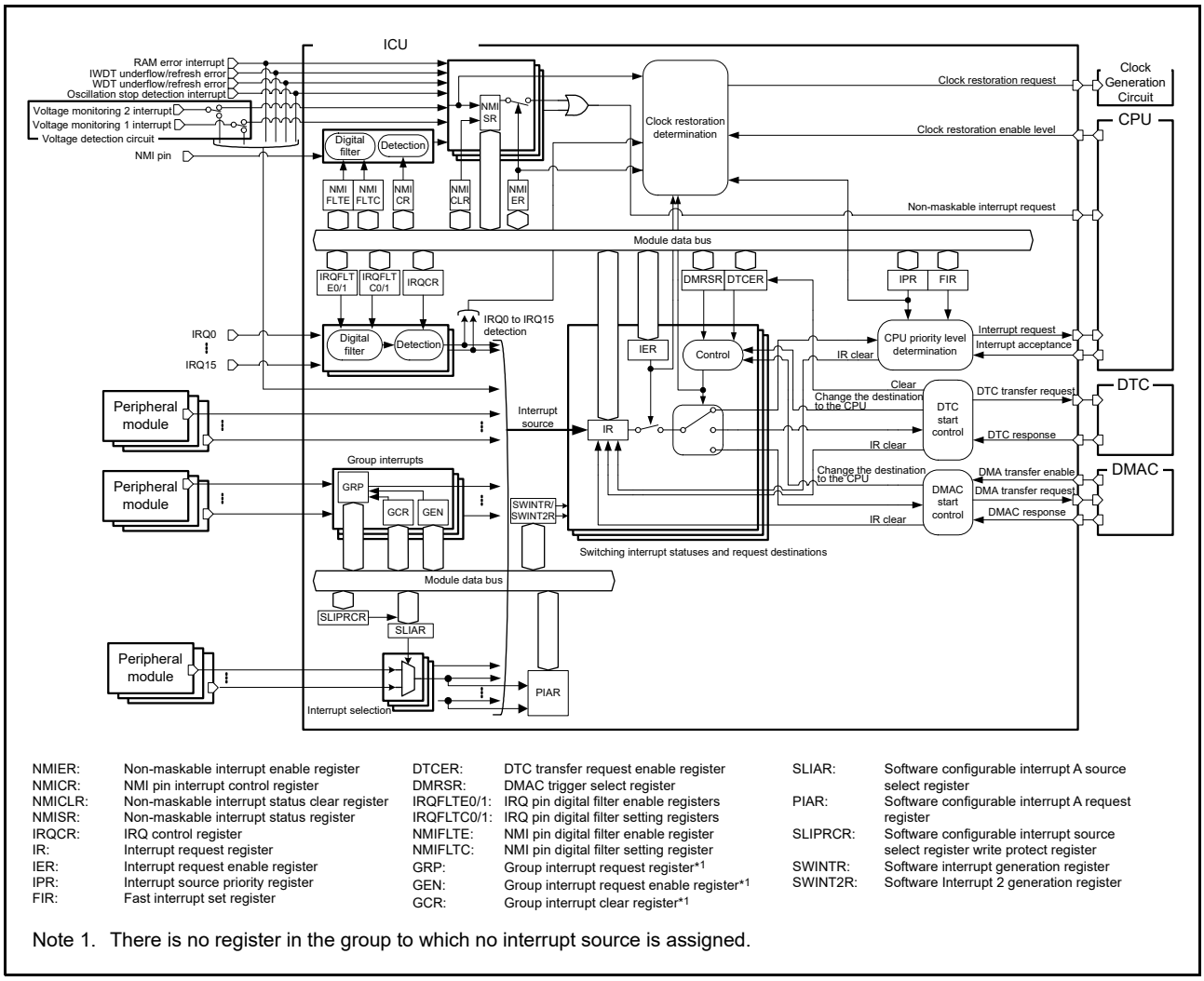
Item		Description

Interrupts	Peripheral interrupts	Interrupts from peripheral modules
		• Interrupt detection method: Edge detection/level detection (fixed for each interrupt source)
		• Group interrupt: Multiple interrupt sources are grouped together and treated as an interrupt source.*1
		  • Group IE0 interrupt: Interrupt sources of coprocessors that use ICLK as the operating clock (edge detection)
		  • Group BE0 interrupt: Interrupt sources of peripheral modules that use PCLKB as the operating clock (edge detection)
		  • Group BL0/BL1/BL2 interrupt: Interrupt sources of peripheral modules that use PCLKB as the operating clock (level detection)
		  • Group AL0/AL1 interrupt: Interrupt sources of peripheral modules that use PCLKA as the operating clock (level detection)
		• Software configurable interrupt A: Any of the interrupt sources for peripheral modules that use PCLKA as the operating clock can be assigned to interrupt vector numbers 208 to 255.

	External pin interrupt	Interrupt by the input signal to the IRQi pin (i = 0 to 15) • Interrupt detection method: Detection of low level, falling edge, rising edge, rising and falling edges One of these detection methods can be set for each source. • Digital filter can be used to remove noise.
	Software interrupt	• Interrupt request can be generated by writing to a register. • Two interrupt sources
	Interrupt priority	Priority level can be set with interrupt source priority register r (IPRr) (r = 000 to 255).
	Fast interrupt function	CPU interrupt response time can be reduced. This function can be used for only one interrupt source.
	DTC/DMAC control	Interrupt sources can be used to start the DTC and DMAC.*2
Non-maskable interrupts *3	NMI pin interrupt	Interrupt by the input signal to the NMI pin • Interrupt detection: Falling edge/rising edge • Digital filter can be used to remove noise.
	Oscillation stop detection interrupt *4	This interrupt occurs when the main clock oscillator stop is detected.
	WDT underflow/refresh error interrupt *4	This interrupt occurs when the watchdog timer (WDT) underflows or a refresh error occurs.
	IWDT underflow/refresh error interrupt *4	This interrupt occurs when the independent watchdog timer (IWDT) underflows or a refresh error occurs.
	Voltage monitoring 1 interrupt *4	Interrupt from voltage detection circuit 1 (LVD1)
	Voltage monitoring 2 interrupt *4	Interrupt from voltage detection circuit 2 (LVD2)
	RAM error interrupt *4	This interrupt occurs when a parity check error is detected in the RAM.
Return from low power consumption states	Sleep mode	• Exit sleep mode by any interrupt source.
	All-module clock stop mode	• Exit all-module clock stop mode by the NMI pin interrupt, external pin interrupt, or peripheral interrupt (voltage monitoring 1, voltage monitoring 2, oscillation stop detection interrupt, IWDT, TMR0 to TMR3).
	Software standby mode	• Exit software standby mode by the NMI pin interrupt, external pin interrupt, or peripheral interrupt (voltage monitoring 1, voltage monitoring 2, IWDT).

Note 1. Groups to which no interrupt source is assigned are reserved. Also, there is no register corresponding to that group.
Note 2. For the DTC and DMAC triggers, refer to Table 14.4, Interrupt Vector Table.
Note 3. Once non-maskable interrupts are enabled, they cannot be disabled.

Note 4. Each source for these non-maskable interrupts can be used for maskable interrupts. When using for maskable interrupts, do not change the NMIER register value from the value after reset. To enable the voltage monitoring 1 interrupt, set the LVD1CR1.LVD1IRQSEL bit to 1, and to enable the voltage monitoring 2 interrupt, set the LVD2CR1.LVD2IRQSEL bit to 1.



NMIER:	Non-maskable interrupt enable register	DTCER:	DTC transfer request enable register	SLIAR:	Software configurable interrupt A source select register
NMICR:	NMI pin interrupt control register	DMRSR:	DMAC trigger select register	PIAR:	Software configurable interrupt A request register
NMICLR:	Non-maskable interrupt status clear register	IRQFLTE0/1:	IRQ pin digital filter enable registers	SLIPRCR:	Software configurable interrupt source select register write protect register
NMISR:	Non-maskable interrupt status register	IRQFLTC0/1:	IRQ pin digital filter setting registers	SWINTR:	Software interrupt generation register
IRQCR:	IRQ control register	NMIFLTE:	NMI pin digital filter enable register	SWINT2R:	Software Interrupt 2 generation register
IR:	Interrupt request register	NMIFLTC:	NMI pin digital filter setting register		
IER:	Interrupt request enable register	GRP:	Group interrupt request register*1		
IPR:	Interrupt source priority register	GEN:	Group interrupt request enable register*1		
FIR:	Fast interrupt set register	GCR:	Group interrupt clear register*1		

Note 1. There is no register in the group to which no interrupt source is assigned.

Figure 14.1 Block Diagram of the ICU

Table 14.2 lists I/O pins used for the ICU.

Table 14.2 ICU I/O Pins

Pin Name	I/O	Description
NMI	Input	Non-maskable interrupt request pin
IRQ0 to IRQ15	Input	External interrupt request pins

14.2 Register Descriptions

14.2.1 Interrupt Request Register n (IRn) (n = 016 to 255)

Address(es): ICU.IR016 0008 7010h to ICU.IR255 0008 70FFh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	IR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IR	Interrupt Status Flag	0: No interrupt request is generated. 1: An interrupt request is generated.	R/(W) *1
b7 to b1	—	Reserved	The read value is 0. The write value should be 0.	R/W

Note 1. For an edge detection interrupt source, only 0 can be written to this bit; do not write 1.
For a level detection interrupt source, neither 0 nor 1 can be written.

The IRn register indicates whether an interrupt request has been generated.
This register is provided for each interrupt vector number, and n matches the interrupt vector number.
For the correspondence between interrupt sources and interrupt vector numbers, refer to Table 14.4, Interrupt Vector Table.

IR Flag (Interrupt Status Flag)

The IR flag is a status flag indicating whether an interrupt request has been generated. This flag becomes 1 when an interrupt request is generated. To detect an interrupt request, set the interrupt enable bit of the peripheral module to enable output of the interrupt request.
An interrupt request can be detected by edge detection or level detection. For interrupts from peripheral modules, the detection method (edge detection or level detection) is determined depending on the source. Refer to Table 14.4, Interrupt Vector Table for details on the detection method for each source. For interrupts from the IRQi pin (i = 0 to 15), edge detection or level detection can be selected by setting the IRQCRi.IRQMD[1:0] bits.
The interrupt status flag for group interrupts is the ISj flag (j = 0 to 31) in the group interrupt request register (GRPBL0, GRPBL1, GRPBL2, GRPAL0, GRPAL1). When any of the ISj flags becomes 1, the IRn.IR flag corresponding to each group interrupt becomes 1. Group interrupts are detected by level detection.
Refer to section 14.4.4, Group Interrupts for details on group interrupts.

(1) Edge detection

This flag becomes 1 under the following condition:

- The IR flag becomes 1 when an interrupt request for peripheral interrupts or external pin interrupts is generated. For interrupt requests for peripheral modules, refer to the corresponding sections.

This flag becomes 0 under any of the following conditions:

- The IR flag becomes 0 when the interrupt request destination accepts an interrupt request.
- The IR flag becomes 0 by writing 0 to the IR flag. Note that when the interrupt request destination is the DTC or DMAC, do not write 0 to the IR flag.

(2) Level detection

This flag becomes 1 under any of the following conditions:

- The IR flag is 1 while an interrupt request for peripheral interrupts or external pin interrupts is generated. For interrupt requests for peripheral modules, refer to the corresponding sections.
- For group interrupts, the IR flag becomes 1 when the ISj flag in the group interrupt request register (GRPBL0, GRPBL1, GRPBL2, GRPAL0, GRPAL1) is 1 (interrupt request is generated) while the ENj bit in the group interrupt request enable register (GENBL0, GENBL1, GENBL2, GENAL0, GENAL1) is 1 (enabled) (j = 0 to 31).

This flag becomes 0 under any of the following conditions:

- The IR flag becomes 0 by clearing output of the peripheral module interrupt request. The IR flag does not become 0 when the interrupt request destination accepts the interrupt request. For interrupt requests for peripheral modules, refer to the corresponding sections.
- For group interrupts, the IR flag becomes 0 when the ENj bit in the group interrupt request enable register is 0 (disabled) or when the ISj flag in the group interrupt request register is 0 (interrupt request is not generated).

When level detection is selected for detecting external pin interrupts, set the input level of the IRQi pin to high (i = 0 to 15) to cancel the external pin interrupt that has occurred. When level detection is selected, do not write to the IR flag.

14.2.2 Interrupt Request Enable Register m (IERm) (m = 02h to 1Fh)

Address(es): ICU.IER02 0008 7202h to ICU.IER1F 0008 721Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IEN0	Interrupt Request Enable 0	0: Interrupt request is disabled. 1: Interrupt request is enabled.	R/W
b1	IEN1	Interrupt Request Enable 1		R/W
b2	IEN2	Interrupt Request Enable 2		R/W
b3	IEN3	Interrupt Request Enable 3		R/W
b4	IEN4	Interrupt Request Enable 4		R/W
b5	IEN5	Interrupt Request Enable 5		R/W
b6	IEN6	Interrupt Request Enable 6		R/W
b7	IEN7	Interrupt Request Enable 7		R/W

Note: When the interrupt source of the interrupt vector number is reserved, set the corresponding bit to 0. The read value is 0.

The IERm register enables or disables output of the interrupt request to the interrupt request destination.

IENj Bit (Interrupt Request Enable j) (j = 0 to 7)

When the IENj bit is 1, an interrupt request is output to the destination. When the IENj bit is 0, an interrupt request is not output to the destination.

The IRn.IR flag (n = 016 to 255) is not affected by the IENj bit setting. Even when the IENj bit is 0, the IR flag changes according to the conditions described in section 14.2.1, Interrupt Request Register n (IRn) (n = 016 to 255).

The IERm.IENj bit is provided for each interrupt vector number.

Refer to Table 14.4, Interrupt Vector Table for the correspondence between interrupt sources and the IERm.IENj bit.

Note that m and j can be calculated by the following formula:

m = quotient when n divided by 8

j = remainder when n divided by 8

Refer to section 14.7.3.1, Interrupt Request Destination Setting Procedure for the procedure to set the IERm.IENj bit to select the interrupt request destination.

14.2.3 Interrupt Source Priority Register r (IPRr) (r = 000 to 255)

Address(es): ICU.IPR000 0008 7300h to ICU.IPR255 0008 73FFh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	IPR[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	IPR[3:0]	Interrupt Priority Level Select	b3 b0 0 0 0 0: Level 0 (interrupt disabled) *1 0 0 0 1: Level 1 0 0 1 0: Level 2 0 0 1 1: Level 3 0 1 0 0: Level 4 0 1 0 1: Level 5 0 1 1 0: Level 6 0 1 1 1: Level 7 1 0 0 0: Level 8 1 0 0 1: Level 9 1 0 1 0: Level 10 1 0 1 1: Level 11 1 1 0 0: Level 12 1 1 0 1: Level 13 1 1 1 0: Level 14 1 1 1 1: Level 15 (highest)	R/W
b7 to b4	—	Reserved	The read value is 0. The write value should be 0.	R/W

Note 1. For an interrupt source of a fast interrupt, even when the IPR[3:0] bits are set to level 0, the priority level is level 15.

The IPRr register sets the interrupt priority level of an interrupt source that is assigned to the corresponding interrupt vector number.

IPR[3:0] Bits (Interrupt Priority Level Select)

The IPR[3:0] bits select the interrupt priority level of the corresponding interrupt source.
The priority level selected by the IPR[3:0] bits is used for only determining the priority level of interrupt requests to the CPU. It does not affect transfer requests to the DTC and DMAC.
The CPU accepts only interrupt requests that have the higher priority level than the processor interrupt priority level indicated by the PSW.IPL[3:0] bits.
When multiple interrupt requests are concurrently generated, the priority levels selected by the corresponding IPR[3:0] bits are compared. When multiple interrupt requests that have the same priority level are concurrently generated, the interrupt request that has the smallest interrupt vector number has priority.
Write to this register while the corresponding IERm.IENj bit is 0 (interrupt request is disabled) (m = 02h to 1Fh; j = 0 to 7).
Refer to Table 14.4, Interrupt Vector Table for the correspondence between interrupt vectors and the IPRr register. Note that r matches the vector number when the interrupt vector number is 32 or greater.

14.2.4 Fast Interrupt Set Register (FIR)

Address(es): ICU.FIR 0008 72F0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FIEN	—	—	—	—	—	—	—	FVCT[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	FVCT[7:0]	Fast Interrupt Vector Number	Set the vector number of an interrupt source that is assigned to a fast interrupt.	R/W
b14 to b8	—	Reserved	The read value is 0. The write value should be 0.	R/W
b15	FIEN	Fast Interrupt Enable	0: Fast interrupt is disabled 1: Fast interrupt is enabled	R/W

The FIR register sets an interrupt source that is handled as the fast interrupt.
The fast interrupt is enabled only when the destination is the CPU. When the destination is the DTC or DMAC, the DTC or DMA transfer request is not affected by setting the interrupt vector number as the fast interrupt.
Write to this register while the corresponding IERm.IENj bit is 0 (m = 02h to 1Fh; j = 0 to 7).
Refer to section 14.9, Fast interrupt for details on the fast interrupt.

FVCT[7:0] Bits (Fast Interrupt Vector Number)

The FVCT[7:0] bits set the interrupt vector number of an interrupt source for the fast interrupt.
Refer to Table 14.4, Interrupt Vector Table for interrupt vector numbers that can be set in the FVCT[7:0] bits. Do not set an interrupt vector number that is reserved.

FIEN Bit (Fast Interrupt Enable)

The FIEN bit enables the fast interrupt to be used.
When the FIEN bit is 1, the interrupt source that is assigned to the interrupt vector number set by the FVCT[7:0] bits is handled as the fast interrupt.
When an interrupt request of the interrupt vector number set by the FVCT[7:0] bits is generated to the CPU while the FIEN bit is 1, an interrupt request is output to the CPU as the fast interrupt regardless of the IPRr register setting (r = 000 to 255). Note that the IPRr register setting is required when using the fast interrupt to exit software standby mode. Refer to section 14.10.3, Exiting Software Standby Mode for details.

14.2.5 Software Interrupt Generation Register (SWINTR)

Address(es): ICU.SWINTR 0008 72E0h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SWINT
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SWINT	Software Interrupt Generation	The read value is 0. When writing 1 to this bit, a software interrupt request is generated. Writing 0 to this bit has no effect.	R/W
b7 to b1	—	Reserved	The read value is 0. The write value should be 0.	R/W

The SWINTR register controls generation of a software interrupt request.

SWINT Bit (Software Interrupt Generation)

When the SWINT bit is set to 1, a software interrupt request (SWINT) is generated, and the IR027.IR flag becomes 1. A software interrupt request (SWINT) can be set as a DTC trigger, but it cannot be set as a DMAC trigger.

14.2.6 Software Interrupt 2 Generation Register (SWINT2R)

Address(es): ICU.SWINT2R 0008 72E1h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SWINT2
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SWINT2	Software Interrupt 2 Generation	The read value is 0. When writing 1 to this bit, a software interrupt request 2 is generated. Writing 0 to this bit has no effect.	R/W
b7 to b1	—	Reserved	The read value is 0. The write value should be 0.	R/W

The SWINT2R register controls generation of software interrupt request 2.

SWINT2 Bit (Software Interrupt 2 Generation)

When the SWINT2 bit is set to 1, software interrupt request 2 (SWINT2) is generated, and the IR026.IR flag becomes 1. Software interrupt request 2 (SWINT2) can be set as the DTC trigger, but it cannot be set as the DMAC trigger.

14.2.7 DTC Transfer Request Enable Register n (DTCERn) (n = 026 to 255)

Address(es): ICU.DTCER026 0008 711Ah to ICU.DTCER255 0008 71FFh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DTCE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DTCE	DTC Transfer Request Enable	0: The corresponding interrupt source is selected as an interrupt request to the CPU or as the DMAC trigger. 1: The corresponding interrupt source is selected as the DTC trigger.	R/W
b7 to b1	—	Reserved	The read value is 0. The write value should be 0.	R/W

The DTCERn register selects the interrupt source corresponding to interrupt vector number n as the DTC trigger.
Do not set the same interrupt source for both the DTC trigger and DMAC trigger. Refer to Table 14.4, Interrupt Vector Table for the correspondence between interrupt sources and interrupt vector numbers and interrupt sources that can be used as the DTC trigger.

DTCE Bit (DTC Transfer Request Enable)

When the DTCE bit is set to 1, the corresponding interrupt source is selected as the DTC trigger.
This bit becomes 1 under the following condition:

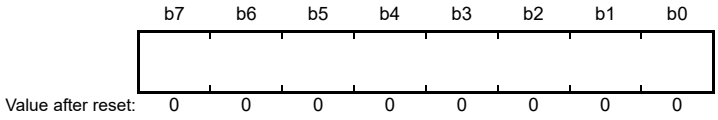
- 1 is written to the DTCE bit

This bit becomes 0 under any of the following conditions:

- The specified number of transfers is completed (for the chain transfer, the number of transfers for the last chain transfer is completed)
- 0 is written to the DTCE bit

14.2.8 DMAC Trigger Select Register m (DMRSRm) (m = DMAC channel number)

Address(es): ICU.DMRSR0 0008 7400h, ICU.DMRSR1 0008 7404h, ICU.DMRSR2 0008 7408h, ICU.DMRSR3 0008 740Ch, ICU.DMRSR4 0008 7410h, ICU.DMRSR5 0008 7414h, ICU.DMRSR6 0008 7418h, ICU.DMRSR7 0008 741Ch



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	—	These bits set the interrupt vector number of the interrupt source as the DMAC trigger.	R/W

The DMRSRm register sets an interrupt source as the DMACm trigger.
Do not set the same vector number for multiple DMRSRm registers. Do not set the same interrupt source for both the DTC trigger and DMAC trigger. Otherwise, the operation is not guaranteed.
Set the interrupt vector number of an interrupt source used as the DMAC trigger in the DMRSRm register. Do not set vector numbers of interrupt sources that cannot be used as the DMAC trigger.
Refer to Table 14.4, Interrupt Vector Table for interrupt vector numbers of interrupt sources.
Write the DMRSRm register while the DMACm.DMCNT.DTE bit is 0.

14.2.9 IRQ Control Register i (IRQCRi) (i = 0 to 15)

Address(es): ICU.IRQCR0 0008 7500h to ICU.IRQCR15 0008 750Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	IRQMD[1:0]		—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	The read value is 0. The write value should be 0.	R/W
b3, b2	IRQMD[1:0]	IRQ Detection Select	b3 b2 0 0: Low level 0 1: Falling edge 1 0: Rising edge 1 1: Rising and falling edges	R/W
b7 to b4	—	Reserved	The read value is 0. The write value should be 0.	R/W

The IRQCRi register selects the detection method for external pin interrupts.
Write to this register while the corresponding IERm.IENj bit is 0 (m = 02h to 1Fh; j = 0 to 7). After writing to this register, set the IRn.IR flag to 0, and then set the IENj bit to 1 (n = 016 to 255). Note that setting the IR flag to 0 is not required when changing the detection method to level detection.

IRQMD[1:0] Bits (IRQ Detection Select)

The IRQMD[1:0] bits set the detection method for the IRQi pin interrupt (i = 0 to 15).
Refer to section 14.7.4, Setting the External Pin Interrupt for the procedure to set the external pin interrupts.

14.2.10 IRQ Pin Digital Filter Enable Register 0 (IRQFLTE0)

Address(es): ICU.IRQFLTE0 0008 7520h

	b7	b6	b5	b4	b3	b2	b1	b0
	FLTEN7	FLTEN6	FLTEN5	FLTEN4	FLTEN3	FLTEN2	FLTEN1	FLTEN0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	FLTEN0	IRQ0 Digital Filter Enable	0: Digital filter is disabled. 1: Digital filter is enabled.	R/W
b1	FLTEN1	IRQ1 Digital Filter Enable		R/W
b2	FLTEN2	IRQ2 Digital Filter Enable		R/W
b3	FLTEN3	IRQ3 Digital Filter Enable		R/W
b4	FLTEN4	IRQ4 Digital Filter Enable		R/W
b5	FLTEN5	IRQ5 Digital Filter Enable		R/W
b6	FLTEN6	IRQ6 Digital Filter Enable		R/W
b7	FLTEN7	IRQ7 Digital Filter Enable		R/W

The IRQFLTE0 register enables and disables digital filters for pins IRQ0 to IRQ7.

FLTENi Bits (IRQi Digital Filter Enable) (i = 0 to 7)

When the FLTENi bit is 1, the digital filter for the IRQi pin is enabled. When the FLTENi bit is 0, the digital filter for the IRQi pin is disabled.

The signal input to the IRQi pin is sampled at the sampling clock set by the IRQFLTC0.FCLKSELi[1:0] bits. When the sampled signal level does not match three times in a row, the input signal is removed.

Refer to section 14.7.6, Digital Filter for details on the digital filter.

14.2.11 IRQ Pin Digital Filter Enable Register 1 (IRQFLTE1)

Address(es): ICU.IRQFLTE1 0008 7521h

	b7	b6	b5	b4	b3	b2	b1	b0
	FLTEN15	FLTEN14	FLTEN13	FLTEN12	FLTEN11	FLTEN10	FLTEN9	FLTEN8
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	FLTEN8	IRQ8 Digital Filter Enable	0: Digital filter is disabled. 1: Digital filter is enabled.	R/W
b1	FLTEN9	IRQ9 Digital Filter Enable		R/W
b2	FLTEN10	IRQ10 Digital Filter Enable		R/W
b3	FLTEN11	IRQ11 Digital Filter Enable		R/W
b4	FLTEN12	IRQ12 Digital Filter Enable		R/W
b5	FLTEN13	IRQ13 Digital Filter Enable		R/W
b6	FLTEN14	IRQ14 Digital Filter Enable		R/W
b7	FLTEN15	IRQ15 Digital Filter Enable		R/W

The IRQFLTE1 register enables or disables digital filters for pins IRQ8 to IRQ15.

FLTENi Bit (IRQi Digital Filter Enable) (i = 8 to 15)

When the FLTENi bit is 1, the digital filter for the IRQi pin is enabled. When the FLTENi bit is 0, the digital filter for the IRQi pin is disabled.

The signal input to the IRQi pin is sampled at the sampling clock set by the IRQFLTC1.FCLKSELi[1:0] bits. When the sampled signal level does not match three times in a row, the input signal is removed.

Refer to section 14.7.6, Digital Filter for details on the digital filter.

14.2.12 IRQ Pin Digital Filter Setting Register 0 (IRQFLTC0)

Address(es): ICU.IRQFLTC0 0008 7528h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FCLKSEL7[1:0]		FCLKSEL6[1:0]		FCLKSEL5[1:0]		FCLKSEL4[1:0]		FCLKSEL3[1:0]		FCLKSEL2[1:0]		FCLKSEL1[1:0]		FCLKSEL0[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	FCLKSEL0[1:0]	IRQ0 Digital Filter Sampling Clock	0 0: PCLKB 0 1: PCLKB/8 1 0: PCLKB/32 1 1: PCLKB/64	R/W
b3, b2	FCLKSEL1[1:0]	IRQ1 Digital Filter Sampling Clock		R/W
b5, b4	FCLKSEL2[1:0]	IRQ2 Digital Filter Sampling Clock		R/W
b7, b6	FCLKSEL3[1:0]	IRQ3 Digital Filter Sampling Clock		R/W
b9, b8	FCLKSEL4[1:0]	IRQ4 Digital Filter Sampling Clock		R/W
b11, b10	FCLKSEL5[1:0]	IRQ5 Digital Filter Sampling Clock		R/W
b13, b12	FCLKSEL6[1:0]	IRQ6 Digital Filter Sampling Clock		R/W
b15, b14	FCLKSEL7[1:0]	IRQ7 Digital Filter Sampling Clock		R/W

The IRQFLTC0 register sets the sampling clock of the digital filter for pins IRQ0 to IRQ7.

FCLKSELi[1:0] Bits (IRQi Digital Filter Sampling Clock) (i = 0 to 7)

The FCLKSELi[1:0] bits set the sampling clock of the digital filter for the IRQi pin.

Refer to section 14.7.6, Digital Filter for details on the digital filter.

14.2.13 IRQ Pin Digital Filter Setting Register 1 (IRQFLTC1)

Address(es): ICU.IRQFLTC1 0008 752Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FCLKSEL15[1:0]		FCLKSEL14[1:0]		FCLKSEL13[1:0]		FCLKSEL12[1:0]		FCLKSEL11[1:0]		FCLKSEL10[1:0]		FCLKSEL9[1:0]		FCLKSEL8[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	FCLKSEL8[1:0]	IRQ8 Digital Filter Sampling Clock	0 0: PCLKB 0 1: PCLKB/8 1 0: PCLKB/32 1 1: PCLKB/64	R/W
b3, b2	FCLKSEL9[1:0]	IRQ9 Digital Filter Sampling Clock		R/W
b5, b4	FCLKSEL10[1:0]	IRQ10 Digital Filter Sampling Clock		R/W
b7, b6	FCLKSEL11[1:0]	IRQ11 Digital Filter Sampling Clock		R/W
b9, b8	FCLKSEL12[1:0]	IRQ12 Digital Filter Sampling Clock		R/W
b11, b10	FCLKSEL13[1:0]	IRQ13 Digital Filter Sampling Clock		R/W
b13, b12	FCLKSEL14[1:0]	IRQ14 Digital Filter Sampling Clock		R/W
b15, b14	FCLKSEL15[1:0]	IRQ15 Digital Filter Sampling Clock		R/W

The IRQFLTC1 register sets the sampling clock of the digital filter for pins IRQ8 to IRQ15.

FCLKSELi[1:0] Bits (IRQi Digital Filter Sampling Clock) (i = 8 to 15)

The FCLKSELi[1:0] bits set the sampling clock of the digital filter for the IRQi pin.

Refer to section 14.7.6, Digital Filter for details on the digital filter.

14.2.14 Non-Maskable Interrupt Status Register (NMISR)

Address(es): ICU.NMISR 0008 7580h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	RAMST	LVD2ST	LVD1ST	IWDTST	WDTST	OSTST	NMIST
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	NMIST	NMI Status Flag	0: NMI pin interrupt is not requested. 1: NMI pin interrupt is requested.	R
b1	OSTST	Oscillation Stop Detection Interrupt Status Flag	0: Oscillation stop detection interrupt is not requested. 1: Oscillation stop detection interrupt is requested.	R
b2	WDTST	WDT Underflow/Refresh Error Status Flag	0: WDT underflow/refresh error interrupt is not requested. 1: WDT underflow/refresh error interrupt is requested.	R
b3	IWDTST	IWDT Underflow/Refresh Error Status Flag	0: IWDT underflow/refresh error interrupt is not requested. 1: IWDT underflow/refresh error interrupt is requested.	R
b4	LVD1ST	Voltage Monitoring 1 Interrupt Status Flag	0: Voltage monitoring 1 interrupt is not requested. 1: Voltage monitoring 1 interrupt is requested.	R
b5	LVD2ST	Voltage Monitoring 2 Interrupt Status Flag	0: Voltage monitoring 2 interrupt is not requested. 1: Voltage monitoring 2 interrupt is requested.	R
b6	RAMST	RAM Error Interrupt Status Flag	0: RAM error interrupt is not requested. 1: RAM error interrupt is requested.	R
b7	—	Reserved	This bit is read as 0 and cannot be modified.	R

The NMISR register indicates whether a non-maskable interrupt request has occurred.
Each flag in the NMISR register is not affected by the setting of the corresponding bit in the NMIER register.
In the non-maskable interrupt handler, read the NMISR register to check if the other non-maskable interrupt has occurred. Confirm that all the status flags are 0 before exiting the interrupt handler.

NMIST Flag (NMI Status Flag)

The NMIST flag indicates whether an NMI pin interrupt request has been generated.
This flag is read-only. To set the NMIST flag to 0, set the NMICLR.NMICLR bit to 1.
This flag becomes 1 under the following condition:

- An edge set in the NMICR.NMIMD bit is input to the NMI pin

This flag becomes 0 under the following condition:

- 1 is written to the NMICLR.NMICLR bit

OSTST Flag (Oscillation Stop Detection Interrupt Status Flag)

The OSTST flag indicates whether an oscillation stop detection interrupt request has been generated.
The OSTST flag is read-only. To set the OSTST flag to 0, set the NMICLR.OSTCLR bit to 1.
This flag becomes 1 under the following condition:

- An oscillation stop detection interrupt occurs

This flag becomes 0 under the following condition:

- 1 is written to the NMICLR.OSTCLR bit

WDTST Flag (WDT Underflow/Refresh Error Status Flag)

The WDTST flag indicates whether an WDT underflow/refresh error interrupt request is generated.
The WDTST flag is read-only. To set the WDTST flag to 0, set the NMICLR.WDTCLR bit to 1.
This flag becomes 1 under the following condition:

- A WDT underflow/refresh error interrupt occurs while the WDTRCR.RSTIRQS bit is 0

This flag becomes 0 under the following condition:

- 1 is written to the NMICLR.WDTCLR bit

IWDTST Flag (IWDT Underflow/Refresh Error Status Flag)

The IWDTST flag indicates whether an IWDT underflow/refresh error interrupt request is generated.
The IWDTST flag is read-only. To set the IWDTST flag to 0, set the NMICLR.IWDTCLR bit to 1.
This flag becomes 1 under the following condition:

- An IWDT underflow/refresh error interrupt occurs while the IWDTRCR.RSTIRQS bit is 0

This flag becomes 0 under the following condition:

- 1 is written to the NMICLR.IWDTCLR bit

LVD1ST Flag (Voltage Monitoring 1 Interrupt Status Flag)

The LVD1ST flag indicates whether a voltage monitoring 1 interrupt request is generated.
The LVD1ST flag is read-only. To set the LVD1ST flag to 0, set the NMICLR.LVD1CLR bit to 1.
This flag becomes 1 under the following condition:

- A voltage monitoring 1 interrupt occurs while the LVD1CR1.LVD1IRQSEL bit is 0

This flag becomes 0 under the following condition:

- 1 is written to the NMICLR.LVD1CLR bit

LVD2ST Flag (Voltage Monitoring 2 Interrupt Status Flag)

The LVD2ST flag indicates whether a voltage monitoring 2 interrupt request is generated.
The LVD2ST flag is read-only. To set the LVD2ST flag to 0, set the NMICLR.LVD2CLR bit to 1.
This flag becomes 1 under the following condition:

- A voltage monitoring 2 interrupt occurs while the LVD2CR1.LVD2IRQSEL bit is 0

This flag becomes 0 under the following condition:

- 1 is written to the NMICLR.LVD2CLR bit

RAMST Flag (RAM Error Interrupt Status Flag)

The RAMST flag indicates whether an RAM error interrupt request is generated from the RAM.
The RAMST flag is read-only. To set the RAMST flag to 0, clear all error status flags of the RAM. Refer to section 47.3.2, RAM Error Interrupt Function for details.
This flag becomes 1 under the following condition:

- A parity check error interrupt occurs (When the RAM.RAMSTS.RAMERR flag becomes 1)

This flag becomes 0 under the following condition:

- When all sources which set the RAMST flag to 1 are cleared.

14.2.15 Non-Maskable Interrupt Enable Register (NMIER)

Address(es): ICU.NMIER 0008 7581h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	RAMEN	LVD2EN	LVD1EN	IWDTEN	WDTEN	OSTEN	NMIEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	NMIEN	NMI Pin Interrupt Enable	0: NMI pin interrupt is disabled. 1: NMI pin interrupt is enabled.	R/(W) *1
b1	OSTEN	Oscillation Stop Detection Interrupt Enable	0: Oscillation stop detection interrupt is disabled. 1: Oscillation stop detection interrupt is enabled.	R/(W) *1
b2	WDTEN	WDT Underflow/Refresh Error Enable	0: WDT underflow/refresh error interrupt is disabled. 1: WDT underflow/refresh error interrupt is enabled.	R/(W) *1
b3	IWDTEN	IWDT Underflow/Refresh Error Enable	0: IWDT underflow/refresh error interrupt is disabled. 1: IWDT underflow/refresh error interrupt is enabled.	R/(W) *1
b4	LVD1EN	Voltage Monitoring 1 Interrupt Enable	0: Voltage monitoring 1 interrupt is disabled. 1: Voltage monitoring 1 interrupt is enabled.	R/(W) *1
b5	LVD2EN	Voltage Monitoring 2 Interrupt Enable	0: Voltage monitoring 2 interrupt is disabled. 1: Voltage monitoring 2 interrupt is enabled.	R/(W) *1
b6	RAMEN	RAM Error Interrupt Enable	0: RAM error interrupt is disabled. 1: RAM error interrupt is enabled.	R/(W) *1
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. Once this bit is set to 1, it cannot be set to 0 by software.

The NMIER register enables and disables generation of non-maskable interrupt requests. When each bit is 1, the corresponding interrupt source is used as a non-maskable interrupt.

NMIEN Bit (NMI Pin Interrupt Enable)

The NMIEN bit enables and disables using the NMI pin interrupt.

OSTEN Bit (Oscillation Stop Detection Interrupt Enable)

The OSTEN bit enables and disables generation of a non-maskable interrupt by the oscillation stop detection interrupt. When the oscillation stop detection interrupt is used as a maskable interrupt, leave this bit set to 0.

WDTEN Bit (WDT Underflow/Refresh Error Enable)

The WDTEN bit enables and disables generation of a non-maskable interrupt by the WDT underflow/refresh error interrupt.
When the WDT underflow/refresh error interrupt is used as a maskable interrupt, leave this bit set to 0.

IWDTEN Bit (IWDT Underflow/Refresh Error Enable)

The IWDTEN bit enables and disables generation of a non-maskable interrupt by the IWDT underflow/refresh error interrupt.
When the IWDT underflow/refresh error interrupt is used as a maskable interrupt, leave this bit set to 0.

LVD1EN Bit (Voltage Monitoring 1 Interrupt Enable)

The LVD1EN bit enables and disables generation of a non-maskable interrupt by the voltage monitoring 1 interrupt.
When the voltage monitoring 1 interrupt is used as a maskable interrupt, leave this bit set to 0.

LVD2EN Bit (Voltage Monitoring 2 Interrupt Enable)

The LVD2EN bit enables and disables generation of a non-maskable interrupt by the voltage monitoring 2 interrupt. When the voltage monitoring 2 interrupt is used as a maskable interrupt, leave this bit set to 0.

RAMEN Bit (RAM Error Interrupt Enable)

The RAMEN bit enables and disables generation of a non-maskable interrupt by the RAM error interrupt. When the RAM error interrupt is used as a maskable interrupt, leave this bit set to 0.

14.2.16 Non-Maskable Interrupt Status Clear Register (NMICLR)

Address(es): ICU.NMICLR 0008 7582h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	LVD2CLR	LVD1CLR	IWDTCLR	WDTCLR	OSTCLR	NMICLR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	NMICLR	NMI Clear	The read value is 0. When 1 is written to this bit, the NMISR.NMIST flag becomes 0. Writing 0 to this bit has no effect.	R/(W)
b1	OSTCLR	OST Clear	The read value is 0. When 1 is written to this bit, the NMISR.OSTST flag becomes 0. Writing 0 to this bit has no effect.	R/(W)
b2	WDTCLR	WDT Clear	The read value is 0. When 1 is written to this bit, the NMISR.WDTST flag becomes 0. Writing 0 to this bit has no effect.	R/(W)
b3	IWDTCLR	IWDT Clear	The read value is 0. When 1 is written to this bit, the NMISR.IWDTST flag becomes 0. Writing 0 to this bit has no effect.	R/(W)
b4	LVD1CLR	LVD1 Clear	The read value is 0. When 1 is written to this bit, the NMISR.LVD1ST flag becomes 0. Writing 0 to this bit has no effect.	R/(W)
b5	LVD2CLR	LVD2 Clear	The read value is 0. When 1 is written to this bit, the NMISR.LVD2ST flag becomes 0. Writing 0 to this bit has no effect.	R/(W)
b7-b6	—	Reserved	The read value is 0. The write value should be 0.	R/W

The NMICLR register clears each flag in the NMISR register.
When writing 1 to a bit in this register, the corresponding status flag becomes 0.

14.2.17 NMI Pin Interrupt Control Register (NMICR)

Address(es): ICU.NMICR 0008 7583h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	NMIMD	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	The read value is 0. The write value should be 0.	R/W
b3	NMIMD	NMI Detection Set	0: Falling edge 1: Rising edge	R/W
b7 to b4	—	Reserved	The read value is 0. The write value should be 0.	R/W

The NMICR register selects a detection method for the NMI pin interrupt.
Write to the NMICR register while the NMIER.NMIEN bit is 0.

14.2.18 NMI Pin Digital Filter Enable Register (NMIFLTE)

Address(es): ICU.NMIFLTE 0008 7590h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	NFLTEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	NFLTEN	NMI Digital Filter Enable	0: Digital filter is disabled. 1: Digital filter is enabled.	R/W
b7 to b1	—	Reserved	The read value is 0. The write value should be 0.	R/W

The NMIFLTE register enables or disables the digital filter for the NMI pin.

NFLTEN Bit (NMI Digital Filter Enable)

When the NFLTEN bit is 1, the digital filter is enabled. When the NFLTEN bit is 0, the digital filter is disabled.
The signal input to the NMI pin is sampled at the sampling clock set by the NMIFLTC.NFCLKSEL[1:0] bits. When the sampled signal level does not match three times in a row, the input signal is removed.
Refer to section 14.7.6, Digital Filter for details on the digital filter.

14.2.19 NMI Pin Digital Filter Setting Register (NMIFLTC)

Address(es): ICU.NMIFLTC 0008 7594h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	NFCLKSEL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	NFCLKSEL[1:0]	NMI Digital Filter Sampling Clock	b1 b0 0 0: PCLKB 0 1: PCLKB/8 1 0: PCLKB/32 1 1: PCLKB/64	R/W
b7 to b2	—	Reserved	The read value is 0. The write value should be 0.	R/W

The NMIFLTC register sets the sampling clock of the digital filter for the NMI pin.

NFCLKSEL[1:0] Bits (NMI Digital Filter Sampling Clock)

The NFCLKSEL[1:0] bits select the sampling clock of the digital filter for the NMI pin.
Refer to section 14.7.6, Digital Filter for details on the digital filter.

14.2.20 Group BL0/BL1/BL2 Interrupt Request Register (GRPBL0/GRPBL1/GRPBL2), Group AL0/AL1 Interrupt Request Register (GRPAL0/GRPAL1)

Address(es): ICU.GRPBL0 0008 7630h, ICU.GRPBL1 0008 7634h, ICU.GRPBL2 0008 7638h, ICU.GRPAL0 0008 7830h, ICU.GRPAL1 0008 7834h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IS31	IS30	IS29	IS28	IS27	IS26	IS25	IS24	IS23	IS22	IS21	IS20	IS19	IS18	IS17	IS16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	IS15	IS14	IS13	IS12	IS11	IS10	IS9	IS8	IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IS0	Interrupt Status Flag 0	0: Interrupt is not requested. 1: Interrupt is requested.	R
b1	IS1	Interrupt Status Flag 1		R
b2	IS2	Interrupt Status Flag 2		R
b3	IS3	Interrupt Status Flag 3		R
b4	IS4	Interrupt Status Flag 4		R
b5	IS5	Interrupt Status Flag 5		R
b6	IS6	Interrupt Status Flag 6		R
b7	IS7	Interrupt Status Flag 7		R
b8	IS8	Interrupt Status Flag 8		R
b9	IS9	Interrupt Status Flag 9		R
b10	IS10	Interrupt Status Flag 10		R
b11	IS11	Interrupt Status Flag 11		R
b12	IS12	Interrupt Status Flag 12		R
b13	IS13	Interrupt Status Flag 13		R
b14	IS14	Interrupt Status Flag 14		R
b15	IS15	Interrupt Status Flag 15		R
b16	IS16	Interrupt Status Flag 16		R
b17	IS17	Interrupt Status Flag 17		R
b18	IS18	Interrupt Status Flag 18		R
b19	IS19	Interrupt Status Flag 19		R
b20	IS20	Interrupt Status Flag 20		R
b21	IS21	Interrupt Status Flag 21		R
b22	IS22	Interrupt Status Flag 22		R
b23	IS23	Interrupt Status Flag 23		R
b24	IS24	Interrupt Status Flag 24		R
b25	IS25	Interrupt Status Flag 25		R
b26	IS26	Interrupt Status Flag 26		R
b27	IS27	Interrupt Status Flag 27		R
b28	IS28	Interrupt Status Flag 28		R
b29	IS29	Interrupt Status Flag 29		R
b30	IS30	Interrupt Status Flag 30		R
b31	IS31	Interrupt Status Flag 31		R

These registers indicate the status of each interrupt request of group interrupt sources.

Registers GRPBL0, GRPBL1, and GRPBL2 contain the statuses of interrupt sources that are detected by level detection

and use PCLKB as the operating clock.
The GRPAL0 and GRPAL1 registers contains the statuses of interrupt sources that are detected by level detection and use PCLKA as the operating clock.
These registers are collectively referred to as the “group interrupt request register”.
Refer to section 14.4.4, Group Interrupts for details on group interrupts.

ISj Flag (Interrupt Status Flag j) (j = 0 to 31)

The ISj flag indicates the status of interrupt sources assigned to group interrupts.
The ISj flag becomes 1 only when the corresponding ENj bit in the group interrupt request enable register is 1. When any of the ISj flags becomes 1, the IRn.IR flag corresponding to the group interrupt becomes 1 (n = 016 to 255).

(1) Group BL0/BL1/BL2

This flag becomes 1 under the following condition:

- The GRPBL0/GRPBL1/GRPBL2.ISj flag becomes 1 while the corresponding peripheral module interrupt request is generated when the GENBL0/GENBL1/GENBL2.ENj bit is 1.

This flag becomes 0 under any of the following conditions:

- The GRPBL0/GRPBL1/GRPBL2.ISj flag becomes 0 when the corresponding peripheral module interrupt request is cleared.
- The GRPBL0/GRPBL1/GRPBL2.ISj flag becomes 0 when the GENBL0/GENBL1/GENBL2.ENj bit is set to 0.

(2) Group AL0/AL1

This flag becomes 1 under the following condition:

- The GRPAL0/GRPAL1.ISj flag becomes 1 while the corresponding peripheral module interrupt request is generated when the GENAL0/GENAL1.ENj bit is 1.

This flag becomes 0 under any of the following conditions:

- The GRPAL0/GRPAL1.ISj flag becomes 0 when the corresponding peripheral module interrupt request is cleared.
- The GRPAL0/GRPAL1.ISj flag becomes 0 when the GENAL0/GENAL1.ENj bit is set to 0.

14.2.21 Group BL0/BL1/BL2 Interrupt Request Enable Register (GENBL0/GENBL1/GENBL2), Group AL0/AL1 Interrupt Request Enable Register (GENAL0/GENAL1)

Address(es): ICU.GENBL0 0008 7670h, ICU.GENBL1 0008 7674h, ICU.GENBL2 0008 7678h, ICU.GENAL0 0008 7870h, ICU.GENAL1 0008 7874h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	EN31	EN30	EN29	EN28	EN27	EN26	EN25	EN24	EN23	EN22	EN21	EN20	EN19	EN18	EN17	EN16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	EN15	EN14	EN13	EN12	EN11	EN10	EN9	EN8	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	EN0	Interrupt Request Enable 0	0: Interrupt request is disabled. 1: Interrupt request is enabled.	R/W
b1	EN1	Interrupt Request Enable 1		R/W
b2	EN2	Interrupt Request Enable 2		R/W
b3	EN3	Interrupt Request Enable 3		R/W
b4	EN4	Interrupt Request Enable 4		R/W
b5	EN5	Interrupt Request Enable 5		R/W
b6	EN6	Interrupt Request Enable 6		R/W
b7	EN7	Interrupt Request Enable 7		R/W
b8	EN8	Interrupt Request Enable 8		R/W
b9	EN9	Interrupt Request Enable 9		R/W
b10	EN10	Interrupt Request Enable 10		R/W
b11	EN11	Interrupt Request Enable 11		R/W
b12	EN12	Interrupt Request Enable 12		R/W
b13	EN13	Interrupt Request Enable 13		R/W
b14	EN14	Interrupt Request Enable 14		R/W
b15	EN15	Interrupt Request Enable 15		R/W
b16	EN16	Interrupt Request Enable 16		R/W
b17	EN17	Interrupt Request Enable 17		R/W
b18	EN18	Interrupt Request Enable 18		R/W
b19	EN19	Interrupt Request Enable 19		R/W
b20	EN20	Interrupt Request Enable 20		R/W
b21	EN21	Interrupt Request Enable 21		R/W
b22	EN22	Interrupt Request Enable 22		R/W
b23	EN23	Interrupt Request Enable 23		R/W
b24	EN24	Interrupt Request Enable 24		R/W
b25	EN25	Interrupt Request Enable 25		R/W
b26	EN26	Interrupt Request Enable 26		R/W
b27	EN27	Interrupt Request Enable 27		R/W
b28	EN28	Interrupt Request Enable 28		R/W
b29	EN29	Interrupt Request Enable 29		R/W
b30	EN30	Interrupt Request Enable 30		R/W
b31	EN31	Interrupt Request Enable 31		R/W

Note: When a bit has no corresponding interrupt source (bit is reserved), set the bit to 0.

These registers select whether the ISj flag in the group interrupt request register is set to 1 when each interrupt request for

group interrupt sources is generated. These registers are collectively referred to as the “group interrupt request enable register”.
Registers GENBL0/GENBL1/GENBL2, and GENAL0/GENAL1 control the ISj flag in registers GRPBL0/GRPBL1/GRPBL2, and GRPAL0/GRPAL1, respectively.
Refer to section 14.4.4, Group Interrupts for details on group interrupts.

ENj Bits (Interrupt Request Enable j) (j = 0 to 31)

The ENj bits select whether the corresponding ISj flag in the group interrupt request register is set to 1 when an interrupt request assigned to group interrupts is generated.

(1) Group BL0/BL1/BL2

When a peripheral module interrupt request is generated while the corresponding GENBL0/GENBL1/GENBL2.ENj bit is 1, the GRPBL0/GRPBL1/GRPBL2.ISj flag becomes 1. When the ENj bit is 0, the ISj flag does not become 1.
When the ENj bit is set to 0, the ISj flag becomes 0.

(2) Group AL0/AL1

When a peripheral module interrupt request is generated while the corresponding GENAL0/GENAL1.ENj bit is 1, the GRPAL0/GRPAL1.ISj flag becomes 1. When the ENj bit is 0, the ISj flag does not become 1.
When the ENj bit is set to 0, the ISj flag becomes 0.

14.2.22 Software Configurable Interrupt A Request Register k (PIARk) (k = 0h to Fh, 12h to 14h)

Address(es): ICU.PIAR0 0008 7900h, ICU.PIAR1 0008 7901h, ICU.PIAR2 0008 7902h, ICU.PIAR3 0008 7903h, ICU.PIAR4 0008 7904h, ICU.PIAR5 0008 7905h, ICU.PIAR6 0008 7906h, ICU.PIAR7 0008 7907h, ICU.PIAR8 0008 7908h, ICU.PIAR9 0008 7909h, ICU.PIARA 0008 790Ah, ICU.PIARB 0008 790Bh, ICU.PIARC 0008 790Ch, ICU.PIARD 0008 790Dh, ICU.PIARE 0008 790Eh, ICU.PIARF 0008 790Fh, ICU.PIAR12 0008 7912h, ICU.PIAR13 0008 7913h, ICU.PIAR14 0008 7914h

	b7	b6	b5	b4	b3	b2	b1	b0
	PIR7	PIR6	PIR5	PIR4	PIR3	PIR2	PIR1	PIR0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PIR0	Software Configurable Interrupt A Status Flag 0	When reading 0: Interrupt is not requested. 1: Interrupt is requested. When writing *1 0: Ignored. 1: The Software Configurable interrupt A status flag is cleared.	R/W
b1	PIR1	Software Configurable Interrupt A Status Flag 1		R/W
b2	PIR2	Software Configurable Interrupt A Status Flag 2		R/W
b3	PIR3	Software Configurable Interrupt A Status Flag 3		R/W
b4	PIR4	Software Configurable Interrupt A Status Flag 4		R/W
b5	PIR5	Software Configurable Interrupt A Status Flag 5		R/W
b6	PIR6	Software Configurable Interrupt A Status Flag 6		R/W
b7	PIR7	Software Configurable Interrupt A Status Flag 7		R/W

Note 1. Do not use bit manipulation instructions. If a bit manipulation instruction is used, multiple status flags may be cleared. Write 1 only to the flag to be cleared and write 0 to the other flags (write to this register in 8-bit units).

The PIARk register is used for polling interrupt requests of interrupt sources that is assigned to software configurable interrupt A by software. For an interrupt request of software configurable interrupt A set in the SLIARn register, use the corresponding IRn.IR flag for polling (n = 208 to 255).
Refer to Table 14.3, Interrupt Sources for Software Configurable Interrupt A for correspondence between interrupt source numbers and interrupt sources for software configurable interrupt A.

PIRj Flag (Software Configurable Interrupt A Status Flag j) (j = 0 to 7)

When an interrupt request of interrupt sources assigned to software configurable interrupt A is generated, the corresponding PIARk.PIRj flag becomes 1 regardless of whether the interrupt source is selected in the SLIARn register. Although the PIRj flag does not become 0 after the interrupt request is accepted by the destination (CPU, DTC, DMAC), generation of interrupt requests is not affected.
When using the PIRj flag for polling, set the PIRj flag to 0 by writing 1 to the flag in advance.
This flag becomes 1 under the following condition:

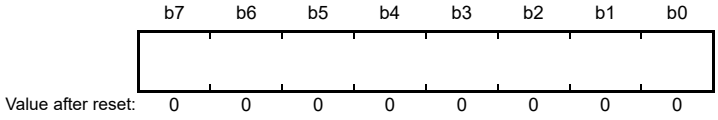
- An interrupt request is generated

This flag becomes 0 under the following condition:

- 1 is written to the PIARk.PIRj flag

14.2.23 Software Configurable Interrupt A Source Select Register n (SLIARn) (n = 208 to 255)

Address(es): ICU.SLIAR208 0008 79D0h to ICU.SLIAR255 0008 79FFh



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	—	00h: No interrupt source selected. 01h: Interrupt source number 1 : FEh: Interrupt source number 254 FFh: No interrupt source selected.	R/(W) *1

Note 1. Writing to these bits is ignored while the SLIPRCR.WPRC bit is set to 1.

The SLIARn register is used to assign interrupt vector numbers 208 to 255 to interrupt sources for software configurable interrupt A.

Table 14.3, Interrupt Sources for Software Configurable Interrupt A lists interrupt sources for software configurable interrupt A. Set the SLIARn register to an interrupt source number that is not reserved. When 00h or FFh are set, no interrupt source is assigned to interrupt vector number n.

Do not assign the same interrupt source to multiple SLIARn registers.

Some interrupt sources can be used to start the DTC or DMAC. Refer to Table 14.3, Interrupt Sources for Software Configurable Interrupt A for details on which sources can be used to start the DTC or DMAC.

Refer to section 14.7.7, Setting Software Configurable Interrupts for the procedure to set software configurable interrupts.

Table 14.3 Interrupt Sources for Software Configurable Interrupt A (1/4)

Interrupt Source No.	Category	Interrupt Request Source	Name	Start the DTC	Start the DMAC	Interrupt Status Flag
0	—	None	No interrupt selected (initial value)	N/A	N/A	PIAR0.PIR0
1	Edge	MTU0	TGIA0 (TGRA input capture/compare match)	✓	✓	PIAR0.PIR1
2			TGIB0 (TGRB input capture/compare match)	✓	✓	PIAR0.PIR2
3			TGIC0 (TGRC input capture/compare match)	✓	✓	PIAR0.PIR3
4			TGID0 (TGRD input capture/compare match)	✓	✓	PIAR0.PIR4
5			TCIV0 (TCNT overflow)	N/A	N/A	PIAR0.PIR5
6			TGIE0 (TGRE compare match)	N/A	N/A	PIAR0.PIR6
7			TGIF0 (TGRF compare match)	N/A	N/A	PIAR0.PIR7
8		MTU1	TGIA1 (TGRA input capture/compare match)	✓	✓	PIAR1.PIR0
9			TGIB1 (TGRB input capture/compare match)	✓	✓	PIAR1.PIR1
10			TCIV1 (TCNT overflow)	N/A	N/A	PIAR1.PIR2
11			TCIU1 (TCNT underflow)	N/A	N/A	PIAR1.PIR3
12		MTU2	TGIA2 (TGRA input capture/compare match)	✓	✓	PIAR1.PIR4
13			TGIB2 (TGRB input capture/compare match)	✓	✓	PIAR1.PIR5
14			TCIV2 (TCNT overflow)	N/A	N/A	PIAR1.PIR6
15			TCIU2 (TCNT underflow)	N/A	N/A	PIAR1.PIR7
16		MTU3	TGIA3 (TGRA input capture/compare match)	✓	✓	PIAR2.PIR0
17			TGIB3 (TGRB input capture/compare match)	✓	✓	PIAR2.PIR1
18			TGIC3 (TGRC input capture/compare match)	✓	✓	PIAR2.PIR2
19			TGID3 (TGRD input capture/compare match)	✓	✓	PIAR2.PIR3
20			TCIV3 (TCNT overflow)	N/A	N/A	PIAR2.PIR4
21		MTU4	TGIA4 (TGRA input capture/compare match)	✓	✓	PIAR2.PIR5
22			TGIB4 (TGRB input capture/compare match)	✓	✓	PIAR2.PIR6
23			TGIC4 (TGRC input capture/compare match)	✓	✓	PIAR2.PIR7
24			TGID4 (TGRD input capture/compare match)	✓	✓	PIAR3.PIR0
25			TCIV4 (TCNT overflow/underflow (only for complementary PWM mode))	✓	✓	PIAR3.PIR1
26		Reserved	—	N/A	N/A	PIAR3.PIR2
27		MTU5	TGIU5 (TGRU input capture/compare match)	✓	✓	PIAR3.PIR3
28			TGIV5 (TGRV input capture/compare match)	✓	✓	PIAR3.PIR4
29			TGIW5 (TGRW input capture/compare match)	✓	✓	PIAR3.PIR5
30		MTU6	TGIA6 (TGRA input capture/compare match)	✓	✓	PIAR3.PIR6
31			TGIB6 (TGRB input capture/compare match)	✓	✓	PIAR3.PIR7
32			TGIC6 (TGRC input capture/compare match)	✓	✓	PIAR4.PIR0
33			TGID6 (TGRD input capture/compare match)	✓	✓	PIAR4.PIR1
34			TCIV6 (TCNT overflow)	N/A	N/A	PIAR4.PIR2
35		MTU7	TGIA7 (TGRA input capture/compare match)	✓	✓	PIAR4.PIR3
36			TGIB7 (TGRB input capture/compare match)	✓	✓	PIAR4.PIR4
37			TGIC7 (TGRC input capture/compare match)	✓	✓	PIAR4.PIR5
38			TGID7 (TGRD input capture/compare match)	✓	✓	PIAR4.PIR6
39			TCIV7 (TCNT overflow/underflow (only for complementary PWM mode))	✓	✓	PIAR4.PIR7
40 to 46		Reserved	—	N/A	N/A	—

Table 14.3 Interrupt Sources for Software Configurable Interrupt A (2/4)

Interrupt Source No.	Category	Interrupt Request Source	Name	Start the DTC	Start the DMAC	Interrupt Status Flag
47	Edge	MTU9	TGIA9 (TGRA input capture/compare match)	✓	✓	PIAR5.PIR7
48			TGIB9 (TGRB input capture/compare match)	✓	✓	PIAR6.PIR0
49			TGIC9 (TGRC input capture/compare match)	✓	✓	PIAR6.PIR1
50			TGID9 (TGRD input capture/compare match)	✓	✓	PIAR6.PIR2
51			TCIV9 (TCNT overflow)	N/A	N/A	PIAR6.PIR3
52			TGIE9 (TGRE compare match)	N/A	N/A	PIAR6.PIR4
53			TGIF9 (TGRF compare match)	N/A	N/A	PIAR6.PIR5
54, 55		Reserved	—	N/A	N/A	—
56		GPTW0	GTCIA0 (GTCCRA input capture/compare match)	✓	✓	PIAR7.PIR0
57			GTCIB0 (GTCCRB input capture/compare match)	✓	✓	PIAR7.PIR1
58			GTCIC0 (GTCCRC compare match)	✓	✓	PIAR7.PIR2
59			GTCID0 (GTCCRD compare match)	✓	✓	PIAR7.PIR3
60			GDTE0 (dead time error)	N/A	N/A	PIAR7.PIR4
61			GTCIE0 (GTCCRE compare match)	✓	✓	PIAR7.PIR5
62			GTCIF0 (GTCCRF compare match)	✓	✓	PIAR7.PIR6
63			GTCIV0 (GTCNT overflow (GTPR compare match))	✓	✓	PIAR7.PIR7
64			GTCIU0 (GTCNT underflow)	✓	✓	PIAR8.PIR0
65		GPTW1	GTCIA1 (GTCCRA input capture/compare match)	✓	✓	PIAR8.PIR1
66			GTCIB1 (GTCCRB input capture/compare match)	✓	✓	PIAR8.PIR2
67			GTCIC1 (GTCCRC compare match)	✓	✓	PIAR8.PIR3
68			GTCID1 (GTCCRD compare match)	✓	✓	PIAR8.PIR4
69			GDTE1 (dead time error)	N/A	N/A	PIAR8.PIR5
70			GTCIE1 (GTCCRE compare match)	✓	✓	PIAR8.PIR6
71			GTCIF1 (GTCCRF compare match)	✓	✓	PIAR8.PIR7
72			GTCIV1 (GTCNT overflow (GTPR compare match))	✓	✓	PIAR9.PIR0
73			GTCIU1 (GTCNT underflow)	✓	✓	PIAR9.PIR1
74		GPTW2	GTCIA2 (GTCCRA input capture/compare match)	✓	✓	PIAR9.PIR2
75			GTCIB2 (GTCCRB input capture/compare match)	✓	✓	PIAR9.PIR3
76			GTCIC2 (GTCCRC compare match)	✓	✓	PIAR9.PIR4
77			GTCID2 (GTCCRD compare match)	✓	✓	PIAR9.PIR5
78			GDTE2 (dead time error)	N/A	N/A	PIAR9.PIR6
79			GTCIE2 (GTCCRE compare match)	✓	✓	PIAR9.PIR7
80			GTCIF2 (GTCCRF compare match)	✓	✓	PIARA.PIR0
81			GTCIV2 (GTCNT overflow (GTPR compare match))	✓	✓	PIARA.PIR1
82			GTCIU2 (GTCNT underflow)	✓	✓	PIARA.PIR2
83		GPTW3	GTCIA3 (GTCCRA input capture/compare match)	✓	✓	PIARA.PIR3
84			GTCIB3 (GTCCRB input capture/compare match)	✓	✓	PIARA.PIR4
85			GTCIC3 (GTCCRC compare match)	✓	✓	PIARA.PIR5
86			GTCID3 (GTCCRD compare match)	✓	✓	PIARA.PIR6
87			GDTE3 (dead time error)	N/A	N/A	PIARA.PIR7
88			GTCIE3 (GTCCRE compare match)	✓	✓	PIARB.PIR0
89			GTCIF3 (GTCCRF compare match)	✓	✓	PIARB.PIR1
90			GTCIV3 (GTCNT overflow (GTPR compare match))	✓	✓	PIARB.PIR2
91			GTCIU3 (GTCNT underflow)	✓	✓	PIARB.PIR3

Table 14.3 Interrupt Sources for Software Configurable Interrupt A (3/4)

Interrupt Source No.	Category	Interrupt Request Source	Name	Start the DTC	Start the DMAC	Interrupt Status Flag
92	Edge	GPTW4	GTCIA4 (GTCCRA input capture/compare match)	✓	✓	PIARB.PIR4
93			GTCIB4 (GTCCRB input capture/compare match)	✓	✓	PIARB.PIR5
94			GTCIC4 (GTCCRC compare match)	✓	✓	PIARB.PIR6
95			GTCID4 (GTCCRD compare match)	✓	✓	PIARB.PIR7
96			GDTE4 (dead time error)	N/A	N/A	PIARC.PIR0
97			GTCIE4 (GTCCRE compare match)	✓	✓	PIARC.PIR1
98			GTCIF4 (GTCCRF compare match)	✓	✓	PIARC.PIR2
99			GTCIV4 (GTCNT overflow (GTPR compare match))	✓	✓	PIARC.PIR3
100			GTCIU4 (GTCNT underflow)	✓	✓	PIARC.PIR4
101		GPTW5	GTCIA5 (GTCCRA input capture/compare match)	✓	✓	PIARC.PIR5
102			GTCIB5 (GTCCRB input capture/compare match)	✓	✓	PIARC.PIR6
103			GTCIC5 (GTCCRC compare match)	✓	✓	PIARC.PIR7
104			GTCID5 (GTCCRD compare match)	✓	✓	PIARD.PIR0
105			GDTE5 (dead time error)	N/A	N/A	PIARD.PIR1
106			GTCIE5 (GTCCRE compare match)	✓	✓	PIARD.PIR2
107			GTCIF5 (GTCCRF compare match)	✓	✓	PIARD.PIR3
108			GTCIV5 (GTCNT overflow (GTPR compare match))	✓	✓	PIARD.PIR4
109			GTCIU5 (GTCNT underflow)	✓	✓	PIARD.PIR5
110		GPTW6	GTCIA6 (GTCCRA input capture/compare match)	✓	✓	PIARD.PIR6
111			GTCIB6 (GTCCRB input capture/compare match)	✓	✓	PIARD.PIR7
112			GTCIC6 (GTCCRC compare match)	✓	✓	PIARE.PIR0
113			GTCID6 (GTCCRD compare match)	✓	✓	PIARE.PIR1
114			GDTE6 (dead time error)	N/A	N/A	PIARE.PIR2
115			GTCIE6 (GTCCRE compare match)	✓	✓	PIARE.PIR3
116			GTCIF6 (GTCCRF compare match)	✓	✓	PIARE.PIR4
117			GTCIV6 (GTCNT overflow (GTPR compare match))	✓	✓	PIARE.PIR5
118			GTCIU6 (GTCNT underflow)	✓	✓	PIARE.PIR6
119		GPTW7	GTCIA7 (GTCCRA input capture/compare match)	✓	✓	PIARE.PIR7
120			GTCIB7 (GTCCRB input capture/compare match)	✓	✓	PIARF.PIR0
121			GTCIC7 (GTCCRC compare match)	✓	✓	PIARF.PIR1
122			GTCID7 (GTCCRD compare match)	✓	✓	PIARF.PIR2
123			GDTE7 (dead time error)	N/A	N/A	PIARF.PIR3
124			GTCIE7 (GTCCRE compare match)	✓	✓	PIARF.PIR4
125			GTCIF7 (GTCCRF compare match)	✓	✓	PIARF.PIR5
126			GTCIV7 (GTCNT overflow (GTPR compare match))	✓	✓	PIARF.PIR6
127			GTCIU7 (GTCNT underflow)	✓	✓	PIARF.PIR7
128 to 145		Reserved	—	N/A	N/A	—
146		RSPIA0	SPCI (communication end)	N/A	N/A	PIAR12.PIR2
147		RSPI0	SPCI0 (communication end)	N/A	N/A	PIAR12.PIR3
148 to 150		Reserved	—	N/A	N/A	—
151		RSCI11	AED (active edge detected)	N/A	N/A	PIAR12.PIR7
152		CANFD	EC1EI (1-bit ECC error)	N/A	N/A	PIAR13.PIR0
153			EC2EI (2-bit ECC error)	N/A	N/A	PIAR13.PIR1
154			ECOVI (ECC overflow)	N/A	N/A	PIAR13.PIR2

Table 14.3 Interrupt Sources for Software Configurable Interrupt A (4/4)

Interrupt Source No.	Category	Interrupt Request Source	Name	Start the DTC	Start the DMAC	Interrupt Status Flag
155 to 159		Reserved	—	N/A	N/A	—
160		GPTW0	GTCEI0 (cycle count end)	N/A	N/A	PIAR14.PIR0
161		GPTW1	GTCEI1 (cycle count end)	N/A	N/A	PIAR14.PIR1
162		GPTW2	GTCEI2 (cycle count end)	N/A	N/A	PIAR14.PIR2
163		GPTW3	GTCEI3 (cycle count end)	N/A	N/A	PIAR14.PIR3
164		GPTW4	GTCEI4 (cycle count end)	N/A	N/A	PIAR14.PIR4
165		GPTW5	GTCEI5 (cycle count end)	N/A	N/A	PIAR14.PIR5
166		GPTW6	GTCEI6 (cycle count end)	N/A	N/A	PIAR14.PIR6
167		GPTW7	GTCEI7 (cycle count end)	N/A	N/A	PIAR14.PIR7
168 to 254		Reserved	—	N/A	N/A	—
255	—	None	No interrupt selected	N/A	N/A	—

14.2.24 Software Configurable Interrupt Source Select Register Write Protect Register (SLIPRCR)

Address(es): ICU.SLIPRCR 0008 7A00h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	WPRC
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	WPRC	Software Configurable Interrupt Source Select Register Write Protect	0: Write enabled. 1: Write disabled.	R/(W) *1
b7 to b1	—	Reserved	The read value is 0. The write value should be 0.	R

Note 1. Once this bit is set to 1, it cannot be set to 0 by software.

The SLIPRCR register protects registers that control assignment of software configurable interrupts from being written to.

WPRC Bit (Software Configurable Interrupt Source Select Register Write Protect)

The WPRC bit disables writing to the SLIARn register.

Once this bit is set to 1, it cannot be set to 0 by software.

After assigning software configurable interrupts, confirm that the WPRC bit is 1 before the corresponding interrupt request is generated. Refer to section 14.7.7, Setting Software Configurable Interrupts for the procedure to set software configurable interrupts.

14.3 Vector Table

There are two types of exceptions detected by the ICU: maskable interrupts (hereinafter referred to as “interrupts”) and non-maskable interrupts.
When the CPU accepts an interrupt or non-maskable interrupt, it acquires a 4-byte vector address from the vector table.

14.3.1 Interrupt Vector Table

The vector table that is used for maskable interrupts is called the interrupt vector table.
The interrupt vector table is allocated to a 1024-byte area (4 bytes × 256 sources) beginning with the address set in the INTB register in the CPU. Set the INTB register before enabling interrupts. Set a multiple of 4 in the INTB register.
An unconditional trap is generated when the INT or BRK instruction is executed. Interrupt vectors for unconditional traps use the same area as the interrupt vector table. The BRK instruction is assigned to interrupt vector number 0. The INT instruction is assigned to the interrupt vector number corresponding to the value set as the operand (0 to 255).
Table 14.4 lists details of the interrupt vectors. Details of the headings in Table 14.4 are listed below.

Heading	Description
Interrupt request generated by	Name of the source that generates the interrupt request (module symbol)
Name	Name of the interrupt source (symbol)
Vector no.	Interrupt vector number
Vector address offset	Offset from the address set in the INTB register
Interrupt detection method	“Edge” indicates that the interrupt is detected by edge detection. “Level” indicates that the interrupt is detected by level detection.
CPU interrupt	Interrupt source indicated by “✓” can be used as an interrupt source to the CPU.
Start the DTC	Interrupt source indicated by “✓” can be used as the DTC trigger.
Start the DMAC	Interrupt source indicated by “✓” can be used as the DMAC trigger.
Exit from SSBY	Interrupt source indicated by “✓” can be used as a source to exit software standby mode.
Exit from ACS	Interrupt source indicated by “✓” can be used as a source to exit all-module clock stop mode.
IER	Name of the bit in the IER register corresponding to the interrupt vector number
IPR	Name of the IPR register corresponding to the interrupt source
DTCER	Name of the DTCER register corresponding to the DTC trigger

Table 14.4 Interrupt Vector Table (1/6)

Interrupt Request is Generated By	Name	Vector No.	Vector Address Offset	Interrupt Detection Method	CPU Interrupt	Start the DTC	Start the DMAC	Exit from SSBY	Exit from ACS	IER	IPR	DTCER
—	For an unconditional trap	0	0000h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	1	0004h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	2	0008h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	3	000Ch	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	4	0010h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	5	0014h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	6	0018h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	7	001Ch	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	8	0020h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	9	0024h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	10	0028h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	11	002Ch	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	12	0030h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	13	0034h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	14	0038h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	15	003Ch	—	N/A	N/A	N/A	N/A	N/A	—	—	—
BSC	BUSERR	16	0040h	Level	✓	N/A	N/A	N/A	N/A	IER02.IEN0	IPR000	—
ICU*1	GROUPIE0	17	0044h	Level	✓	N/A	N/A	N/A	N/A	IER02.IEN1	IPR000	—
RAM	RAMERR*2	18	0048h	Level	✓	N/A	N/A	N/A	N/A	IER02.IEN2	IPR000	—
—	Reserved	19	004Ch	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	20	0050h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
FCU	FIFERR	21	0054h	Level	✓	N/A	N/A	N/A	N/A	IER02.IEN5	IPR001	—
—	Reserved	22	0058h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
FCU	FRDYI	23	005Ch	Edge	✓	N/A	N/A	N/A	N/A	IER02.IEN7	IPR002	—
—	Reserved	24	0060h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	25	0064h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
ICU	SWINT2	26	0068h	Edge	✓	✓	N/A	N/A	N/A	IER03.IEN2	IPR003	DTCER026
	SWINT	27	006Ch	Edge	✓	✓	N/A	N/A	N/A	IER03.IEN3		DTCER027
CMT0	CMI0 (for OS)	28	0070h	Edge	✓	✓	✓	N/A	N/A	IER03.IEN4	IPR004	DTCER028
CMT1	CMI1	29	0074h	Edge	✓	✓	✓	N/A	N/A	IER03.IEN5	IPR005	DTCER029
CMT2	CMI2	30	0078h	Edge	✓	✓	✓	N/A	N/A	IER03.IEN6	IPR006	DTCER030
CMT3	CMI3	31	007Ch	Edge	✓	✓	✓	N/A	N/A	IER03.IEN7	IPR007	DTCER031
—	Reserved	32	0080h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	33	0084h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	34	0088h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	35	008Ch	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	36	0090h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	37	0094h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
RSPI0	SPRI0	38	0098h	Edge	✓	✓	✓	N/A	N/A	IER04.IEN6	IPR038	DTCER038
	SPTI0	39	009Ch	Edge	✓	✓	✓	N/A	N/A	IER04.IEN7	IPR039	DTCER039
RI3C0	RESPI	40	00A0h	Edge	✓	✓	✓	N/A	N/A	IER05.IEN0	IPR040	DTCER040
	CMDI	41	00A4h	Edge	✓	✓	✓	N/A	N/A	IER05.IEN1	IPR041	DTCER041
	IBII	42	00A8h	Edge	✓	✓	✓	N/A	N/A	IER05.IEN2	IPR042	DTCER042
	RCVI	43	00ACh	Edge	✓	✓	✓	N/A	N/A	IER05.IEN3	IPR043	DTCER043

Table 14.4 Interrupt Vector Table (2/6)

Interrupt Request is Generated By	Name	Vector No.	Vector Address Offset	Interrupt Detection Method	CPU Interrupt	Start the DTC	Start the DMAC	Exit from SSBY	Exit from ACS	IER	IPR	DTCER
—	Reserved	44	00B0h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	45	00B4h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	46	00B8h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	47	00BCh	—	N/A	N/A	N/A	N/A	N/A	—	—	—
RSPIA0	SPRI	48	00C0h	Edge	✓	✓	✓	N/A	N/A	IER06.IEN0	IPR048	DTCER048
	SPTI	49	00C4h	Edge	✓	✓	✓	N/A	N/A	IER06.IEN1	IPR049	DTCER049
—	Reserved	50	00C8h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	51	00CCh	—	N/A	N/A	N/A	N/A	N/A	—	—	—
RIIC0	RXI0	52	00D0h	Edge	✓	✓	✓	N/A	N/A	IER06.IEN4	IPR052	DTCER052
	TXI0	53	00D4h	Edge	✓	✓	✓	N/A	N/A	IER06.IEN5	IPR053	DTCER053
—	Reserved	54	00D8h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	55	00DCh	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	56	00E0h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	57	00E4h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	58	00E8h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	59	00ECh	—	N/A	N/A	N/A	N/A	N/A	—	—	—
SCI1	RXI1	60	00F0h	Edge	✓	✓	✓	N/A	N/A	IER07.IEN4	IPR060	DTCER060
	TXI1	61	00F4h	Edge	✓	✓	✓	N/A	N/A	IER07.IEN5	IPR061	DTCER061
—	Reserved	62	00F8h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	63	00FCh	—	N/A	N/A	N/A	N/A	N/A	—	—	—
ICU	IRQ0	64	0100h	Edge/ Level	✓	✓	✓	✓	✓	IER08.IEN0	IPR064	DTCER064
	IRQ1	65	0104h	Edge/ Level	✓	✓	✓	✓	✓	IER08.IEN1	IPR065	DTCER065
	IRQ2	66	0108h	Edge/ Level	✓	✓	✓	✓	✓	IER08.IEN2	IPR066	DTCER066
	IRQ3	67	010Ch	Edge/ Level	✓	✓	✓	✓	✓	IER08.IEN3	IPR067	DTCER067
	IRQ4	68	0110h	Edge/ Level	✓	✓	✓	✓	✓	IER08.IEN4	IPR068	DTCER068
	IRQ5	69	0114h	Edge/ Level	✓	✓	✓	✓	✓	IER08.IEN5	IPR069	DTCER069
	IRQ6	70	0118h	Edge/ Level	✓	✓	✓	✓	✓	IER08.IEN6	IPR070	DTCER070
	IRQ7	71	011Ch	Edge/ Level	✓	✓	✓	✓	✓	IER08.IEN7	IPR071	DTCER071
	IRQ8	72	0120h	Edge/ Level	✓	✓	✓	✓	✓	IER09.IEN0	IPR072	DTCER072
	IRQ9	73	0124h	Edge/ Level	✓	✓	✓	✓	✓	IER09.IEN1	IPR073	DTCER073
	IRQ10	74	0128h	Edge/ Level	✓	✓	✓	✓	✓	IER09.IEN2	IPR074	DTCER074
	IRQ11	75	012Ch	Edge/ Level	✓	✓	✓	✓	✓	IER09.IEN3	IPR075	DTCER075
	IRQ12	76	0130h	Edge/ Level	✓	✓	✓	✓	✓	IER09.IEN4	IPR076	DTCER076
	IRQ13	77	0134h	Edge/ Level	✓	✓	✓	✓	✓	IER09.IEN5	IPR077	DTCER077
	IRQ14	78	0138h	Edge/ Level	✓	✓	✓	✓	✓	IER09.IEN6	IPR078	DTCER078
	IRQ15	79	013Ch	Edge/ Level	✓	✓	✓	✓	✓	IER09.IEN7	IPR079	DTCER079

Table 14.4 Interrupt Vector Table (3/6)

Interrupt Request is Generated By	Name	Vector No.	Vector Address Offset	Interrupt Detection Method	CPU Interrupt	Start the DTC	Start the DMAC	Exit from SSBY	Exit from ACS	IER	IPR	DTCER
—	Reserved	80	0140h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	81	0144h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	82	0148h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	83	014Ch	—	N/A	N/A	N/A	N/A	N/A	—	—	—
SCI5	RXI5	84	0150h	Edge	✓	✓	✓	N/A	N/A	IER0A.IEN4	IPR084	DTCER084
	TXI5	85	0154h	Edge	✓	✓	✓	N/A	N/A	IER0A.IEN5	IPR085	DTCER085
SCI6	RXI6	86	0158h	Edge	✓	✓	✓	N/A	N/A	IER0A.IEN6	IPR086	DTCER086
	TXI6	87	015Ch	Edge	✓	✓	✓	N/A	N/A	IER0A.IEN7	IPR087	DTCER087
LVD1	LVD1	88	0160h	Edge	✓	N/A	N/A	✓	✓	IER0B.IEN0	IPR088	—
LVD2	LVD2	89	0164h	Edge	✓	N/A	N/A	✓	✓	IER0B.IEN1	IPR089	—
—	Reserved	90	0168h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	91	016Ch	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	92	0170h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	93	0174h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	94	0178h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
IWDT	IWUNI*2	95	017Ch	Edge	✓	N/A	N/A	✓	✓	IER0B.IEN7	IPR095	—
WDT	WUNI*2	96	0180h	Edge	✓	N/A	N/A	N/A	N/A	IER0C.IEN0	IPR096	—
—	Reserved	97	0184h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	98	0188h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	99	018Ch	—	N/A	N/A	N/A	N/A	N/A	—	—	—
RSCI8	RXI	100	0190h	Edge	✓	✓	✓	N/A	N/A	IER0C.IEN4	IPR100	DTCER100
	TXI	101	0194h	Edge	✓	✓	✓	N/A	N/A	IER0C.IEN5	IPR101	DTCER101
RSCI9	RXI	102	0198h	Edge	✓	✓	✓	N/A	N/A	IER0C.IEN6	IPR102	DTCER102
	TXI	103	019Ch	Edge	✓	✓	✓	N/A	N/A	IER0C.IEN7	IPR103	DTCER103
—	Reserved	104	01A0h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	105	01A4h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
ICU*1	GROUPBE0	106	01A8h	Level	✓	N/A	N/A	N/A	N/A	IER0D.IEN2	IPR106	—
	GROUPBL2	107	01ACh	Level	✓	N/A	N/A	N/A	N/A	IER0D.IEN3	IPR107	—
	Reserved	108	01B0h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
	Reserved	109	01B4h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
	GROUPBL0	110	01B8h	Level	✓	N/A	N/A	N/A	N/A	IER0D.IEN6	IPR110	—
	GROUPBL1	111	01BCh	Level	✓	N/A	N/A	N/A	N/A	IER0D.IEN7	IPR111	—
	GROUPAL0	112	01C0h	Level	✓	N/A	N/A	N/A	N/A	IER0E.IEN0	IPR112	—
	GROUPAL1	113	01C4h	Level	✓	N/A	N/A	N/A	N/A	IER0E.IEN1	IPR113	—
RSCI11	RXI	114	01C8h	Edge	✓	✓	✓	N/A	N/A	IER0E.IEN2	IPR114	DTCER114
	TXI	115	01CCh	Edge	✓	✓	✓	N/A	N/A	IER0E.IEN3	IPR115	DTCER115
SCI12	RXI12	116	01D0h	Edge	✓	✓	✓	N/A	N/A	IER0E.IEN4	IPR116	DTCER116
	TXI12	117	01D4h	Edge	✓	✓	✓	N/A	N/A	IER0E.IEN5	IPR117	DTCER117
RI3C0	RXI	118	01D8h	Edge	✓	✓	✓	N/A	N/A	IER0E.IEN6	IPR118	DTCER118
	TXI	119	01DCh	Edge	✓	✓	✓	N/A	N/A	IER0E.IEN7	IPR119	DTCER119
DMAC	DMAC0I	120	01E0h	Edge	✓	✓	N/A	N/A	N/A	IER0F.IEN0	IPR120	DTCER120
	DMAC1I	121	01E4h	Edge	✓	✓	N/A	N/A	N/A	IER0F.IEN1	IPR121	DTCER121
	DMAC2I	122	01E8h	Edge	✓	✓	N/A	N/A	N/A	IER0F.IEN2	IPR122	DTCER122
	DMAC3I	123	01ECh	Edge	✓	✓	N/A	N/A	N/A	IER0F.IEN3	IPR123	DTCER123
	DMAC74I	124	01F0h	Level	✓	N/A	N/A	N/A	N/A	IER0F.IEN4	IPR124	—

Table 14.4 Interrupt Vector Table (4/6)

Interrupt Request is Generated By	Name	Vector No.	Vector Address Offset	Interrupt Detection Method	CPU Interrupt	Start the DTC	Start the DMAC	Exit from SSBY	Exit from ACS	IER	IPR	DTCER
OST	OSTDI*2	125	01F4h	Edge	✓	N/A	N/A	N/A	N/A	IER0F.IEN5	IPR125	—
—	Reserved	126	01F8h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	127	01FCh	—	N/A	N/A	N/A	N/A	N/A	—	—	—
S12AD	S12ADI	128	0200h	Edge	✓	✓	✓	N/A	N/A	IER10.IEN0	IPR128	DTCER128
	S12GBADI	129	0204h	Edge	✓	✓	✓	N/A	N/A	IER10.IEN1	IPR129	DTCER129
	S12GCADI	130	0208h	Edge	✓	✓	✓	N/A	N/A	IER10.IEN2	IPR130	DTCER130
—	Reserved	131	020Ch	—	N/A	N/A	N/A	N/A	N/A	—	—	—
S12AD1	S12ADI1	132	0210h	Edge	✓	✓	✓	N/A	N/A	IER10.IEN4	IPR132	DTCER132
	S12GBADI1	133	0214h	Edge	✓	✓	✓	N/A	N/A	IER10.IEN5	IPR133	DTCER133
	S12GCADI1	134	0218h	Edge	✓	✓	✓	N/A	N/A	IER10.IEN6	IPR134	DTCER134
—	Reserved	135	021Ch	—	N/A	N/A	N/A	N/A	N/A	—	—	—
S12AD2	S12ADI2	136	0220h	Edge	✓	✓	✓	N/A	N/A	IER11.IEN0	IPR136	DTCER136
	S12GBADI2	137	0224h	Edge	✓	✓	✓	N/A	N/A	IER11.IEN1	IPR137	DTCER137
	S12GCADI2	138	0228h	Edge	✓	✓	✓	N/A	N/A	IER11.IEN2	IPR138	DTCER138
—	Reserved	139	022Ch	—	N/A	N/A	N/A	N/A	N/A	—	—	—
CANFD	RFDREQ0	140	0230h	Edge	✓	✓	✓	N/A	N/A	IER11.IEN4	IPR140	DTCER140
	RFDREQ1	141	0234h	Edge	✓	✓	✓	N/A	N/A	IER11.IEN5	IPR141	DTCER141
CANFD0	CFDREQ0	142	0238h	Edge	✓	✓	✓	N/A	N/A	IER11.IEN6	IPR142	DTCER142
—	Reserved	143	023Ch	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	144	0240h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	145	0244h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
TMR0	CMIA0	146	0248h	Edge	✓	✓	N/A	N/A	✓	IER12.IEN2	IPR146	DTCER146
	CMIB0	147	024Ch	Edge	✓	✓	N/A	N/A	✓	IER12.IEN3		DTCER147
	OVI0	148	0250h	Edge	✓	N/A	N/A	N/A	✓	IER12.IEN4		—
TMR1	CMIA1	149	0254h	Edge	✓	✓	N/A	N/A	✓	IER12.IEN5	IPR149	DTCER149
	CMIB1	150	0258h	Edge	✓	✓	N/A	N/A	✓	IER12.IEN6		DTCER150
	OVI1	151	025Ch	Edge	✓	N/A	N/A	N/A	✓	IER12.IEN7		—
TMR2	CMIA2	152	0260h	Edge	✓	✓	N/A	N/A	✓	IER13.IEN0	IPR152	DTCER152
	CMIB2	153	0264h	Edge	✓	✓	N/A	N/A	✓	IER13.IEN1		DTCER153
	OVI2	154	0268h	Edge	✓	N/A	N/A	N/A	✓	IER13.IEN2		—
TMR3	CMIA3	155	026Ch	Edge	✓	✓	N/A	N/A	✓	IER13.IEN3	IPR155	DTCER155
	CMIB3	156	0270h	Edge	✓	✓	N/A	N/A	✓	IER13.IEN4		DTCER156
	OVI3	157	0274h	Edge	✓	N/A	N/A	N/A	✓	IER13.IEN5		—
TMR4	CMIA4	158	0278h	Edge	✓	✓	N/A	N/A	N/A	IER13.IEN6	IPR158	DTCER158
	CMIB4	159	027Ch	Edge	✓	✓	N/A	N/A	N/A	IER13.IEN7		DTCER159
	OVI4	160	0280h	Edge	✓	N/A	N/A	N/A	N/A	IER14.IEN0		—
TMR5	CMIA5	161	0284h	Edge	✓	✓	N/A	N/A	N/A	IER14.IEN1	IPR161	DTCER161
	CMIB5	162	0288h	Edge	✓	✓	N/A	N/A	N/A	IER14.IEN2		DTCER162
	OVI5	163	028Ch	Edge	✓	N/A	N/A	N/A	N/A	IER14.IEN3		—
TMR6	CMIA6	164	0290h	Edge	✓	✓	N/A	N/A	N/A	IER14.IEN4	IPR164	DTCER164
	CMIB6	165	0294h	Edge	✓	✓	N/A	N/A	N/A	IER14.IEN5		DTCER165
	OVI6	166	0298h	Edge	✓	N/A	N/A	N/A	N/A	IER14.IEN6		—
TMR7	CMIA7	167	029Ch	Edge	✓	✓	N/A	N/A	N/A	IER14.IEN7	IPR167	DTCER167
	CMIB7	168	02A0h	Edge	✓	✓	N/A	N/A	N/A	IER15.IEN0		DTCER168
	OVI7	169	02A4h	Edge	✓	N/A	N/A	N/A	N/A	IER15.IEN1		—

Table 14.4 Interrupt Vector Table (5/6)

Interrupt Request is Generated By	Name	Vector No.	Vector Address Offset	Interrupt Detection Method	CPU Interrupt	Start the DTC	Start the DMAC	Exit from SSBY	Exit from ACS	IER	IPR	DTCER
—	Reserved	170	02A8h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	171	02ACh	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	172	02B0h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	173	02B4h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	174	02B8h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
ELC	ELSR18I	175	02BCh	Edge	✓	✓	✓	N/A	N/A	IER15.IEN7	IPR175	DTCER175
	ELSR19I	176	02C0h	Edge	✓	✓	✓	N/A	N/A	IER16.IEN0	IPR176	DTCER176
TSIP	RD	177	02C4h	Edge	✓	✓	✓	N/A	N/A	IER16.IEN1	IPR177	DTCER177
	WR	178	02C8h	Edge	✓	✓	✓	N/A	N/A	IER16.IEN2		DTCER178
	ERR	179	02CCh	Edge	✓	N/A	N/A	N/A	N/A	IER16.IEN3	IPR179	—
CMPC0	CMPC0	180	02D0h	Edge	✓	✓	✓	N/A	N/A	IER16.IEN4	IPR180	DTCER180
CMPC1	CMPC1	181	02D4h	Edge	✓	✓	✓	N/A	N/A	IER16.IEN5	IPR181	DTCER181
CMPC2	CMPC2	182	02D8h	Edge	✓	✓	✓	N/A	N/A	IER16.IEN6	IPR182	DTCER182
CMPC3	CMPC3	183	02DCh	Edge	✓	✓	✓	N/A	N/A	IER16.IEN7	IPR183	DTCER183
CMPC4	CMPC4	184	02E0h	Edge	✓	✓	✓	N/A	N/A	IER17.IEN0	IPR184	DTCER184
CMPC5	CMPC5	185	02E4h	Edge	✓	✓	✓	N/A	N/A	IER17.IEN1	IPR185	DTCER185
CMTW0	CMWI0	186	02E8h	Edge	✓	✓	✓	N/A	N/A	IER17.IEN2	IPR186	DTCER186
	IC0I0	187	02ECh	Edge	✓	✓	✓	N/A	N/A	IER17.IEN3	IPR187	DTCER187
	IC1I0	188	02F0h	Edge	✓	✓	✓	N/A	N/A	IER17.IEN4	IPR188	DTCER188
	OC0I0	189	02F4h	Edge	✓	✓	✓	N/A	N/A	IER17.IEN5	IPR189	DTCER189
	OC1I0	190	02F8h	Edge	✓	✓	✓	N/A	N/A	IER17.IEN6	IPR190	DTCER190
CMTW1	CMWI1	191	02FCh	Edge	✓	✓	✓	N/A	N/A	IER17.IEN7	IPR191	DTCER191
	IC0I1	192	0300h	Edge	✓	✓	✓	N/A	N/A	IER18.IEN0	IPR192	DTCER192
	IC1I1	193	0304h	Edge	✓	✓	✓	N/A	N/A	IER18.IEN1	IPR193	DTCER193
	OC0I1	194	0308h	Edge	✓	✓	✓	N/A	N/A	IER18.IEN2	IPR194	DTCER194
	OC1I1	195	030Ch	Edge	✓	✓	✓	N/A	N/A	IER18.IEN3	IPR195	DTCER195
—	Reserved	196	0310h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
RSCI9	AED	197	0314h	Edge	✓	✓	✓	N/A	N/A	IER18.IEN5	IPR197	DTCER197
—	Reserved	198	0318h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	199	031Ch	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	200	0320h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	201	0324h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	202	0328h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	203	032Ch	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	204	0330h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	205	0334h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	206	0338h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	207	033Ch	—	N/A	N/A	N/A	N/A	N/A	—	—	—
PERIA (software configurable interrupt A *3)	INTA208	208	0340h	Edge	✓	✓	✓	N/A	N/A	IER1A.IEN0	IPR208	DTCER208
	INTA209	209	0344h	Edge	✓	✓	✓	N/A	N/A	IER1A.IEN1	IPR209	DTCER209
	INTA210	210	0348h	Edge	✓	✓	✓	N/A	N/A	IER1A.IEN2	IPR210	DTCER210
	INTA211	211	034Ch	Edge	✓	✓	✓	N/A	N/A	IER1A.IEN3	IPR211	DTCER211
	INTA212	212	0350h	Edge	✓	✓	✓	N/A	N/A	IER1A.IEN4	IPR212	DTCER212
	INTA213	213	0354h	Edge	✓	✓	✓	N/A	N/A	IER1A.IEN5	IPR213	DTCER213
	INTA214	214	0358h	Edge	✓	✓	✓	N/A	N/A	IER1A.IEN6	IPR214	DTCER214

Table 14.4 Interrupt Vector Table (6/6)

Interrupt Request is Generated By	Name	Vector No.	Vector Address Offset	Interrupt Detection Method	CPU Interrupt	Start the DTC	Start the DMAC	Exit from SSBY	Exit from ACS	IER	IPR	DTCER
PERIA (software configurable interrupt A *3)	INTA215	215	035Ch	Edge	✓	✓	✓	N/A	N/A	IER1A.IEN7	IPR215	DTCER215
	INTA216	216	0360h	Edge	✓	✓	✓	N/A	N/A	IER1B.IEN0	IPR216	DTCER216
	INTA217	217	0364h	Edge	✓	✓	✓	N/A	N/A	IER1B.IEN1	IPR217	DTCER217
	INTA218	218	0368h	Edge	✓	✓	✓	N/A	N/A	IER1B.IEN2	IPR218	DTCER218
	INTA219	219	036Ch	Edge	✓	✓	✓	N/A	N/A	IER1B.IEN3	IPR219	DTCER219
	INTA220	220	0370h	Edge	✓	✓	✓	N/A	N/A	IER1B.IEN4	IPR220	DTCER220
	INTA221	221	0374h	Edge	✓	✓	✓	N/A	N/A	IER1B.IEN5	IPR221	DTCER221
	INTA222	222	0378h	Edge	✓	✓	✓	N/A	N/A	IER1B.IEN6	IPR222	DTCER222
	INTA223	223	037Ch	Edge	✓	✓	✓	N/A	N/A	IER1B.IEN7	IPR223	DTCER223
	INTA224	224	0380h	Edge	✓	✓	✓	N/A	N/A	IER1C.IEN0	IPR224	DTCER224
	INTA225	225	0384h	Edge	✓	✓	✓	N/A	N/A	IER1C.IEN1	IPR225	DTCER225
	INTA226	226	0388h	Edge	✓	✓	✓	N/A	N/A	IER1C.IEN2	IPR226	DTCER226
	INTA227	227	038Ch	Edge	✓	✓	✓	N/A	N/A	IER1C.IEN3	IPR227	DTCER227
	INTA228	228	0390h	Edge	✓	✓	✓	N/A	N/A	IER1C.IEN4	IPR228	DTCER228
	INTA229	229	0394h	Edge	✓	✓	✓	N/A	N/A	IER1C.IEN5	IPR229	DTCER229
	INTA230	230	0398h	Edge	✓	✓	✓	N/A	N/A	IER1C.IEN6	IPR230	DTCER230
	INTA231	231	039Ch	Edge	✓	✓	✓	N/A	N/A	IER1C.IEN7	IPR231	DTCER231
	INTA232	232	03A0h	Edge	✓	✓	✓	N/A	N/A	IER1D.IEN0	IPR232	DTCER232
	INTA233	233	03A4h	Edge	✓	✓	✓	N/A	N/A	IER1D.IEN1	IPR233	DTCER233
	INTA234	234	03A8h	Edge	✓	✓	✓	N/A	N/A	IER1D.IEN2	IPR234	DTCER234
	INTA235	235	03ACh	Edge	✓	✓	✓	N/A	N/A	IER1D.IEN3	IPR235	DTCER235
	INTA236	236	03B0h	Edge	✓	✓	✓	N/A	N/A	IER1D.IEN4	IPR236	DTCER236
	INTA237	237	03B4h	Edge	✓	✓	✓	N/A	N/A	IER1D.IEN5	IPR237	DTCER237
	INTA238	238	03B8h	Edge	✓	✓	✓	N/A	N/A	IER1D.IEN6	IPR238	DTCER238
	INTA239	239	03BCh	Edge	✓	✓	✓	N/A	N/A	IER1D.IEN7	IPR239	DTCER239
	INTA240	240	03C0h	Edge	✓	✓	✓	N/A	N/A	IER1E.IEN0	IPR240	DTCER240
	INTA241	241	03C4h	Edge	✓	✓	✓	N/A	N/A	IER1E.IEN1	IPR241	DTCER241
	INTA242	242	03C8h	Edge	✓	✓	✓	N/A	N/A	IER1E.IEN2	IPR242	DTCER242
	INTA243	243	03CCh	Edge	✓	✓	✓	N/A	N/A	IER1E.IEN3	IPR243	DTCER243
	INTA244	244	03D0h	Edge	✓	✓	✓	N/A	N/A	IER1E.IEN4	IPR244	DTCER244
	INTA245	245	03D4h	Edge	✓	✓	✓	N/A	N/A	IER1E.IEN5	IPR245	DTCER245
	INTA246	246	03D8h	Edge	✓	✓	✓	N/A	N/A	IER1E.IEN6	IPR246	DTCER246
	INTA247	247	03DCh	Edge	✓	✓	✓	N/A	N/A	IER1E.IEN7	IPR247	DTCER247
	INTA248	248	03E0h	Edge	✓	✓	✓	N/A	N/A	IER1F.IEN0	IPR248	DTCER248
	INTA249	249	03E4h	Edge	✓	✓	✓	N/A	N/A	IER1F.IEN1	IPR249	DTCER249
	INTA250	250	03E8h	Edge	✓	✓	✓	N/A	N/A	IER1F.IEN2	IPR250	DTCER250
	INTA251	251	03ECh	Edge	✓	✓	✓	N/A	N/A	IER1F.IEN3	IPR251	DTCER251
	INTA252	252	03F0h	Edge	✓	✓	✓	N/A	N/A	IER1F.IEN4	IPR252	DTCER252
	INTA253	253	03F4h	Edge	✓	✓	✓	N/A	N/A	IER1F.IEN5	IPR253	DTCER253
	INTA254	254	03F8h	Edge	✓	✓	✓	N/A	N/A	IER1F.IEN6	IPR254	DTCER254
	INTA255	255	03FCh	Edge	✓	✓	✓	N/A	N/A	IER1F.IEN7	IPR255	DTCER255

Note: This table lists the interrupt vectors for the maximum specification. The interrupt vectors for individual products correspond to the functions listed in Table 1.2. For details, refer to Table 1.2, Comparison of Functions for Different Packages.
Note 1. For the group interrupt sources, refer to Table 14.6, Group Interrupt Requests.
Note 2. This is the case where the corresponding non-maskable interrupt enable bit is set to 0 (disabled).
Note 3. For the software configurable interrupt A sources, refer to Table 14.3, Interrupt Sources for Software Configurable Interrupt A.

Note that some interrupt sources cannot start the DTC or DMAC.

14.3.2 Fast Interrupt Vector Area

The interrupt set as the fast interrupt uses the FINTV register in the CPU. Set the FINTV register before enabling the fast interrupt.

14.3.3 Non-maskable Interrupt Vector Area

Non-maskable interrupts use the vector area in the exception vector table.
The exception vector table is allocated to the 128-byte area (4 bytes × 32 sources) beginning with the address set in the EXTB register in the CPU. Set the EXTB register before enabling non-maskable interrupts. Set a multiple of 4 in the EXTB register.

14.4 Types of Interrupts

Interrupts are divided into maskable interrupts and non-maskable interrupts. Maskable interrupts can be masked by the PSW.I bit or IPL[3:0] bits of the processor status word in the CPU. Non-maskable interrupts can be accepted by the CPU regardless of those bits. While interrupt sources assigned to vector numbers 0 to 207 are fixed, an interrupt source assigned to each vector number from 208 to 255 (software configurable interrupt) can be selected from multiple sources. Note that maskable interrupts are referred to as interrupts in this chapter.
Figure 14.2 shows types of interrupts.

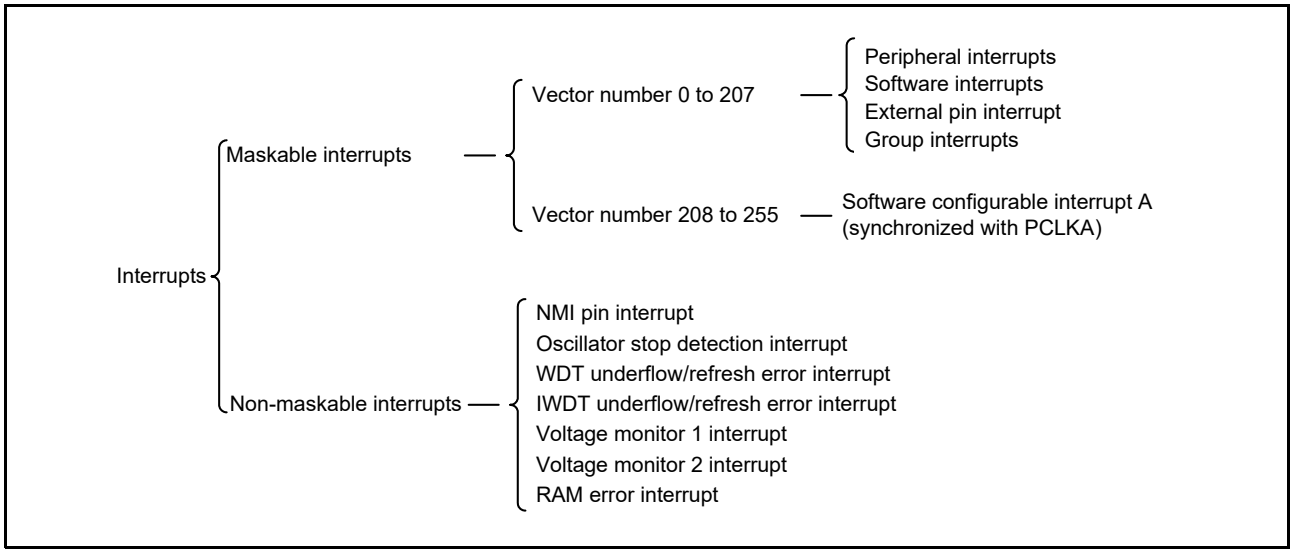


Figure 14.2 Types of Interrupts

14.4.1 Peripheral Interrupts

Peripheral interrupt is generated by peripherals. Peripheral interrupts sources assigned to vector numbers 0 to 207 cannot be assigned to the software configurable interrupts. Refer to section 14.4.5, Software Configurable Interrupts for details on software configurable interrupts.

14.4.2 Software Interrupts

When the SWINTR.SWINT bit and SWINT2R.SWINT2 bit are set to 1, the SWINT interrupt and SWINT2 interrupt occur, respectively.

14.4.3 External Pin Interrupt

An external pin interrupt is generated by signals input to the IRQi pin (i = 0 to 15). Refer to section 14.7.4, Setting the External Pin Interrupt for the procedure to set the external pin interrupt.

14.4.4 Group Interrupts

Multiple peripheral interrupt requests (up to 32 requests) are grouped together as one interrupt request. Interrupts are grouped depending on the peripheral operating clock (ICLK, PCLKB, or PCLKA) and method to detect interrupt requests (edge detection or level detection).

(1) Types of Group Interrupts

Table 14.5 lists types of group interrupts.

Table 14.5 Types of Group Interrupts

Interrupt Vector Number	Interrupt Name	Group Interrupt Source	
		Peripheral operating clock	Interrupt detection method
17	GROUPIE0	ICLK	Edge detection
106	GROUPBE0	PCLKB	
110	GROUPBL0		Level detection
111	GROUPBL1		
107	GROUPBL2		
112	GROUPAL0	PCLKA	
113	GROUPAL1		

(2) Configuration of Group Interrupts

When an interrupt request is generated while the corresponding ENj bit in the group interrupt request enable register (GENIE0, GENBE0, GENBL0, GENBL1, GENBL2, GENAL0, GENAL1*1) is 1, the ISj flag in the group interrupt request register (GRPIE0, GRPBE0, GRPBL0, GRPBL1, GRPBL2, GRPAL0, GRPAL1*1) becomes 1 (j = 0 to 31). Figure 14.3 shows the configuration of group interrupts.

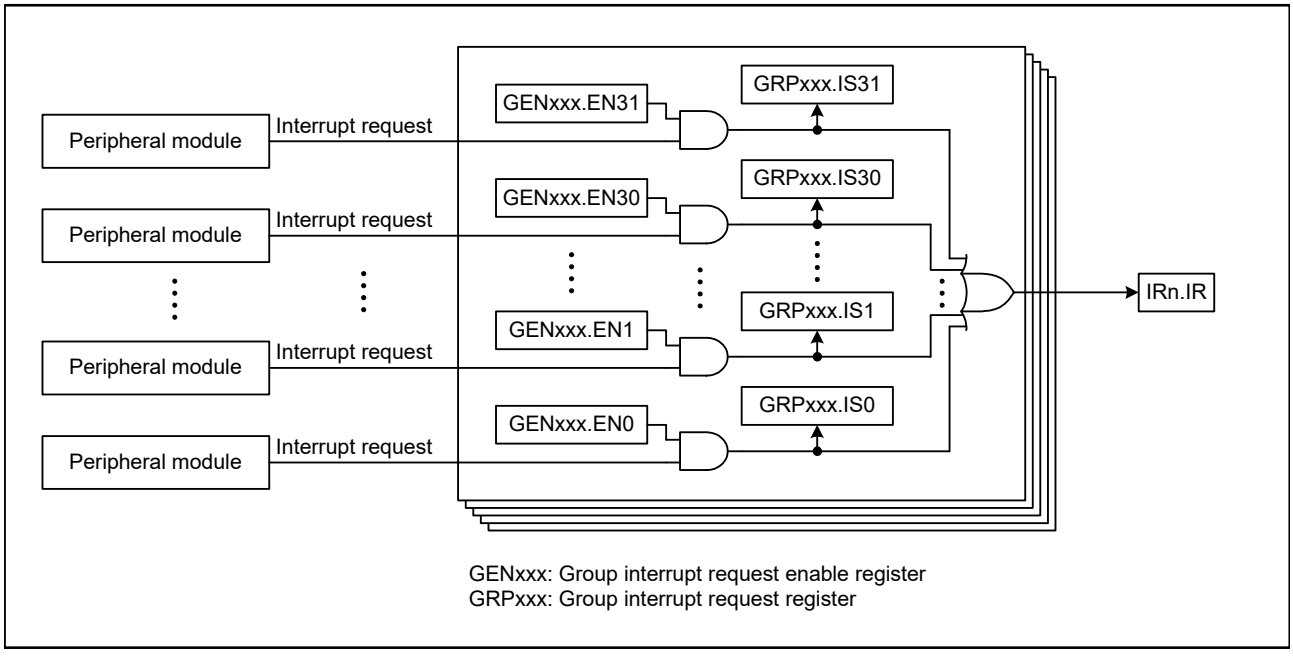


Figure 14.3 Group Interrupt Configuration (n = 17, 106, 107, 110 to 113)

Note 1. There is no register in the group to which no interrupt source is assigned.

(3) Group Interrupt Sources

Table 14.6 lists peripheral interrupt sources that are assigned to group interrupts.

Table 14.6 Group Interrupt Requests (1/3)

Group	No.	Interrupt Request Source	Name	Interrupt Request Enable Bit	Interrupt Status Flag	Interrupt Source Clear Bit	Vector No. (IRn.IR)
IE0	0 to 31	Reserved	—	—	—	—	17
BE0	0 to 31	Reserved	—	—	—	—	106
BL0	0, 1	Reserved	—	—	—	—	110
	2	SCI1	TEI1 (transmission end)	GENBL0.EN2	GRPBL0.IS2	—	
	3		ERI1 (receive error)	GENBL0.EN3	GRPBL0.IS3	—	
	4 to 9	Reserved	—	—	—	—	
	10	SCI5	TEI5 (transmission end)	GENBL0.EN10	GRPBL0.IS10	—	
	11		ERI5 (receive error)	GENBL0.EN11	GRPBL0.IS11	—	
	12	SCI6	TEI6 (transmission end)	GENBL0.EN12	GRPBL0.IS12	—	
	13		ERI6 (receive error)	GENBL0.EN13	GRPBL0.IS13	—	
	14, 15	Reserved	—	—	—	—	
	16	SCI12	TEI12 (transmission end)	GENBL0.EN16	GRPBL0.IS16	—	
	17		ERI12 (receive error)	GENBL0.EN17	GRPBL0.IS17	—	
	18		SCIX0 (Break Field Low width detection)	GENBL0.EN18	GRPBL0.IS18	—	
	19		SCIX1 (Control Field 0 match) (Control Field 1 match) (priority interrupt bit detection)	GENBL0.EN19	GRPBL0.IS19	—	
	20		SCIX2 (bus collision detection)	GENBL0.EN20	GRPBL0.IS20	—	
	21		SCIX3 (valid edge detection)	GENBL0.EN21	GRPBL0.IS21	—	
	22 to 25	Reserved	—	—	—	—	
	26	CAC	FERRI (frequency error)	GENBL0.EN26	GRPBL0.IS26	—	
	27		MENDI (measurement end)	GENBL0.EN27	GRPBL0.IS27	—	
	28		OVFI (overflow interrupt)	GENBL0.EN28	GRPBL0.IS28	—	
	29	DOC	DOPCI (data operation circuit interrupt)	GENBL0.EN29	GRPBL0.IS29	—	
	30, 31	Reserved	—	—	—	—	

Table 14.6 Group Interrupt Requests (2/3)

Group	No.	Interrupt Request Source	Name	Interrupt Request Enable Bit	Interrupt Status Flag	Interrupt Source Clear Bit	Vector No. (IRn.IR)
BL1	0	POEG	POEGGAI (group A interrupt)	GENBL1.EN0	GRPBL1.IS0	—	111
	1		POEGGBI (group B interrupt)	GENBL1.EN1	GRPBL1.IS1	—	
	2		POEGGCI (group C interrupt)	GENBL1.EN2	GRPBL1.IS2	—	
	3		POEGGDI (group D interrupt)	GENBL1.EN3	GRPBL1.IS3	—	
	4 to 7	Reserved	—	—	—	—	
	8	POE	OEI5 (output enable interrupt 5)	GENBL1.EN8	GRPBL1.IS8	—	
	9		OEI1 (output enable interrupt 1)	GENBL1.EN9	GRPBL1.IS9	—	
	10		OEI2 (output enable interrupt 2)	GENBL1.EN10	GRPBL1.IS10	—	
	11		OEI3 (output enable interrupt 3)	GENBL1.EN11	GRPBL1.IS11	—	
	12		OEI4 (output enable interrupt 4)	GENBL1.EN12	GRPBL1.IS12	—	
	13	RIIC0	TEI0 (transmission end)	GENBL1.EN13	GRPBL1.IS13	—	
	14		EEI0 (communication error/ communication event)	GENBL1.EN14	GRPBL1.IS14	—	
	15 to 17	Reserved	—	—	—	—	
	18	S12AD2	S12CMPAI2 (compare interrupt)	GENBL1.EN18	GRPBL1.IS18	—	
	19		S12CMPBI2 (compare interrupt)	GENBL1.EN19	GRPBL1.IS19	—	
	20	S12AD	S12CMPAI (compare interrupt)	GENBL1.EN20	GRPBL1.IS20	—	
	21		S12CMPBI (compare interrupt)	GENBL1.EN21	GRPBL1.IS21	—	
	22	S12AD1	S12CMPAI1 (compare interrupt)	GENBL1.EN22	GRPBL1.IS22	—	
	23		S12CMPBI1 (compare interrupt)	GENBL1.EN23	GRPBL1.IS23	—	
	24	RSCI8	TEI (transmission end)	GENBL1.EN24	GRPBL1.IS24	—	
	25		ERI (receive error)	GENBL1.EN25	GRPBL1.IS25	—	
	26	RSCI9	TEI (transmission end)	GENBL1.EN26	GRPBL1.IS26	—	
	27		ERI (receive error)	GENBL1.EN27	GRPBL1.IS27	—	
	28 to 30	Reserved	—	—	—	—	
	31	RSCI9	BFD (break field detect)	GENBL1.EN31	GRPBL1.IS31	—	
BL2	0	Reserved	—	—	—	—	107
	1	CANFD0	CHEI (channel error)	GENBL2.EN1	GRPBL2.IS1	—	
	2		CFRI (common FIFO receive)	GENBL2.EN2	GRPBL2.IS2	—	
	3	CANFD	GLEI (global error)	GENBL2.EN3	GRPBL2.IS3	—	
	4		RFRI (receive FIFO)	GENBL2.EN4	GRPBL2.IS4	—	
	5	CANFD0	CHTI (channel transmit)	GENBL2.EN5	GRPBL2.IS5	—	
	6	CANFD	RMRI (receive message buffer)	GENBL2.EN6	GRPBL2.IS6	—	
	7 to 31	Reserved	—	—	—	—	
AL0	0 to 11	Reserved	—	—	—	—	112
	12	RSCI11	TEI (transmission end)	GENAL0.EN12	GRPAL0.IS12	—	
	13		ERI (receive error)	GENAL0.EN13	GRPAL0.IS13	—	
	14		BFD (break field detect)	GENAL0.EN14	GRPAL0.IS14	—	
	15	Reserved	—	—	—	—	
	16	RSPI0	SPII0 (idle interrupt)	GENAL0.EN16	GRPAL0.IS16	—	
	17		SPEI0 (error interrupt)	GENAL0.EN17	GRPAL0.IS17	—	
	18 to 21	Reserved	—	—	—	—	
	22	RSPIA0	SPII (idle interrupt)	GENAL0.EN22	GRPAL0.IS22	—	
	23		SPEI (error interrupt)	GENAL0.EN23	GRPAL0.IS23	—	
	24 to 31	Reserved	—	—	—	—	

Table 14.6 Group Interrupt Requests (3/3)

Group	No.	Interrupt Request Source	Name	Interrupt Request Enable Bit	Interrupt Status Flag	Interrupt Source Clear Bit	Vector No. (IRn.IR)
AL1	0 to 12	Reserved	—	—	—	—	113
	13	RI3C0	EEI (communication error/ communication event)	GENAL1.EN13	GRPAL1.IS13	—	
	14 to 31	Reserved	—	—	—	—	

14.4.5 Software Configurable Interrupts

An interrupt source assigned to each interrupt vector number from 208 to 255 can be selected from multiple sources. Figure 14.4 shows the software configurable interrupt configuration.

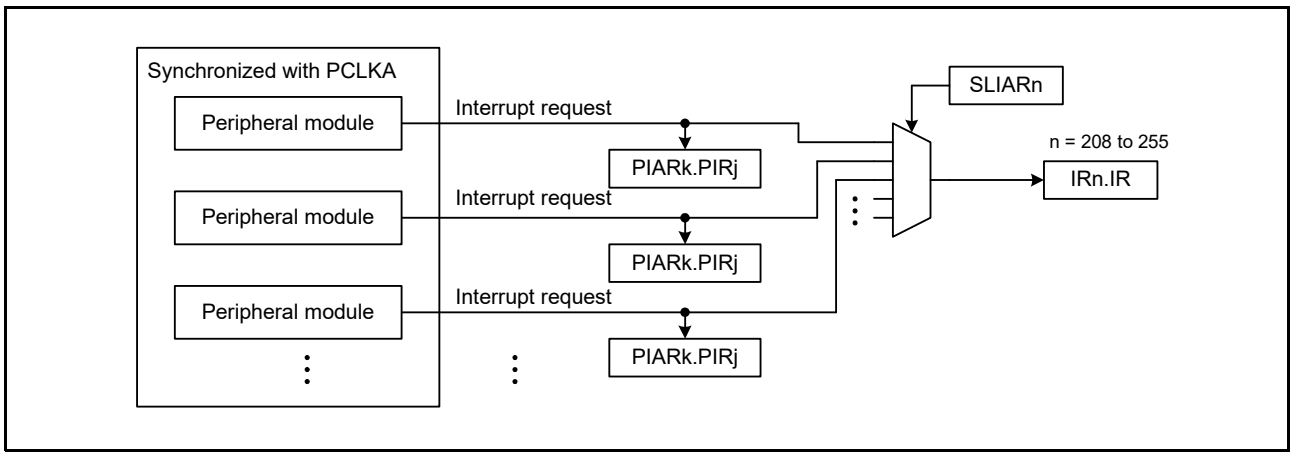


Figure 14.4 Software Configurable Interrupt Configuration

14.4.5.1 Software Configurable Interrupt A

Interrupt sources of peripherals that operates in synchronization with PCLKA can be assigned to interrupt number from 208 to 255. The abbreviation for software configurable interrupt A is PERIA. Interrupt names are indicated by INTA208 to INTA255.

Refer to Table 14.3, Interrupt Sources for Software Configurable Interrupt A for interrupt sources that can be assigned for software configurable interrupt A.

14.4.6 Non-Maskable Interrupts

Non-maskable interrupts include the NMI pin interrupt, oscillation stop detection interrupt, WDT underflow/refresh error interrupt, IWDT underflow/refresh error interrupt, voltage monitoring 1 interrupt, voltage monitoring 2 interrupt, and RAM error interrupt.

Non-maskable interrupts have the highest priority in all interrupts, including the fast interrupt, so are accepted regardless of the settings of the PSW.I bit (interrupt enable) and IPL[3:0] bits (processor interrupt priority level) in the CPU.

The NMISR register can be used to confirm whether a non-maskable interrupt is generated.

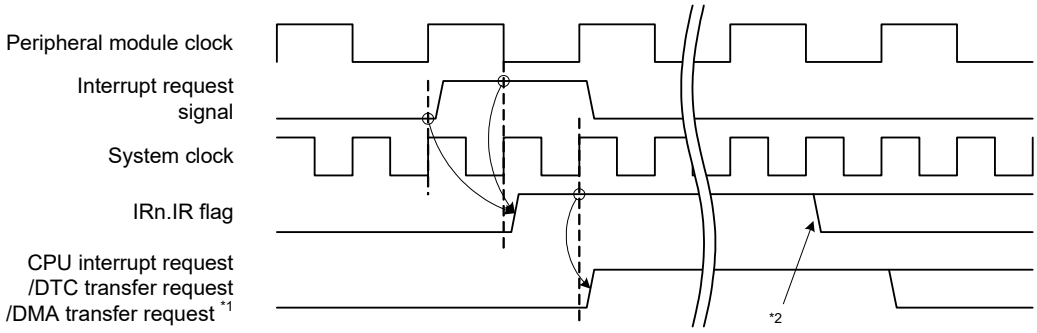
Only the CPU can be selected as the interrupt request destination of non-maskable interrupt. The DTC and DMAC cannot be selected.

14.5 Interrupt Detection

Level detection or edge detection can be used for detecting an interrupt request.
For interrupt requests from the peripherals, edge detection or level detection is fixed for each interrupt source. For interrupt requests of the external pin interrupt, edge detection or level detection can be selected with the IRQCRi.IRQMD[1:0] bits (i = 0 to 15).
Refer to Table 14.4, Interrupt Vector Table for the method to detect each interrupt request.
For group interrupts, interrupt sources are grouped depending on the method to detect interrupt requests.
Interrupt requests by interrupt sources assigned to groups IE0 and BE0 are detected by edge detection. Interrupt requests by interrupt sources assigned to groups BL0, BL1, BL2, AL0, and AL1 are detected by level detection. Note that group interrupts (GROUPIE0, GROUPBE0, GROUPBL0, GROUPBL1, GROUPBL2, GROUPAL0, GROUPAL1) are detected by level detection.
Refer to section 14.4.4, Group Interrupts for group interrupts. Refer to section 14.5.3, Group Interrupts Using Edge Detection and section 14.5.4, Group Interrupts Using Level Detection for interrupt requests of group interrupts.

14.5.1 Edge Detection

Figure 14.5 shows the operation of the IRn.IR flag at edge detection (n = 023 to 255).
The IRn.IR flag becomes 1 when the rising edge of the interrupt request signal is detected. Then, the IRn.IR flag does not become 0 by disabling the interrupt request of the peripheral module. When the CPU accepts the interrupt request or the DTC/DMAC accepts the transfer request, the IRn.IR flag automatically becomes 0. It is not required to set the IRn.IR flag to 0 by software. Refer to Table 14.7, Operations When Starting the DTC/DMAC for details on clearing the IRn.IR flag by DTC/DMAC.
For the external pin interrupts of interrupt vector number 64 to 79 and interrupt sources of interrupt vector number 88 to 95, the timing when the IRn.IR flag becomes 1 after an interrupt signal occurs is different from the other interrupts. For the external pin interrupts, the timing is delayed for internal delay plus two cycles of PCLKB after a signal is input to the IRQ pin (i = 0 to 15). For interrupts of interrupt vector number 88 to 95, the timing is delayed for two cycles of PCLKB.



Note 1. A CPU interrupt request, DTC transfer request, or DMA transfer request is generated. Refer to section 14.7.3.1, Interrupt Request Destination Setting Procedure.
Note 2. For a CPU interrupt request, the IRn.IR flag becomes 0 when the CPU accepts an interrupt request. For the timing for a DTC transfer request or DMA transfer request, refer to Table 14.7, Operations When Starting the DTC/DMAC.

Figure 14.5 IRn.IR Flag Operation for Interrupts Detected by Edge Detection

(1) Detecting Consecutive Interrupt Request Signals

When interrupt request signals occur every cycle, the latter interrupt signal cannot be detected. To accept consecutive interrupt request signals, the interval of at least two cycles of the system clock or peripheral module clock, whichever is lower, is required between interrupt signals. Figure 14.6 shows the interval for accepting consecutive interrupt request signals.

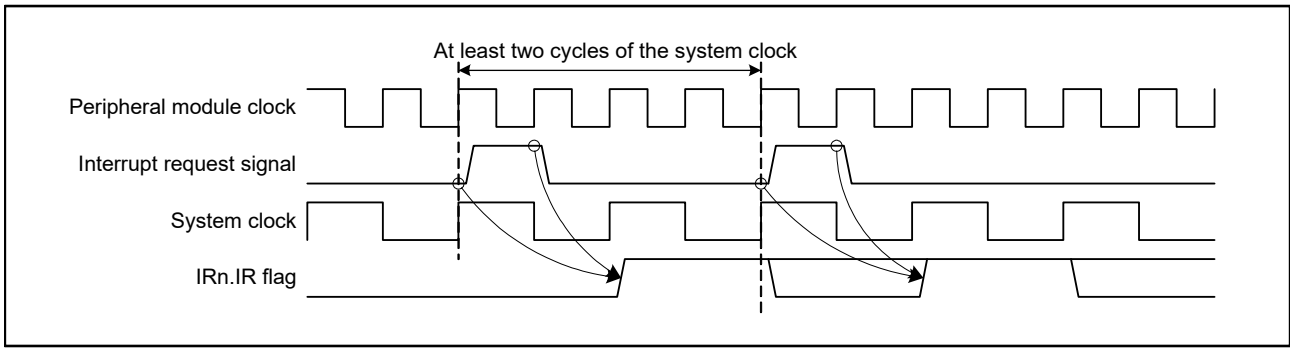


Figure 14.6 Accepting Consecutive Interrupt Signals (when the system clock frequency is lower than the peripheral clock frequency)

When an interrupt request is generated again while the IRn.IR flag is 1, the interrupt request is ignored (n = 023 to 255). However, for transmit interrupt requests, receive interrupt requests, and buffer access interrupt requests of the SCI, RSCI, RIIC, RI3C, RSPI, and RSPIA, when an interrupt request occurs while the IRn.IR flag is 1, the interrupt request is retained in the module. After the IRn.IR flag becomes 0, the IRn.IR flag is set to 1 again by the retained request. Refer to the descriptions for interrupts in each chapter of peripheral modules for details.
Figure 14.7 shows the timing when the IRn.IR flag is set again.

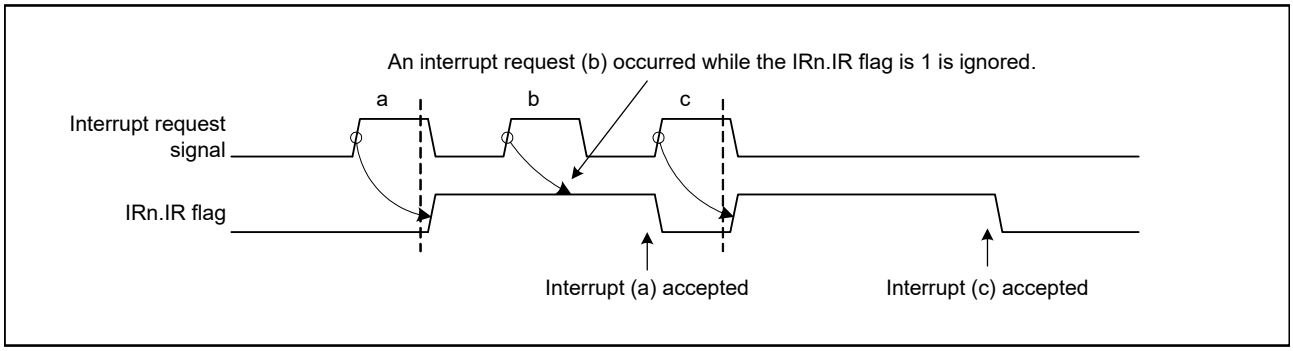


Figure 14.7 Timing to Set the IRn.IR Flag Again

(2) Relation between the IRn.IR flag and Interrupt Request Enable Bits

After the IRn.IR flag becomes 1, the IRn.IR flag does not become 0 even when an interrupt request enable bit in the corresponding peripheral module is set to 0.

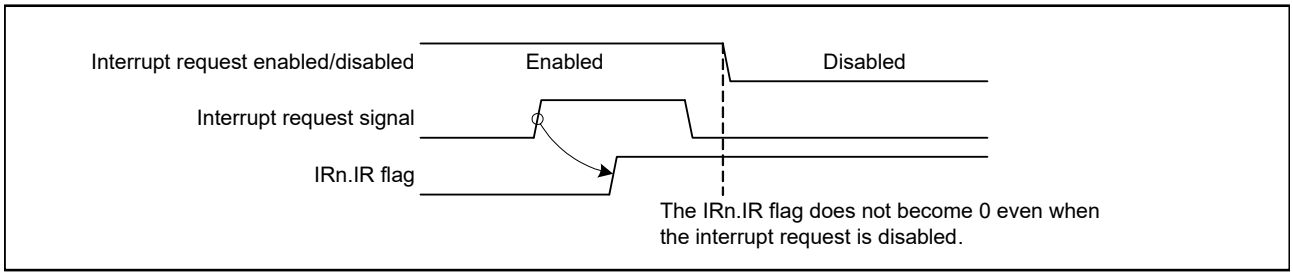


Figure 14.8 Relation Between Disabling the Interrupt Request and the IRn.IR Flag

14.5.2 Level Detection

Figure 14.9 shows operation of the interrupt request signal and the IRn.IR flag for level detection (n = 016 to 124). The IRn.IR flag is 1 while the interrupt request signal is 1. To set the IRn.IR flag to 0, set the corresponding interrupt request signal of the peripheral module to 0. Set the corresponding interrupt status flag of the peripheral module to 0 and wait for the time until the value is reflected in the IRn.IR flag before exiting the interrupt handler. Refer to (2) Notes on writing to I/O registers in section 5, I/O Registers for details on waiting for the reflection.

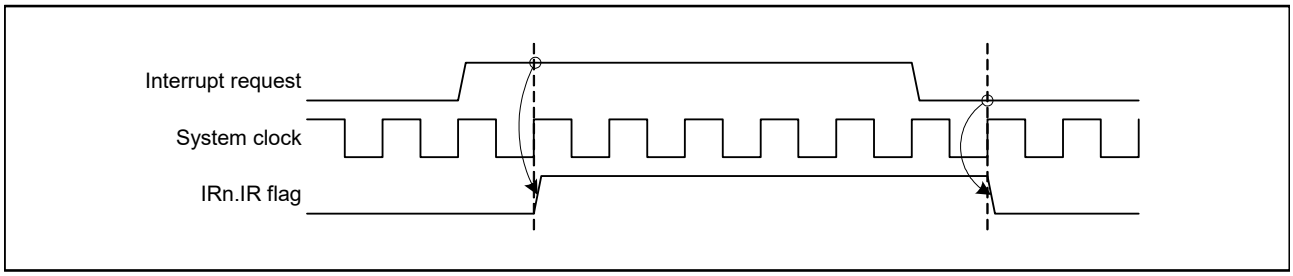


Figure 14.9 IRn.IR Flag Operation for Level Detection

Figure 14.10 shows an example of the procedure to handle interrupts for level detection.

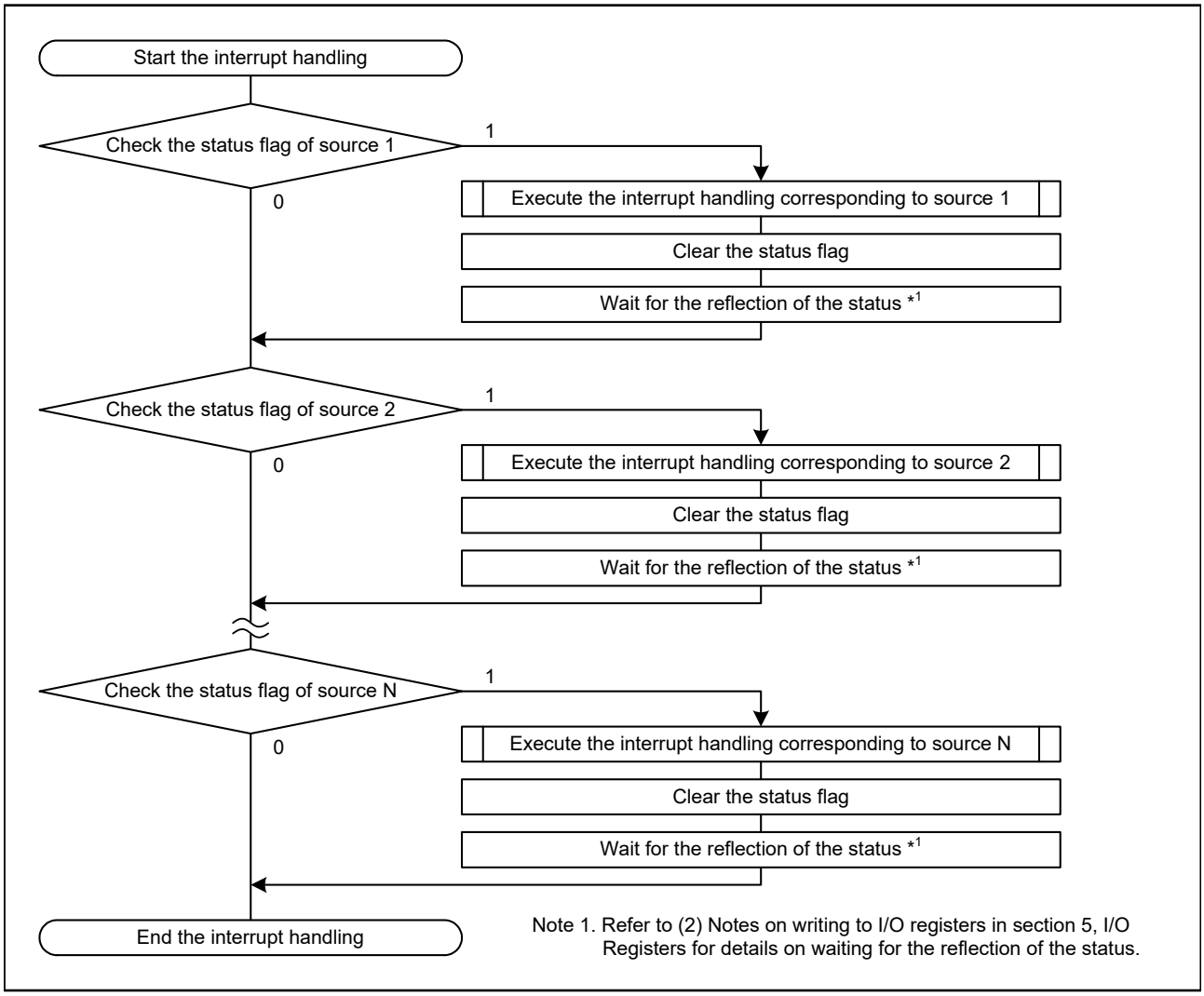


Figure 14.10 Example of Level Detection Interrupt Handling Procedure (N indicates the number of status flags)

14.5.3 Group Interrupts Using Edge Detection

Groups IE0 and BE0 of group interrupts include interrupt sources that are detected by edge detection.
While the IR017.IR flag corresponding to the GROUPIE0 interrupt and the IR106.IR flag corresponding to the GROUPBE0 interrupt become 1 under the same conditions as edge detection, the IR017.IR and IR106.IR flags become 0 under the same conditions as level detection.
When the rising edge of an interrupt request signal is detected while the corresponding GENIE0/GENBE0.ENj bit is 1, both the GRPIE0/GRPBE0.ISj flag and IR017/IR106.IR flag become 1 (j = 0 to 31). Then, GRPIE0/GRPBE0.ISj flag and IR017/IR106.IR flag do not become 0 by disabling the interrupt request of the peripheral module or setting the GENIE0/GENBE0.ENj bit to 0.
When the GCRIE0/GCRBE0.CLRj bit is set to 1, the GRPIE0/GRPBE0.ISj flag becomes 0, and consequently the IR017/IR106.IR flag becomes 0.
Figure 14.11 and Figure 14.12 show operation examples of group interrupts using edge detection. Figure 14.13 shows an example of operation when interrupt requests are generated by multiple interrupt sources in the same group.

Note: • There is no register in the group to which no interrupt source is assigned.

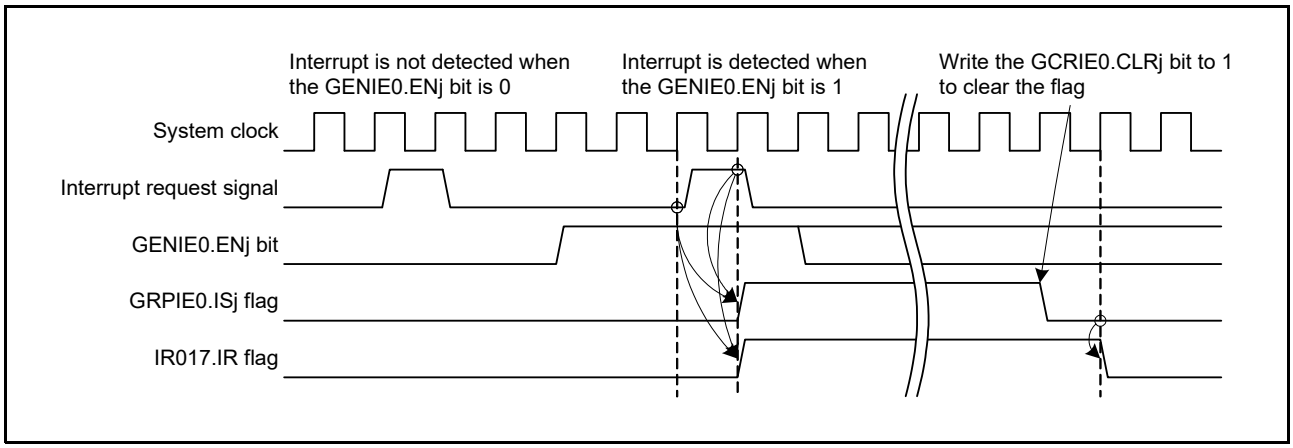


Figure 14.11 Example of Interrupt Request for Group Interrupt Using Edge Detection (Group IE0)

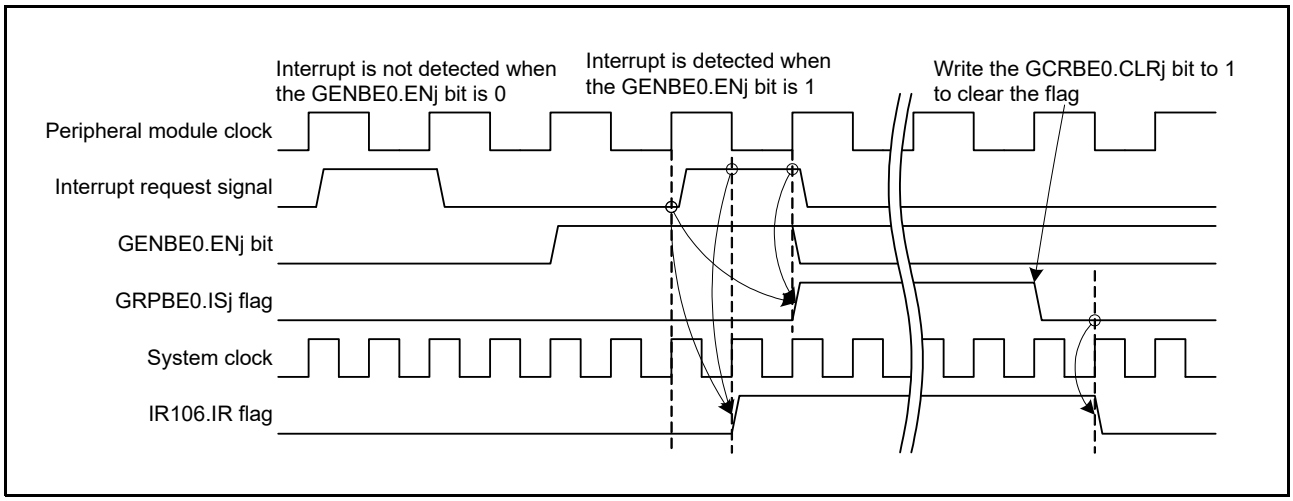


Figure 14.12 Example of Interrupt Request for Group Interrupt Using Edge Detection (Group BE0)

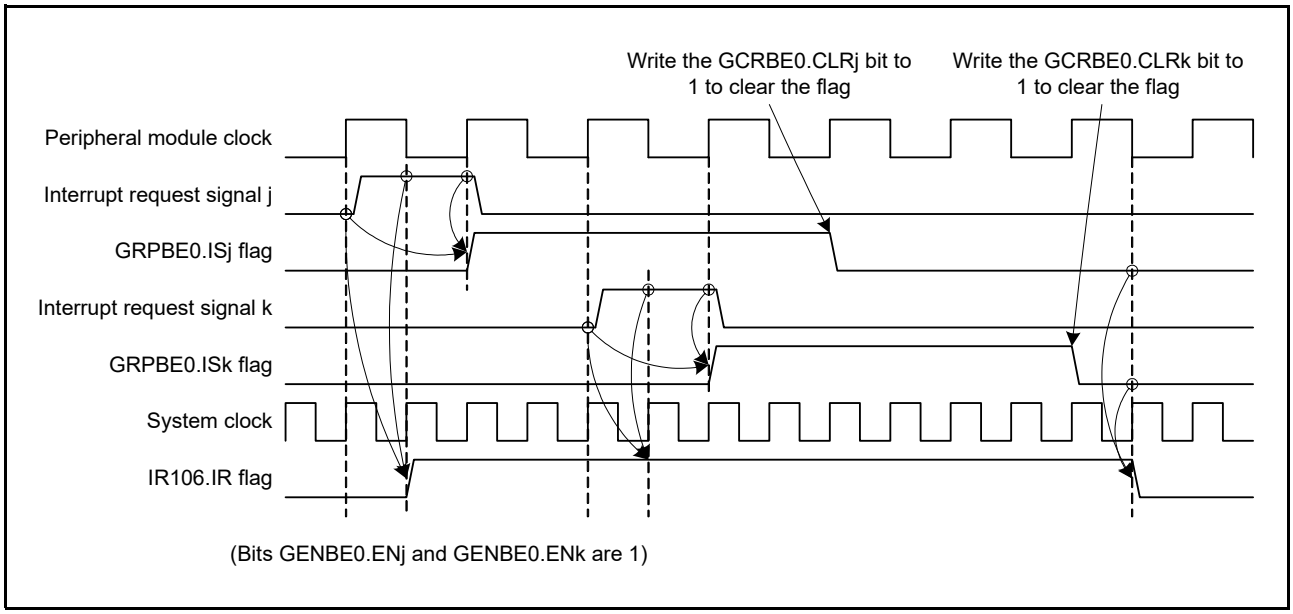


Figure 14.13 Example of Operation When Multiple Edge Detection Interrupt Requests Are Generated in the Same Group (Group BE0)

Figure 14.14 and Figure 14.15 show examples of the procedure to handle group interrupts using edge detection.

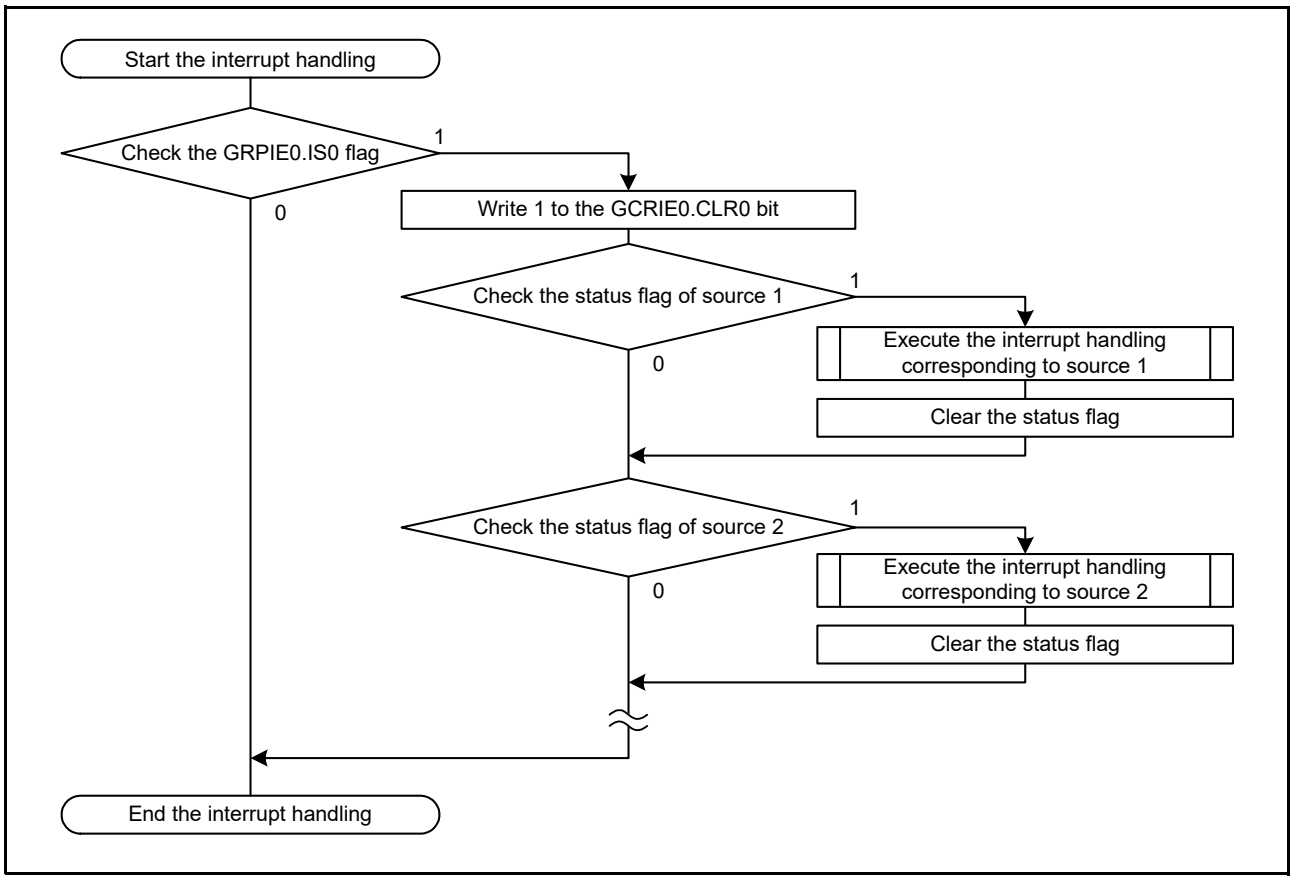


Figure 14.14 Example of Procedure to Handle Group Interrupts Using Edge Detection (Group IE0)

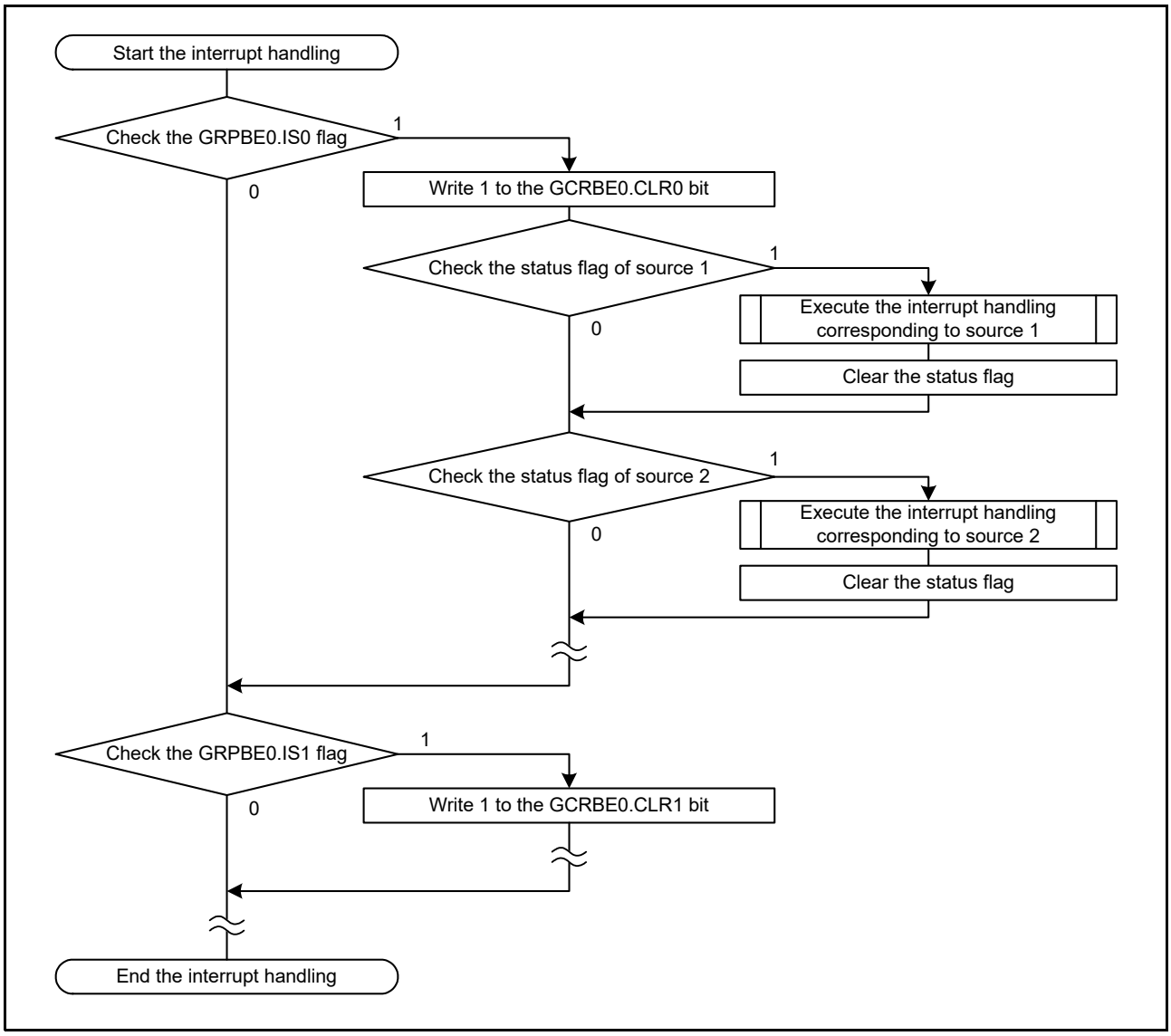


Figure 14.15 Example of Procedure to Handle Group Interrupts Using Edge Detection (Group BE0)

14.5.4 Group Interrupts Using Level Detection

Groups BL0, BL1, BL2, AL0, and AL1 of group interrupts includes interrupt sources that are detected by level detection. The IR110.IR flag corresponding to the GROUPBL0 interrupt, the IR111.IR flag corresponding to the GROUPBL1 interrupt, the IR107.IR flag corresponding to the GROUPBL2 interrupt, the IR112.IR flag corresponding to the GROUPAL0 interrupt, and the IR113.IR flag corresponding to the GROUPAL1 interrupt change under the same conditions as level detection.

When an interrupt signal becomes 1 while the corresponding GENBL0/GENBL1/GENBL2/GENAL0/GENAL1.ENj bit is 1, the GRPBL0/GRPBL1/GRPBL2/GRPAL0/GRPAL1.ISj flag and the IRn.IR flag become 1 (j = 0 to 31). Then, the GRPBL0/GRPBL1/GRPBL2/GRPAL0/GRPAL1.ISj flag and IRn.IR flag becomes 0 when the corresponding interrupt request signal becomes 0. Also, when the GENBL0/GENBL1/GENBL2/GENAL0/GENAL1.ENj bit is set to 0, the corresponding GRPBL0/GRPBL1/GRPBL2/GRPAL0/GRPAL1.ISj flag and IRn.IR flag become 0.

Figure 14.16 shows an operation example of group interrupts using edge detection. Figure 14.17 shows an example of operation when interrupt requests are generated by multiple interrupt sources in the same group.

Note: • There is no register in the group to which no interrupt source is assigned.

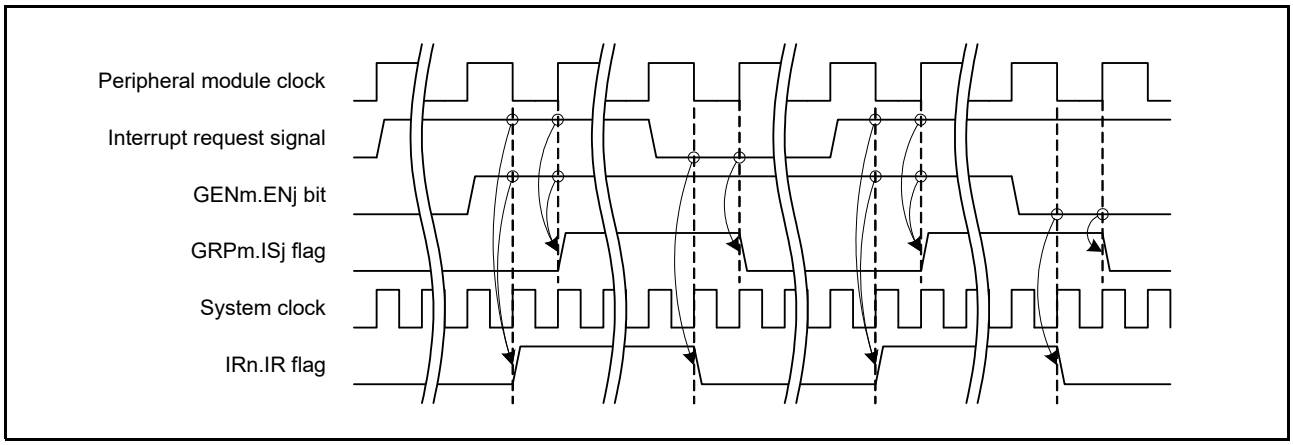


Figure 14.16 Operation Example of Group Interrupt Using Level Detection (m = BL0, BL1, BL2, AL0, AL1)

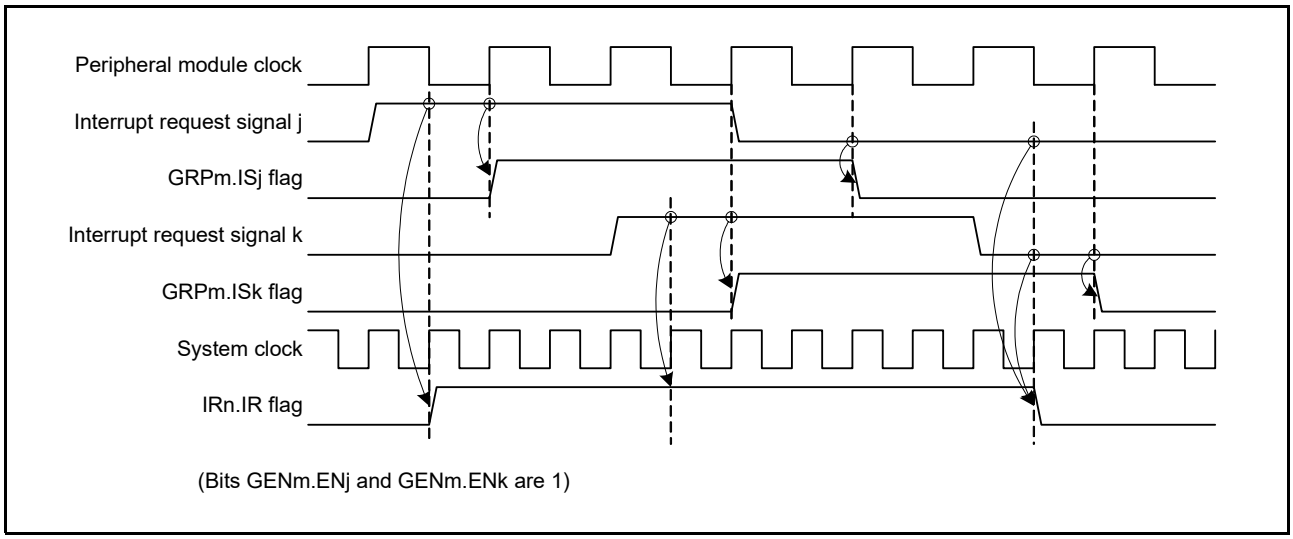


Figure 14.17 Operation Example When Multiple Interrupt Requests are Generated in the Same Group (m = BL0, BL1, BL2, AL0, AL1)

Figure 14.18 shows the procedure to handle group interrupts for level detection.

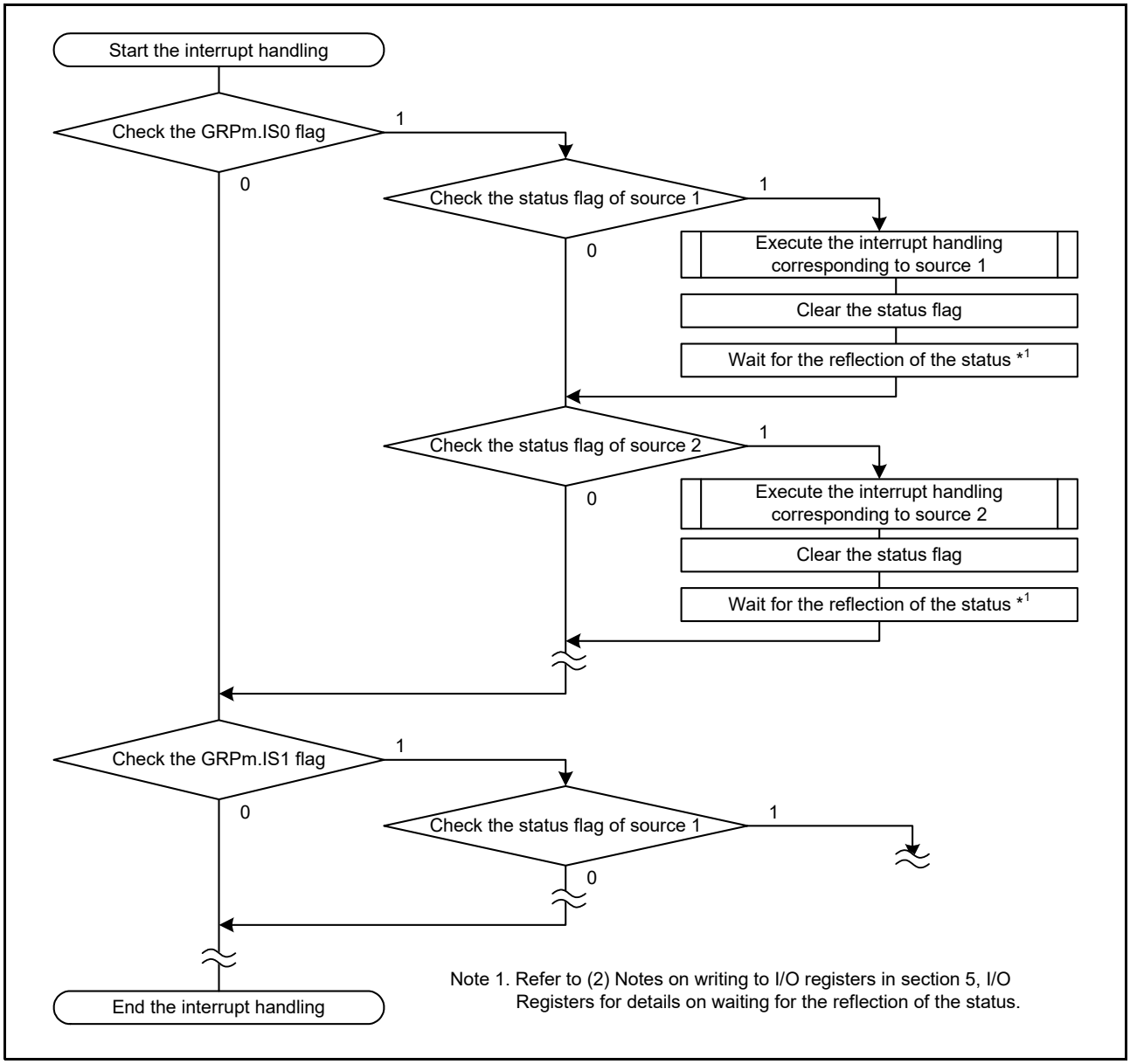


Figure 14.18 Example of Group Interrupt Handling Procedure for Level Detection (m = BL0, BL1, BL2, AL0, AL1)

14.5.5 Software Configurable Interrupts

Interrupt sources and interrupt requests for software configurable interrupts are detected by edge detection.
Figure 14.19 shows an operation example of the interrupt request and interrupt status flag for software configurable interrupts.

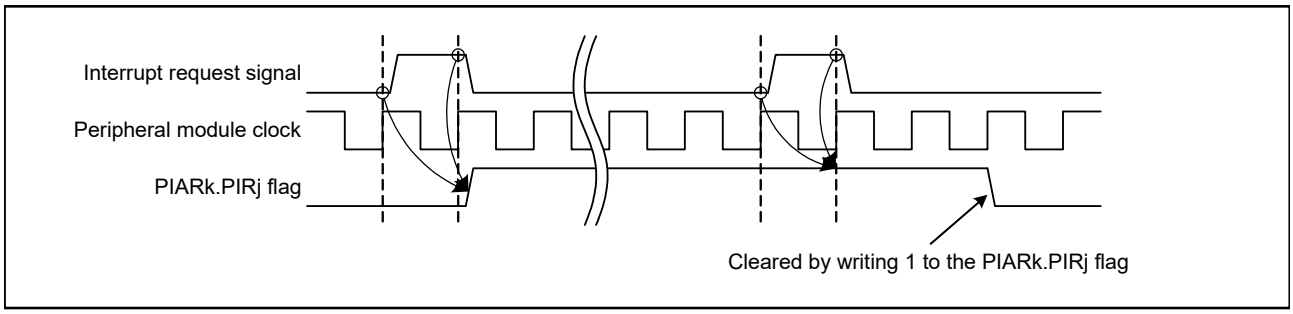


Figure 14.19 Operation Example of the Interrupt Request and Interrupt Status Flag for Software Configurable Interrupts

14.6 Determining Priority of Interrupt Requests

The ICU determines the priority for each interrupt request destination. The priority for each interrupt request destination is determined as follows.

(1) Determining Priority when the CPU is the Interrupt Request Destination

A source selected for the fast interrupt has the highest priority. Next to the fast interrupt, the priority is determined by the IPRr.IPR[3:0] bit value, and an interrupt source with a larger value has priority (r = 000 to 255). If multiple sources has the same IPRr.IPR[3:0] bit value, the priority is determined by the interrupt vector number, and the source with the smaller number has priority.

(2) Determining Priority when the DTC is the Interrupt Request Destination

The IPRr.IPR[3:0] bits have no effect (r = 000 to 255). The priority is determined by only the interrupt vector number, and an interrupt source with a smaller number has priority.

(3) Determining Priority when the DMAC is the Interrupt Request Destination

The IPRr.IPR[3:0] bits have no effect. The priority is determined by the DMAC channel number. Refer to section 17, DMA Controller (DMACAa) for details on the DMAC channel priority.

14.7 Interrupt Setting Procedure

14.7.1 Enabling Interrupt Requests

The following describes the procedure to enable interrupt requests.

(1) Set the interrupt request enable bit of the peripheral modules to enable output of the interrupt request.
(2) For group interrupts, set the corresponding ENj bit in the group interrupt request enable register to 1 to enable output of the interrupt request to the ISj flag in the group interrupt request register (j = 0 to 31).
(3) Set the corresponding IERm.IENj bit to 1 to enable output of the interrupt request to the interrupt request destination (m = 02h to 1Fh; j = 0 to 7).

After the above procedure is completed, when a peripheral interrupt occurs, the IRn.IR flag corresponding to the interrupt source becomes 1 (n = 016 to 255).
For group interrupts, the ISj flag in the group interrupt request register becomes 1, the IRn.IR flag corresponding to the group becomes 1, and an interrupt request is output to the interrupt request destination.
When the IERm.IENj bit is 0, the interrupt request corresponding to the interrupt source is not output to the interrupt request destination.

14.7.2 Disabling Interrupt Requests

The following describes the procedure to disable interrupt requests.

(1) Set the corresponding IERm.IENj bit to 0 (m = 02h to 1Fh; j = 0 to 7).
(2) For group interrupts, set the corresponding ENj bit in the group interrupt request enable register to 0 to disable output of the interrupt request to the ISj flag in the group interrupt request register (j = 0 to 31).
(3) Set the interrupt request enable bit of the peripheral modules to disable output of the interrupt request. Read the register that has been set to confirm that the value is reflected.
(4) As needed, read the IRn.IR flag or set the IR flag to 0.*1
For group interrupts, confirm that the ISj flag in the group interrupt request register is 0 or set the ISj flag to 0.

Note 1. When disabling the transmit interrupt request, receive interrupt request, or buffer access interrupt requests of the SCI, RSCI, RIIC, RI3C, RSPI, or RSPIA, set the IRn.IR flag to 0 according to the above procedure. Refer to the description of interrupts in the corresponding section of peripheral modules for details.

14.7.3 Selecting Interrupt Request Destination

14.7.3.1 Interrupt Request Destination Setting Procedure

The destination of an interrupt request can be selected from the CPU, DTC, or DMAC for each interrupt source. Destinations that can be selected differ depending on the interrupt source. Refer to Table 14.4, Interrupt Vector Table for details on the destinations. Do not select a destination that is not indicated as "✓" in Table 14.4.
When set the external pin interrupt as the DTC or DMAC trigger, set the IRQCRi.IRQMD[1:0] bits to select edge detection (i = 0 to 15).
The following describes the procedure to select a destination of an interrupt request.

(1) Setting Interrupt Sources as the DMAC trigger

Perform the following settings while the IERm.IENj bit is 0 of the interrupt source that is selected as the DMAC trigger (m = 02h to 1Fh; j = 0 to 7).

(1) Set the interrupt vector number of the interrupt source used as the DMAC trigger in the DMRSRm register corresponding to the DMAC channel (m = DMAC channel number).*1
(2) Set the DMTMD.DCTG[1:0] bits corresponding to the DMAC channel to 01b in order to select the peripheral interrupt or external pin interrupt as the DMAC trigger.
(3) Set the DMCNT.DTE bit corresponding to the DMAC channel to 1.

After the above settings are completed, set the corresponding IERm.IENj bit to 1.
Also, set the DMAST.DMST bit to 1 before or after the above settings.
Refer to section 17.3.7, Activating the DMAC in section 17, DMA Controller (DMACAa) for the procedure to set the DMAC.

(2) Setting Interrupt Sources as the DTC trigger

Perform the following setting while the IERm.IENj bit of the interrupt source that is selected as the DTC trigger is 0.

(1) Set the DTCERn.DTCE bit corresponding to the interrupt vector number n used for the DTC trigger to 1 (n = 026 to 255).*1

After the above setting is completed, set the IERm.IENj bit to 1.
Also set the DTCST.DTCST bit to 1 before or after the above settings.
Refer to section 18.5, DTC Setting Procedure in section 18, Data Transfer Controller (DTCb) for the procedure to set the DTC.

Note 1. Do not set the same interrupt source as DTC and DMAC triggers. Also, do not set the same interrupt source as triggers of multiple DMAC channels.

(3) Setting Interrupt Sources for the CPU

When an interrupt source is not selected as the DTC or DMAC trigger, the interrupt request is output to the CPU.
Set the IERm.IENj bit to 1 while the interrupt source is not selected as a DTC or DMAC trigger.

14.7.3.2 Operations When the DTC/DMAC Selected

Table 14.7 lists operations when the DTC or DMAC is set as an interrupt request destination.

Table 14.7 Operations When Starting the DTC/DMAC

Interrupt Request Destination	DISEL *1	Number of Remaining Transfers	Operation per Request	IR Flag Clear Timing *2	Interrupt Request Destination after Transfer
DTC *3	1	≠ 0	DTC transfer → CPU interrupt	Cleared when the CPU accepts an interrupt request.	DTC
		= 0	DTC transfer → CPU interrupt	Cleared when the CPU accepts an interrupt request.	CPU (DTCERn.DTCE bit becomes 0)
	0	≠ 0	DTC transfer	Cleared when the DTC starts data transfer.	DTC
		= 0	DTC transfer → CPU interrupt *4	Cleared when the CPU accepts an interrupt request. *4	CPU (DTCERn.DTCE bit becomes 0)
DMAC	1	≠ 0	DMA transfer → CPU interrupt	Cleared when the CPU accepts an interrupt request.	DMAC
		= 0	DMA transfer → CPU interrupt	Cleared when the CPU accepts an interrupt request.	CPU (DMACm.DMCNT.DTE bit becomes 0)
	0	≠ 0	DMA transfer	Cleared when the DMAC starts data transfer.	DMAC
		= 0	DMA transfer *4	Cleared when the DMAC starts data transfer. *4	CPU (DMACm.DMCNT.DTE bit becomes 0)

Note 1. For the DTC, set the DTC.MRB.DISEL bit; For the DMAC, set the DMACm.DMCSL.DISEL bit.
Note 2. When the IRn.IR flag is 1, an interrupt request (DTC or DMA transfer request) that is generated again is ignored.
Note 3. For chain transfer, the DTC transfer continues until the chain transfer ends. After chain transfer ends, whether a CPU interrupt occurs, the IRn.IR flag clear timing, and the interrupt request destination differ depending on the DISEL bit value the number of remaining transfers. For the chain transfer, refer to Table 18.4, Chain Transfer Conditions in section 18, Data Transfer Controller (DTCb).
Note 4. When the DISEL bit is 0 and the number of remaining transfers is 0, operations in the DTC and DMAC are different.

14.7.3.3 Changing the Interrupt Request Destination

Set the IERm.IENj bit to 0 before changing the interrupt request destination (m = 02h to 1Fh; j = 0 to 7).

(1) When the current interrupt request destination is the DMAC

To change interrupt request destinations or change the DMAC trigger to another interrupt source while the DMA transfer is not completed (DMCNT.DTE bit is not cleared) after the procedure described in (1) Setting Interrupt Sources as the DMAC trigger of section 14.7.3.1, Interrupt Request Destination Setting Procedure, follow the procedure below.

(1) Set the IERm.IENj bit of both the current and new triggers to 0.
(2) Check the DMAC transfer status. If transfer is not completed, wait until the completion of transfer.
(3) Perform the procedure described in 14.7.3.1 Interrupt Request Destination Setting Procedure.

(2) When the current interrupt request destination is the DTC

To change interrupt request destinations or change the DTC transfer information while the DTC transfer is not completed (the DTCERn.DTCE bit is not cleared) after the procedure described in (2) Setting Interrupt Sources as the DTC trigger of section 14.7.3.1, Interrupt Request Destination Setting Procedure, follow the procedure below (n = 026 to 255).

(1) Set the IERm.IENj bit of both the current and new triggers to 0 (m = 02h to 1Fh; j = 0 to 7).
(2) Check the DTC transfer status. If transfer is not completed, wait until the completion of transfer.
(3) Perform the procedure described in 14.7.3.1 Interrupt Request Destination Setting Procedure.

14.7.4 Setting the External Pin Interrupt

The following describes the procedure to use the external pin interrupt.

(1) Set the IERm.IENj bit corresponding to the IRQi pin to 0 (interrupt request is disabled) (m = 02h to 1Fh; j = 0 to 7; i = 0 to 15).
(2) Set the IRQFLTE0.FLTENi or IRQFLTE1.FLTENi bit to 0 (digital filter is disabled).
(3) Set the IRQFLTC0.FCLKSELi[1:0] or IRQFLTC1.FCLKSELi[1:0] bits to select the sampling clock of the digital filter.
(4) Set the I/O port and confirm the setting.
(5) Set the IRQCRi.IRQMD[1:0] bits to select the detection method.
(6) When edge detection is selected, set the corresponding IRn.IR flag to 0 (n = 016 to 255).
(7) Set the IRQFLTE0.FLTENi or IRQFLTE1.FLTENi bit to 1 (digital filter is enabled).
(8) To select the DTC as the interrupt request destination, set the DTCERn.DTCE bit. To select the DMAC as the interrupt request destination, set the DMRSRm register. When neither the DTCERn.DTCE bit nor the DMRSRm register is set, the interrupt request is sent to the CPU (m = DMAC channel number; n = 026 to 255).
(9) Set the corresponding IERm.IENj bit to 1 (interrupt request is enabled).

14.7.5 Setting Non-Maskable Interrupts

After reset, non-maskable interrupts are disabled. To use non-maskable interrupts, follow the procedure below.

(1) Set the stack pointer (SP).
(2) When using the NMI pin, set the NMIFLTE.NFLTEN bit to 0 (digital filter is disabled).
(3) When using the NMI pin, set the NMIFLTC.NFCLKSEL[1:0] bits to select the sampling clock of the digital filter.
(4) When using the NMI pin, set the NMICR.NMIMD bit to select the edge for detection.
(5) When using the NMI pin, write 1 to the NMICLR.NMICLR bit to set the NMISR.NMIST flag to 0.
(6) When using the NMI pin, set the NMIFLTE.NFLTEN bit to 1 (digital filter is enabled).
(7) To enable generation of the non-maskable interrupt, set the bit in the NMIER register corresponding to the used interrupt source to 1.

Once a bit in the NMIER register is set to 1 (enabled), the bit cannot be rewritten so it cannot be set to 0 (disabled). To disable a non-maskable interrupt that has been enabled, reset the MCU.
Refer to section 13, Exception Handling for details on the flow of non-maskable interrupt handling.
Excluding the RAMST flag, each flag in the NMISR register becomes 0 by writing 1 to the corresponding bit in the NMICLR register. To set the RAMST flag to 0, set the RAM.RAMSTS.RAMERR flag that has become 1 to 0.
Confirm that all flags in the NMISR register are 0 before exiting the interrupt handler of non-maskable interrupts.
Non-maskable interrupts, excluding the NMI pin interrupt, can be used as a maskable interrupt. When using as a maskable interrupt, do not change the NMIER register value from the value after reset. In addition, set the LVD1CR1.LVD1IRQSEL bit and LVD2CR1.LVD2IRQSEL bit to 1 when using voltage monitoring 1 interrupt and voltage monitoring 2 interrupt as a maskable interrupt.

14.7.6 Digital Filter

Noise included in signals input to pins IRQi and NMI can be reduced by enabling the digital filter (i = 0 to 15).
The digital filter samples signals input to pins using the sampling clock (PCLKB, PCLKB/8, PCLKB/32, PCLKB/64) for the digital filter, and passes the input signal only when three consecutive sampled signals are the same level.
When using the digital filter for the IRQi pin, refer to section 14.7.4, Setting the External Pin Interrupt to set the associated registers. When using the digital filter for the NMI pin, refer to section 14.7.5, Setting Non-Maskable Interrupts to set the associated registers.
When the external pin interrupt or the NMI pin interrupt is used as a source to exit software standby mode, the digital filter cannot be used. Set the IRQFLTE0.FLTENi bit, IRQFLTE1.FLTENi bit or NMIFLTE.NFLTEN bit to 0 before entering software standby mode. To enable the digital filter again, set the IRQFLTE0.FLTENi bit, IRQFLTE1.FLTENi bit, or NMIFLTE.NFLTEN bit to 1.
Figure 14.20 shows an example of digital filter operation.

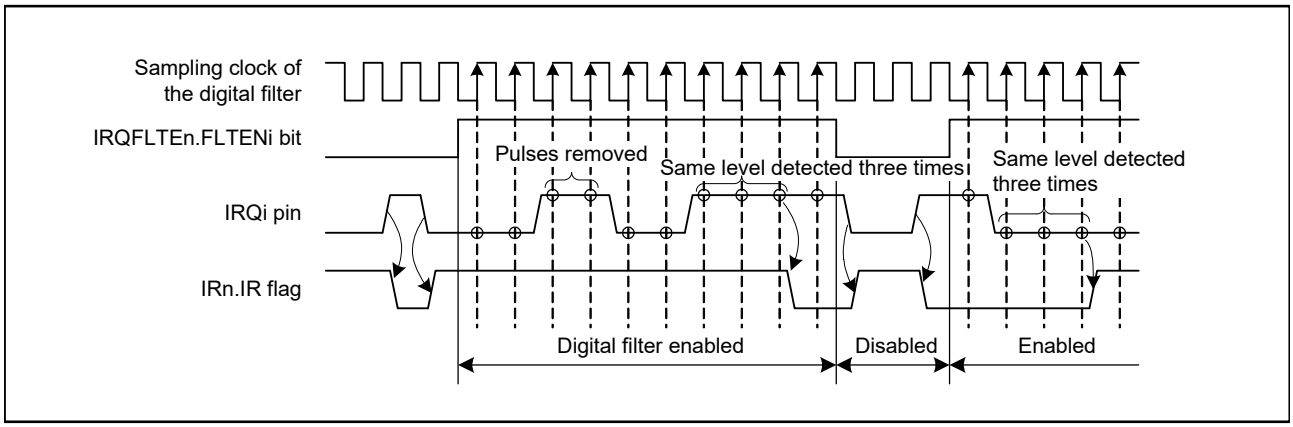


Figure 14.20 Digital Filter Operation Example (when the IRQCRi.IRQMD[1:0] bits are 00b (low level))

14.7.7 Setting Software Configurable Interrupts

The following describes the procedure to assign interrupt sources to software configurable interrupts.

(1) Set the IERm.IENj bit to 0 (m = 02h to 1Fh; j = 0 to 7). This setting is not required when the value does not change from the value after reset.
(2) Set the interrupt source number in the SLIARn (n = 208 to 255) register. Refer to Table 14.3, Interrupt Sources for Software Configurable Interrupt A for details on interrupt source numbers that are assigned to software configurable interrupt A.
(3) Set the SLIPRCR.WPRC bit to 1.
(4) Confirm that the SLIPRCR.WPRC bit is 1.
(5) Select the interrupt request destination from the CPU, DTC, or DMAC. Refer to section 14.7.3.1, Interrupt Request Destination Setting Procedure for details on the setting procedure.
(6) Write 0 to the IRn.IR flag only when edge detection is selected (n = 208 to 255).
(7) Set the IERm.IENj bit to 1.

14.7.7.1 Polling for Software Configurable Interrupts

When polling an interrupt request by reading the PIARk.PIRj (k = 0h to Fh, 12h to 14h), follow the procedure below (j = 0 to 7).

(1) Set the peripheral interrupt used.
(2) Clear the PIARk.PIRj flag for polling by writing 1 to the flag.*1
(3) Enable output of the peripheral interrupt request.
(4) As needed, read the PIARk.PIRj flag to check the value.
(5) When clearing the PIARk.PIRj flag, write 1 to the targeted flag.*1
(6) As needed, repeat step (4) and (5).

Note 1. Do not use bit manipulation instructions. Multiple status flags may be cleared if a bit manipulation instruction is used. To clear a flag, write the PIARk register in 8-bit units as follows: set the flag that is to be cleared to 1 and set the other flags to 0.

14.8 Multiple Interrupt

To enable another interrupt while processing an interrupt (multiple interrupt), set the PSW.I bit to 1 (interrupt enabled) in the interrupt handler of an accepted interrupt.
The PSW.IPL[3:0] bits in the interrupt handler are the same value as the priority level of the accepted interrupt request. In this case, when an interrupt request with the higher priority level than the PSW.IPL[3:0] bit value is generated, the interrupt request is accepted.
The PSW.I bit can be rewritten only in supervisor mode. Since the PSW.PM bit becomes 0 (supervisor mode is selected) when an interrupt is accepted, the PSW.I bit can be rewritten in the interrupt handler.

14.9 Fast interrupt

The fast interrupt is an interrupt that the CPU can respond to fast. Only one interrupt source can be assigned to the fast interrupt.
The priority level of the fast interrupt is 15 (highest) regardless of the IPRr.IPR[3:0] bit setting (r = 000 to 255). Also, the fast interrupt has higher priority than the other interrupt sources of which the priority level is 15. Note that the fast interrupt cannot be accepted when the PSW.IPL[3:0] bits are 1111b (priority level 15).
To assign an interrupt source to the fast interrupt, set the FIR.FVCT[7:0] bits to select the vector number of the interrupt source, and set the FIR.FIEN bit to 1 (fast interrupt is enabled).
The fast interrupt is enabled only when the CPU is selected as the interrupt request destination. When the DTC or DMAC is selected as the destination, the fast interrupt is disabled.
Refer to section 2, CPU and section 13, Exception Handling for details on the fast interrupt.

14.10 Exiting Low Power Consumption State

Interrupts can be used for exiting sleep mode, all-module clock stop mode, and software standby mode.
Refer to section 11, Low Power Consumption for details. This section describes the procedure to set an interrupt source for exiting each low power consumption mode.

14.10.1 Exiting Sleep Mode

Non-maskable interrupts and all interrupt sources can be used for exiting sleep mode. The following conditions must be satisfied.

(1) Non-maskable interrupts

- Generation of the interrupt that is used for exiting is enabled in the NMIER register.

(2) Interrupts

- The CPU is selected as the interrupt request destination.
- The interrupt request is enabled by the IERm.IENj bit (m = 02h to 1Fh; j = 0 to 7).
- The priority level is higher than the PSW.IPL[3:0] bit value in the CPU.
- For group interrupts, the interrupt request is enabled by the corresponding ENj bit in the group interrupt request enable register (GENBL0, GENBL1, GENBL2, GENAL0, GENAL1) (j = 0 to 31).

14.10.2 Exiting All-Module Clock Stop Mode

Non-maskable interrupts and interrupt sources that have a "✓" in the Exit from ACS column in Table 14.4, Interrupt Vector Table can be used for exiting all-module clock stop mode. The conditions below must be satisfied.

(1) Non-maskable interrupts

- Generation of the interrupt that is used for exiting is enabled in the NMIER register.

(2) Interrupts

- The interrupt source can be used for exiting all-module clock stop mode.
- The CPU is selected as the interrupt request destination.
- The interrupt request is enabled by the IERm.IENj bit (m = 02h to 1Fh; j = 0 to 7).
- The priority level is higher than the PSW.IPL[3:0] bit value in the CPU.

14.10.3 Exiting Software Standby Mode

Non-maskable interrupts (excluding oscillation stop detection interrupt) and interrupt sources that have a "✓" in the Exit from SSBY column in Table 14.4, Interrupt Vector Table can be used for exiting software standby mode. The conditions below must be satisfied.

(1) Non-maskable interrupts

- Generation of the interrupt that is used for exiting is enabled in the NMIER register.
- When using the NMI pin interrupt, the digital filter is disabled.

(2) Interrupts

- The interrupt source can be used for exiting software standby mode.
- The CPU is selected as the interrupt request destination.
- The interrupt request is enabled by the IERm.IENj bit (m = 02h to 1Fh; j = 0 to 7).
- The priority level is higher than the PSW.IPL[3:0] bit value in the CPU.
  When using the fast interrupt, set not only the FIR register but also the corresponding IPRr.IPR[3:0] bits (r = 000 to 255). Set the IPRr.IPR[3:0] bits to a higher level than the PSW.IPL[3:0] bit value in the CPU.
- When using the external pin interrupt, the digital filter of the IRQi pin used is disabled.

Refer to section 14.7.6, Digital Filter for details on the procedure to set the digital filter.

14.11 Usage Notes

14.11.1 Notes on the WAIT instruction When Using the Non-Maskable Interrupt

Confirm that all status flags in the NMISR register are 0 before executing the WAIT instruction.

14.11.2 Interrupt Requests in Software Standby Mode

When an interrupt request occurs in software standby mode but the interrupt source is not set as a source for exiting software standby mode, the request is held in the ICU. The request is handled after exiting by another interrupt source. Note that the interrupt request for the external pin interrupt is not held.

15. Buses

15.1 Overview

Table 15.1 lists the bus specifications, Figure 15.1 shows the bus configuration, and Table 15.2 lists the addresses assigned for each bus.

Table 15.1 Bus Specifications

Bus Type		Description
CPU bus	Instruction bus	• Connected to the CPU (for instructions) • Connected to on-chip memory (RAM, code flash memory) • Operates in synchronization with the system clock (ICLK)
	Operand bus	• Connected to the CPU (for operands) • Connected to on-chip memory (RAM, code flash memory) • Operates in synchronization with the system clock (ICLK)
Memory bus	Memory bus 1	• Connected to RAM
	Memory bus 2	• Connected to code flash memory
Internal main bus	Internal main bus 1	• Connected to the CPU • Operates in synchronization with the system clock (ICLK)
	Internal main bus 2	• Connected to the DMAC and DTC • Connected to on-chip memory (RAM, code flash memory) • Operates in synchronization with the system clock (ICLK)
Internal peripheral bus	Internal peripheral bus 1	• Connected to peripheral modules (TFU, DTC, DMAC, interrupt controller, and bus error monitoring section) • Operates in synchronization with the system clock (ICLK)
	Internal peripheral bus 2	• Connected to peripheral modules (modules other than those connected to internal peripheral buses 1, 3, 4, and 5) • Operates in synchronization with the peripheral-module clock (PCLKB)
	Internal peripheral bus 3	• Connected to peripheral modules (DOC, RSCI, CANFD, and CMPC) • Operates in synchronization with the peripheral-module clock (PCLKB)
	Internal peripheral bus 4	• Connected to peripheral modules (MTU, GPTW, HRPWM, and RSPI) • Operates in synchronization with the peripheral-module clock (PCLKA)
	Internal peripheral bus 5	• Connected to peripheral modules (RSCI, RSPIA, RI3C, and CANFD) • Operates in synchronization with the peripheral-module clock (PCLKA)
	Internal peripheral bus 6	• Connected to code flash (in P/E) and data flash memory • Operates in synchronization with the FlashIF clock (FCLK)

P/E: Programming/Erasure

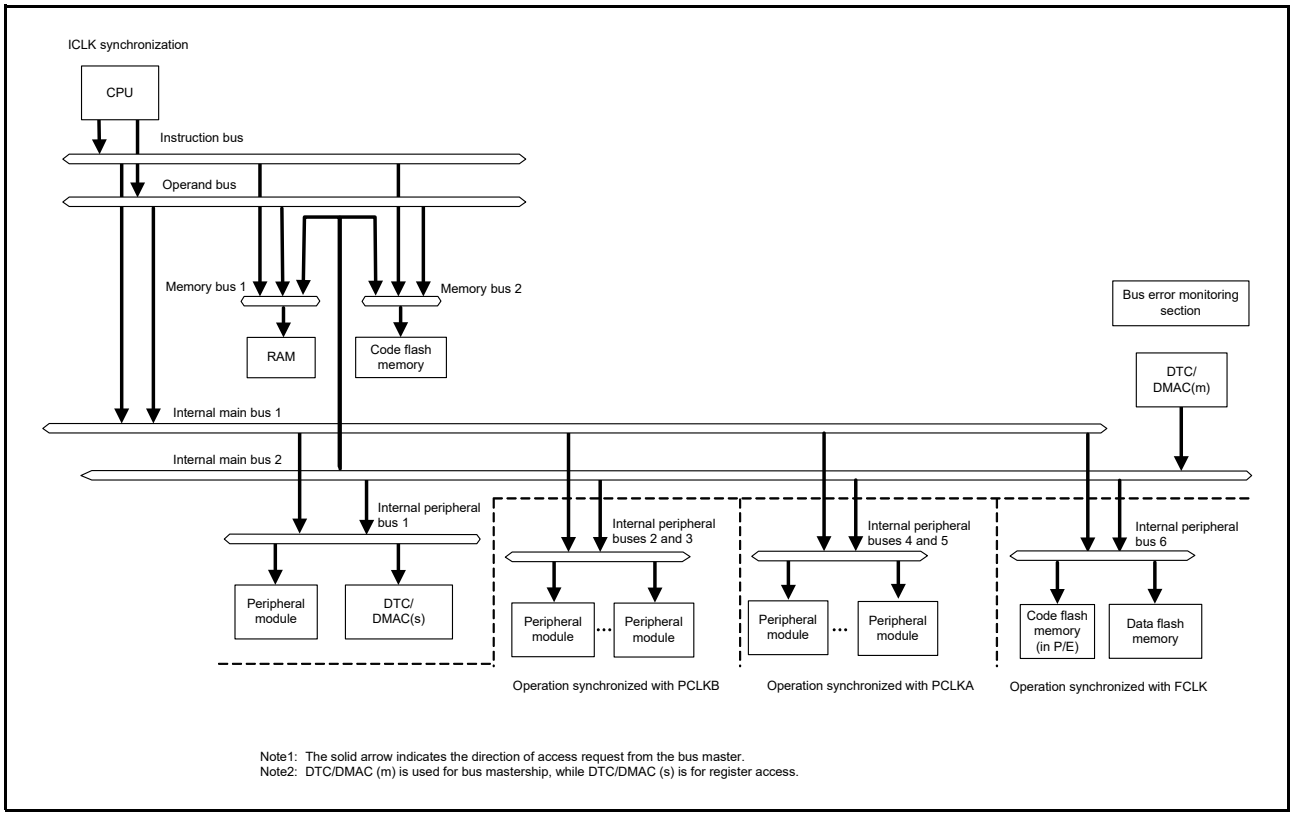


Figure 15.1 Bus Configuration

Table 15.2 Addresses Assigned for Each Bus

Address	Bus	Area
0000 0000h to 0000 FFFFh	Memory bus 1	RAM
0001 0000h to 0007 FFFFh		Reserved area
0008 0000h to 0008 7FFFh	Internal peripheral bus 1	Peripheral I/O registers
0008 8000h to 0009 FFFFh	Internal peripheral bus 2	
000A 0000h to 000B FFFFh	Internal peripheral bus 3	
000C 0000h to 000D FFFFh	Internal peripheral bus 4	
000E 0000h to 000F FFFFh	Internal peripheral bus 5	
0010 0000h to 007F FFFFh	Internal peripheral bus 6	Data flash memory, code flash memory (for programming only)
0080 0000h to 7FFF FFFFh	Reserved area	Reserved area
8000 0000h to FFFF FFFFh	Memory bus 2	Code flash memory (for reading only)

15.2 Description of Buses

15.2.1 CPU Buses

The CPU buses consist of the instruction and operand buses, which are connected to internal main bus 1. As the names suggest, the instruction bus is used to fetch instructions for the CPU, while the operand bus is used for operand access. The instruction bus is 64 bits while the operand bus is 32 bits.
Connection of the instruction and operand buses to RAM and code flash memory provides the CPU with direct access to these areas, i.e. access is not via internal main bus 1. However, only reading is possible in direct access to code flash memory by the CPU; programming and erasure are handled via an internal peripheral bus.
Bus requests for instruction fetching and operand access are arbitrated through internal main bus 1. The order of priority is operand access then instruction fetching.
If instruction fetching and operand access are requested for different buses (memory bus 1, memory bus 2, and internal main bus 1), the bus-access operations can proceed simultaneously. For example, parallel access to code flash memory and RAM is possible.

15.2.2 Memory Buses

The memory buses consist of memory bus 1 and memory bus 2. The RAM is connected to memory bus 1 and code flash memory is connected to memory bus 2. The memory buses are 64 bits. Requests for bus mastership from the CPU buses (instruction fetching and operand) and internal main bus 2 are arbitrated through memory buses 1 and 2.
The priority order of the buses can be set using the memory bus 1 (RAM) priority control bits (BPRA[1:0]) and memory bus 2 (code flash memory) priority control bits (BPRO[1:0]) in the bus priority control register (BUSPRI) for the corresponding memory buses. When the priority order is fixed, internal main bus 2 has priority over the CPU bus (operand over instruction fetching). When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.

15.2.3 Internal Main Buses

The internal main buses consist of a bus for use by the CPU (internal main bus 1) and a bus for use by the other bus-master modules, i.e. the DTC and DMAC (internal main bus 2).
Bus requests for instruction fetching and operand access are arbitrated through internal main bus 1. The order of priority is operand access then instruction fetching.
Requests for bus mastership from the DTC and DMAC are arbitrated by internal main bus 2. The order of priority is DMAC, and then DTC as listed in Table 15.3.
Between the DTC and DMAC, only the one that accepted the transfer request issues the bus mastership request. The priority order of transfer requests between the DTC and DMAC is DMAC0, DMAC1, DMAC2, DMAC3, DMAC4, DMAC5, DMAC6, DMAC7 and then DTC, regardless of the BUSPRI setting.
If the CPU and another bus master are requesting access to different buses (on-chip memory and internal peripheral buses 1 to 6), the respective bus-access operations can proceed simultaneously.
However, when the CPU executes the XCHG instruction, requests for bus access from masters other than the CPU are not accepted until data transfer for the XCHG instruction is completed regardless of the bus priority control register (BUSPRI) setting. Furthermore, requests for bus access from masters other than the DTC are not accepted during reading and writing-back of transfer control information for the DTC.

Table 15.3 Order of Priority for Bus Masters

Priority	Bus Master
High	DMAC
↑	DTC
Low	CPU

Note: The above applies when the priority order of the buses is fixed.

15.2.4 Internal Peripheral Buses

Connection of peripheral modules to the internal peripheral buses is as described in Table 15.4.

Table 15.4 Connection of Peripheral Modules to the Internal Peripheral Buses

Type of Bus	Peripheral Modules
Internal peripheral bus 1	TFU, DTC, DMAC, interrupt controller, and bus error monitoring section
Internal peripheral bus 2	Peripheral modules other than those connected to internal peripheral buses 1, 3, 4, and 5
Internal peripheral bus 3	DOC, RSCI, CANFD, CMPC
Internal peripheral bus 4	MTU, GPTW, HRPWM, and RSPI
Internal peripheral bus 5	RSCI, RSPIA, RI3C, CANFD
Internal peripheral bus 6	Code flash memory (in P/E) or data flash memory

Requests for bus mastership from the CPU (internal main bus 1) and other bus masters (internal main bus 2) are arbitrated through internal peripheral buses 1 to 6.
The priority order of two internal main buses can be set using the bus priority control register (BUSPRI). The priority order can be set with the internal peripheral bus 1 priority control bits (BUSPRI.BPIB[1:0]), internal peripheral bus 2 and 3 priority control bits (BUSPRI.BPGB[1:0]), internal peripheral bus 4 and 5 priority control bits (BUSPRI.BPHB[1:0]), and internal peripheral bus 6 priority control bits (BUSPRI.BPFB[1:0]) for the corresponding internal peripheral buses.
When the priority order is fixed, internal main bus 2 has priority over internal main bus 1. When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted (round-robin method).
The order of accepting requests may change depending on the BUSPRI setting (Refer to Figure 15.2).

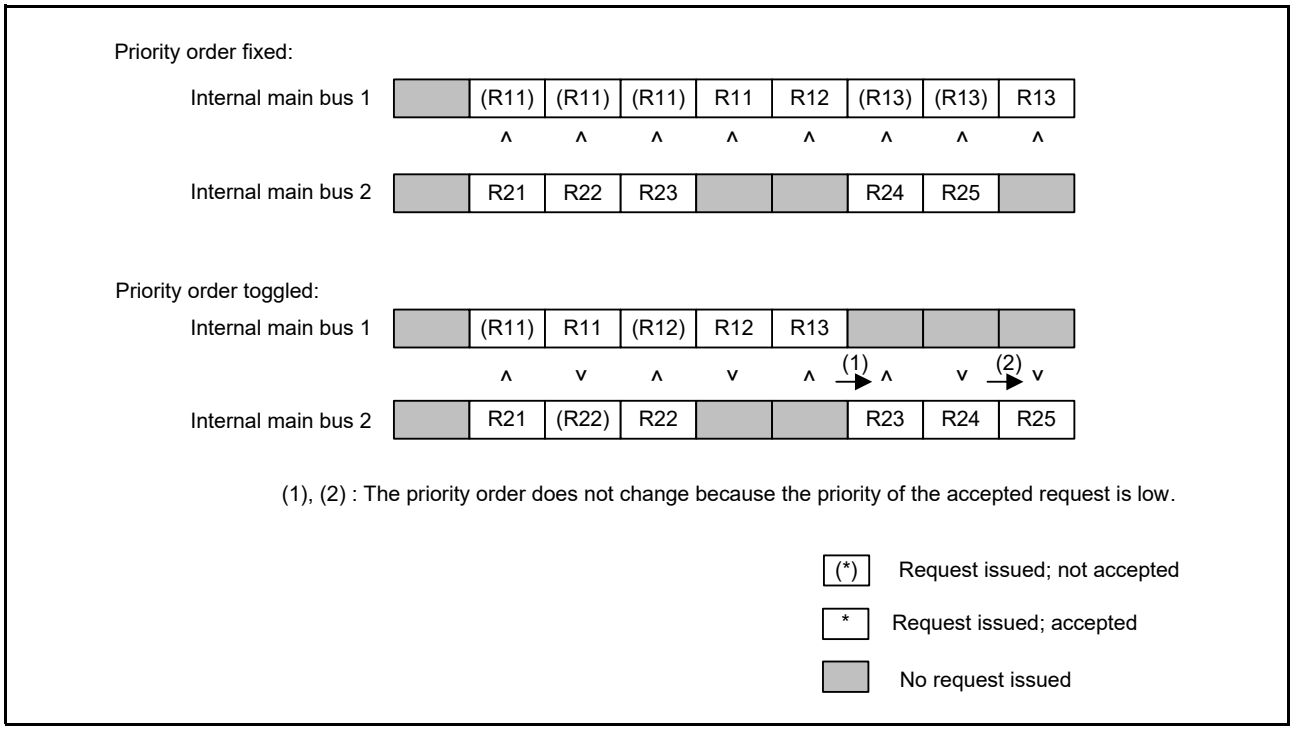


Figure 15.2 Priority Order between Internal Peripheral Bus Accesses

15.2.5 Write Buffer Function (Internal Peripheral Bus)

The internal peripheral bus has the write buffer function, which allows the next round of bus access to start, before the current write access is completed, in write access. However, if the following round of bus access is from the same bus master but to the different internal peripheral bus, it is suspended until the bus operations already in progress are completed. When access to the internal memory is scheduled after the write access to the internal peripheral bus from the CPU, the following round of bus access can be started before the current bus operation is completed and thus the order of accesses may be changed (Refer to Figure 15.3).

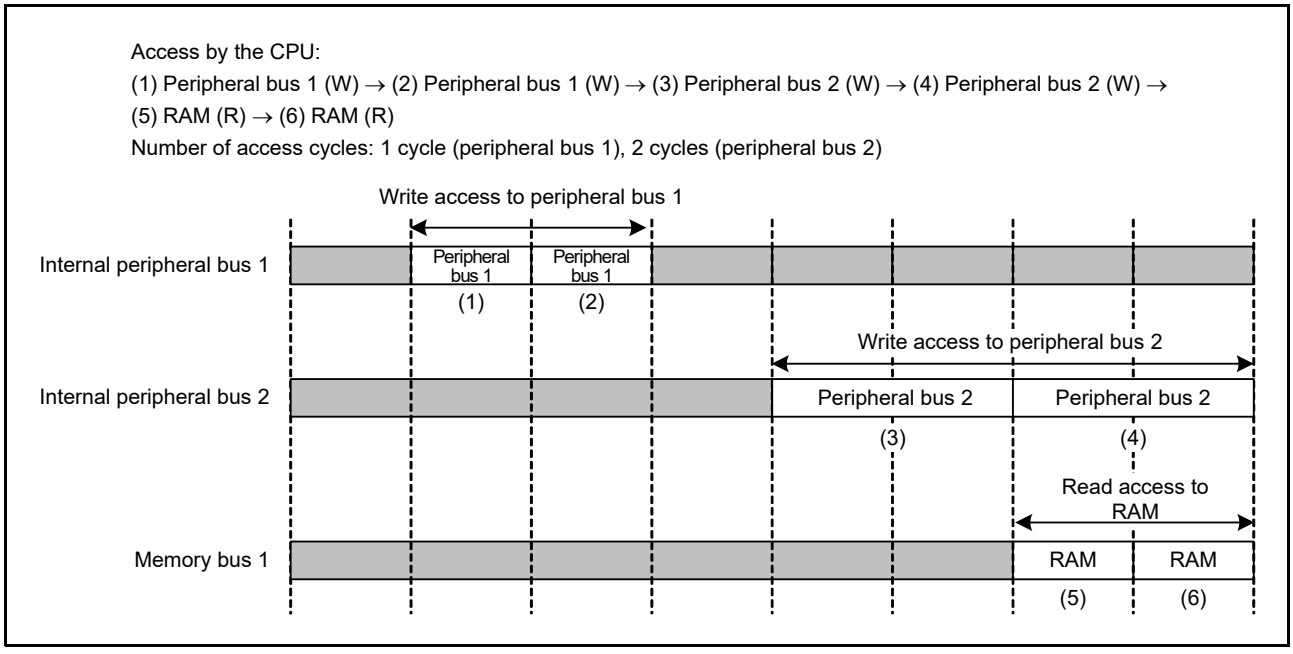


Figure 15.3 Write Buffer Function

15.2.6 Parallel Operation

Parallel operation is possible when different bus-master modules are requesting access to different slave modules. For example, if the CPU is fetching an instruction from code flash memory and an operand from RAM, the DMAC is able to handle transfer between peripheral buses at the same time.

An example of parallel operations is shown in Figure 15.4. In this example, the CPU is able to employ the instruction and operand buses for simultaneous access to code flash memory and RAM, respectively. Furthermore, the DMAC simultaneously employs internal main bus 2 for access to peripheral buses during access to RAM and code flash memory by the CPU.

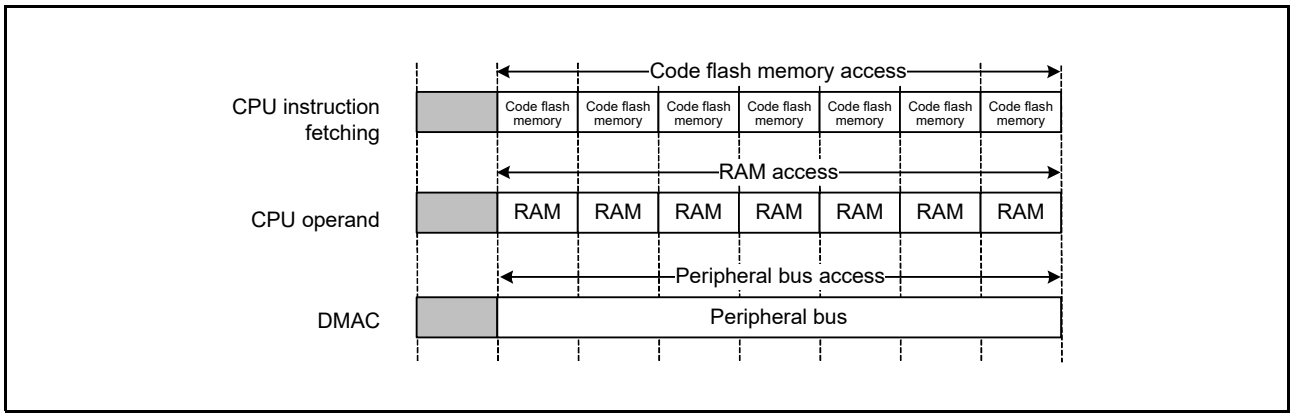


Figure 15.4 Example of Parallel Operations

15.2.7 Restrictions

(1) Prohibition of Access that Spans Areas of Address Space

Single access that spans two areas of the address space is prohibited, and operation of such an access is not guaranteed. Setting must be made so that two areas are not accessed at the same time by a single word or longword access.

(2) Restrictions in Relation to RMPA and String-Manipulation Instructions

The allocation of data to be handled by RMPA or string-manipulation instructions to I/O registers is prohibited, and operation is not guaranteed if this restriction is not observed.

15.3 Register Descriptions

15.3.1 Bus Error Status Clear Register (BERCLR)

Address(es): 0008 1300h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	STSCLR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	STSCLR	Status Clear	0: Invalid 1: Bus error status register cleared	(W)*1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only writing 1 is effective; i.e. writing 0 has no effect.

STSCLR Bit (Status Clear)

Writing 1 to this bit clears the bus error status registers 1 and 2 (BERSR1 and BERSR2).
Writing 0 has no effect. It is read as 0.

15.3.2 Bus Error Monitoring Enable Register (BEREN)

Address(es): 0008 1304h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	TOEN	IGAEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IGAEN	Illegal Address Access Detection Enable	0: Illegal address access detection is disabled. 1: Illegal address access detection is enabled.	R/W
b1	TOEN	Timeout Detection Enable*1, *2	0: Bus timeout detection is disabled. 1: Bus timeout detection is enabled.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When detection is disabled (the TOEN bit is set to 0), bus access can cause the bus to freeze.
Note 2. Do not set the TOEN bit to 0 (bus timeout detection disabled) while timeout errors are being detected.

15.3.3 Bus Error Status Register 1 (BERSR1)

Address(es): 0008 1308h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	MST[2:0]			—	—	TO	IA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IA	Illegal Address Access	0: Illegal address access not made 1: Illegal address access made	R
b1	TO	Timeout	0: Timeout not generated 1: Timeout generated	R
b3, b2	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b6 to b4	MST[2:0]	Bus Master Code	b6 b4 0 0 0: CPU 0 0 1: Reserved 0 1 0: Reserved 0 1 1: DTC/DMAC 1 0 0: Reserved 1 0 1: Reserved 1 1 0: Reserved 1 1 1: Reserved	R
b7	—	Reserved	This bit is read as 0. Writing to this bit has no effect.	R

MST[2:0] Bits (Bus Master Code)

These bits indicate the bus master that accessed a bus when a bus error occurred.

15.3.4 Bus Error Status Register 2 (BERSR2)

Address(es): 0008 130Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ADDR[12:0]													—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b15 to b3	ADDR[12:0]	Bus Error Occurrence Address	The upper 13 bits of an address that was accessed when a bus error occurred (in units of 512 Kbytes).	R

15.3.5 Bus Priority Control Register (BUSPRI)

Address(es): 0008 1310h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	BPFB[1:0]		BPHB[1:0]		BPGB[1:0]		BPIB[1:0]		BPRO[1:0]		BPRA[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	BPRA[1:0]	Memory Bus 1 (RAM) Priority Control	b1 b0 0 0: The order of priority is fixed. 0 1: The order of priority is toggled. 1 0: Setting prohibited 1 1: Setting prohibited	R(/W) *1
b3, b2	BPRO[1:0]	Memory Bus 2 (Code Flash Memory) Priority Control	b3 b2 0 0: The order of priority is fixed. 0 1: The order of priority is toggled. 1 0: Setting prohibited 1 1: Setting prohibited	R(/W) *1
b5, b4	BPIB[1:0]	Internal Peripheral Bus 1 Priority Control	b5 b4 0 0: The order of priority is fixed. 0 1: The order of priority is toggled. 1 0: Setting prohibited 1 1: Setting prohibited	R(/W) *1
b7, b6	BPGB[1:0]	Internal Peripheral Bus 2 and 3 Priority Control	b7 b6 0 0: The order of priority is fixed. 0 1: The order of priority is toggled. 1 0: Setting prohibited 1 1: Setting prohibited	R(/W) *1
b9, b8	BPHB[1:0]	Internal Peripheral Bus 4 and 5 Priority Control	b9 b8 0 0: The order of priority is fixed. 0 1: The order of priority is toggled. 1 0: Setting prohibited 1 1: Setting prohibited	R(/W) *1
b11, b10	BPFB[1:0]	Internal Peripheral Bus 6 Priority Control	b11 b10 0 0: The order of priority is fixed. 0 1: The order of priority is toggled. 1 0: Setting prohibited 1 1: Setting prohibited	R(/W) *1
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. These bits can be written to only once while the DTC and DMAC are stopped. Operation is not guaranteed if it is written more than once.

BPRA[1:0] Bits (Memory Bus 1 (RAM) Priority Control)

These bits specify the priority order for memory bus 1 (RAM).

When the priority order is fixed, internal main bus 2 has priority over CPU buses (instruction and operand buses).

When the priority order is toggled, the bus from which a request has currently been accepted has the lower priority.

BPRO[1:0] Bits (Memory Bus 2 (Code Flash Memory) Priority Control)

These bits specify the priority order for memory bus 2 (code flash memory).

When the priority order is fixed, internal main bus 2 has priority over CPU buses (instruction and operand buses).

When the priority order is toggled, the bus from which a request has currently been accepted has the lower priority.

BPIB[1:0] Bits (Internal Peripheral Bus 1 Priority Control)

These bits specify the priority order for internal peripheral bus 1.

When the priority order is fixed, internal main bus 2 has priority over internal main bus 1.

When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.

BPGB[1:0] Bits (Internal Peripheral Bus 2 and 3 Priority Control)

These bits specify the priority order for internal peripheral buses 2 and 3.
When the priority order is fixed, internal main bus 2 has priority over internal main bus 1.
When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.

BPHB[1:0] Bits (Internal Peripheral Bus 4 and 5 Priority Control)

These bits specify the priority order for internal peripheral buses 4 and 5.
When the priority order is fixed, internal main bus 2 has priority over internal main bus 1.
When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.

BPFB[1:0] Bits (Internal Peripheral Bus 6 Priority Control)

These bits specify the priority order for internal peripheral bus 6.
When the priority order is fixed, internal main bus 2 has priority over internal main bus 1.
When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.

15.4 Bus Error Monitoring Section

The bus error monitoring section monitors the individual areas for bus errors, and when a bus error occurs, the error is indicated to the bus master.

15.4.1 Types of Bus Error

There are two types of bus error: illegal address access and timeout.
Illegal address access is the detection of illegal access to an area, and time-out is the detection of a bus-access operation not being completed within 768 cycles.

15.4.1.1 Illegal Address Access

When the illegal address access detection enable bit (IGAEN) in the bus error monitoring enable register (BEREN) is set to 1, access to an illegal address area leads to illegal address access errors.
The address ranges where access will lead to illegal address access errors are listed in Table 15.5.

15.4.1.2 Timeout

When the timeout detection enable bit (TOEN) in the bus error monitoring enable register (BEREN) is set to 1, bus access that is not completed within 768 cycles leads to a timeout error.

- Internal peripheral buses (2 and 3): Bus access is not completed within 768 peripheral module clock (PCLKB) cycles from the start of the access. In this MCU, a timeout error does not occur.
  Once a timeout error occurs, accesses from the bus master are rejected for 256 PCLKB cycles.
- Internal peripheral buses (4 and 5): Bus access is not completed within 768 peripheral module clock (PCLKA) cycles from the start of the access.
  Once a timeout error occurs, accesses from the bus master are rejected for 256 PCLKA cycles.
- Internal peripheral bus (6): Bus access is not completed within 768 FlashIF clock (FCLK) cycles from the start of the access.
  Once a timeout error occurs, accesses from the bus master are rejected for 256 FCLK cycles. In this MCU, a timeout error does not occur.

15.4.2 Operations When a Bus Error Occurs

When a bus error occurs, the error is indicated to the CPU. Operation is not guaranteed when a bus error occurs.

- Bus error indication to the CPU
  An interrupt is generated. The IERn register in the ICU can specify whether to generate an interrupt in the case of a bus error.

15.4.3 Conditions Leading to Bus Errors

Table 15.5 lists the types of bus errors for each area in the respective address space.
If an illegal address access error or timeout is detected when no bus error has occurred (bus error status register n (BERSRn; n = 1 or 2) is cleared), the detected error is reflected on the BERSRn. Once a bus error occurs, no subsequent bus errors are reflected on the register unless the register is cleared.
If bus errors are simultaneously caused by two or more bus masters, error information of only one bus master is reflected. Once a bus error occurs, the status is retained until BERSRn is cleared.

Table 15.5 Types of Bus Errors

Address	Type of Area	Type of Error	
		Illegal Address Access	Timeout
0000 0000h to 0007 FFFFh	Memory bus 1	—	—
0008 0000h to 0008 7FFFh	Internal peripheral bus 1	—	—
0008 8000h to 0009 FFFFh	Internal peripheral bus 2	Δ	—
000A 0000h to 000B FFFFh	Internal peripheral bus 3	Δ	—
000C 0000h to 000D FFFFh	Internal peripheral bus 4	Δ	✓
000E 0000h to 000F FFFFh	Internal peripheral bus 5	Δ	—
0010 0000h to 0011 FFFFh	Internal peripheral bus 6	—	—
0012 0000h to 007F FFFFh		Δ	—
0080 0000h to 00FF FFFFh	Reserved area	—	—
0100 0000h to 7FFF FFFFh		✓	—
8000 0000h to FEFF FFFFh	Memory bus 2	—	—
FF00 0000h to FF7F FFFFh		—	—
FF80 0000h to FFFF FFFFh		—	—

—: A bus error does not result.
Δ: A bus error may or may not result.
✓: A bus error results.
Note: The capacity of the RAM, data flash memory, and code flash memory differs depending on the product. For details, see section 47, RAM, section 48, Flash Memory (FLASH).

15.5 Interrupt

15.5.1 Interrupt Source

An illegal address access error or detection of a timeout leads to a bus error signal for the interrupt controller.

Table 15.6 Interrupt Source

Name	Interrupt Source	DTC Trigger	DMAC Trigger
BUSERR	Illegal address access error or timeout	Not possible	Not possible

16. Memory-Protection Unit (MPU)

16.1 Overview

The RXv3 CPU incorporates a memory-protection unit that checks the addresses of CPU access to the overall address space (0000 0000h to FFFF FFFFh).

Access-control information can be set for up to eight regions, and permission for access to each region is in accord with this information. The default response to the detection of access to a region where permission has not been set is the generation of a memory-protection error.

The supported access-control information for the individual regions consists of permission to read, permission to write, and permission to execute. This access-control information is effective when the processor mode of the CPU is user mode. Memory protection is not applied when the CPU is in supervisor mode.

Table 16.1 lists the specifications of the memory-protection unit, and Figure 16.1 shows a block diagram of the memory-protection unit.

Table 16.1 Specifications of Memory Protection

Specifications	Description
Region to be covered by memory protection and processor mode	0000 0000h to FFFF FFFFh (in user mode) No memory protection in supervisor mode
Number of regions	8
Page size (smallest unit of protection)	16 bytes
Specifying addresses of individual regions	Setting the page numbers where regions start and end
Setting to make memory protection effective or ineffective in individual regions	A V (valid) bit in each region-n end page number register (REPAGEn) makes the settings effective or ineffective for the corresponding region (n = 0 to 7).
Access-control information settings for individual regions	Instruction execution: Permission to execute Operand access: Permission to read, permission to write
Start of memory-protection operations	After the memory-protection unit has been enabled, access monitoring starting up with the transition to user mode.
Memory-protection error processing	Generation of access exceptions
Addresses where memory-protection errors are generated	Address in instruction execution: The PC value is preserved on the stack. Address in operand access: The address is stored in the data memory-protection error address register (MPDEA).
Determining the reasons for memory-protection errors	The memory-protection error status register (MPESTS) holds indicators of the reason.
Background region setting	Access-control information can be set for the background region (the whole address space).
Processing where regions overlap	The access-control information for access to an overlap between regions is the logical OR of the attributes for the given regions, and permission is given priority.

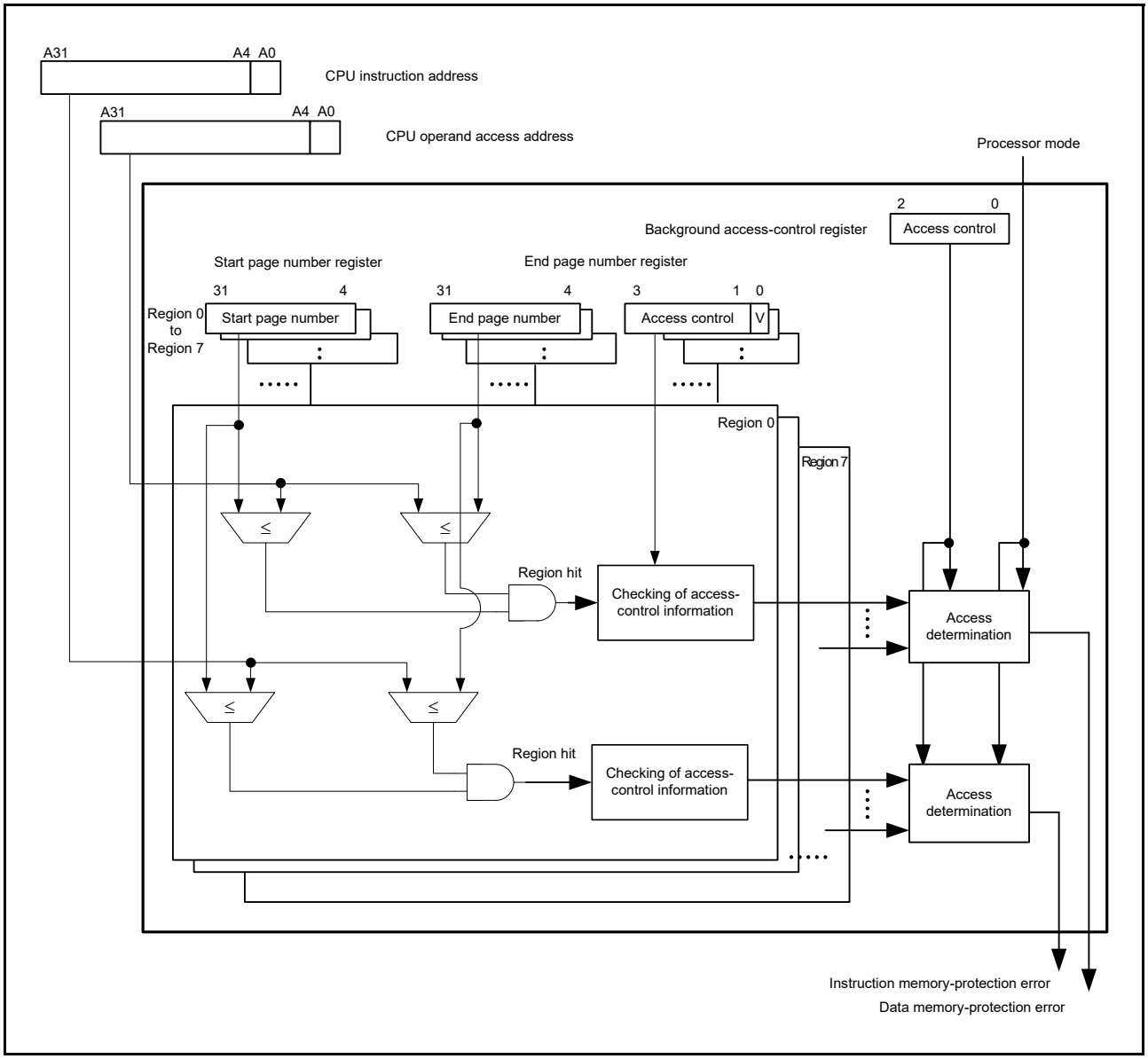


Figure 16.1 Block Diagram of the Memory-Protection Unit

16.1.1 Types of Access Control

There are three types of access control information: permission for instruction execution, permission to read operands, and permission to write operands. Violations of these types of access control are only detected when programs are running in user mode. Violations are not detected when programs are running in supervisor mode.

16.1.2 Regions for Access Control

Up to eight regions for access control are definable. Settings of the range of memory for each access-control region are made in the corresponding region-n start page number register (RSPAGEn) and region-n end page number register (REPAGEn), where n = 0 to 7.
The minimum unit for control of access is the “page”, by which the address space is divided into 16-byte units. The 28 higher-order bits ([31:4]) of the address [31:0] bits correspond to the page number.
The REPAGEn register specifies the access-control information for each area and whether the area is enabled or not.

16.1.3 Background Region

“Background region” refers to the whole address space (0000 0000h to FFFF FFFFh). Access-control information for the background region is set in the background-region access-control register (MPBAC). In contrast to the access-control information for the eight individual regions, protection information for the background region is effective as long as memory protection is enabled (the MPEN bit in the MPEN register is 1).

16.1.4 Overlap between Regions

In cases of overlap between multiple regions, the access-control information becomes the logical OR of the access-control bits for the overlapping regions (including the background region), with permission given priority.

16.1.5 Instructions and Data that Span Regions

Operations in response to the detection of memory-protection errors when instructions or data span regions for which different access-control settings have been made are undefined. Ensure that instructions and data do not span regions for which different access-control settings have been made.

16.2 Register Descriptions

16.2.1 Region-n Start Page Number Register (RSPAGEn) (n = 0 to 7)

Address(es): RSPAGE0 0008 6400h, RSPAGE1 0008 6408h, RSPAGE2 0008 6410h, RSPAGE3 0008 6418h, RSPAGE4 0008 6420h, RSPAGE5 0008 6428h, RSPAGE6 0008 6430h, RSPAGE7 0008 6438h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RSPN[27:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RSPN[27:0]												—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31 to b4	RSPN[27:0]	Region-Start Page Number	Page number where the region starts, for use in region determination	R/W

RSPN[27:0] Bits (Region-Start Page Number)

These bits specify the page number where the region starts.

16.2.2 Region-n End Page Number Register (REPAGEn) (n = 0 to 7)

Address(es): REPAGE0 0008 6404h, REPAGE1 0008 640Ch, REPAGE2 0008 6414h, REPAGE3 0008 641Ch, REPAGE4 0008 6424h, REPAGE5 0008 642Ch, REPAGE6 0008 6434h, REPAGE7 0008 643Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	REPN[27:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	REPN[27:0]												UAC[2:0]			V
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	V	Valid Bit	0: Region setting invalid 1: Region setting valid	R/W
b3 to b1	UAC[2:0]	Access Control Bits in User Mode	b3 0: Reading prohibited 1: Reading permitted b2 0: Writing prohibited 1: Writing permitted b1 0: Execution prohibited 1: Execution permitted	R/W
b31 to b4	REPN[27:0]	Region-End Page Number	Page number where the region ends, for use in region determination	R/W

V Bit (Valid Bit)

This bit enables or disables the settings for the corresponding region.
This bit is set to 0 when the region invalidation operation register (MPOPI) invalidates all access-controlled areas.

UAC[2:0] Bits (Access Control Bits in User Mode)

These bits specify the access control in user mode.

REPN[27:0] Bits (Region-End Page Number)

These bits specify the page number where the region ends.
Specify a value that is greater than or equal to the page number where the corresponding region starts. The page specified by the region-end page number is part of the target region for memory protection.

16.2.3 Memory-Protection Enable Register (MPEN)

Address(es): 0008 6500h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MPEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MPEN	Memory-Protection Enable	0: The memory protection is disabled. 1: The memory protection is enabled.	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

MPEN Bit (Memory-Protection Enable)

This bit enables or disables the memory protection.

After 1 has been written to this bit, address checking for memory protection by the CPU starts on the execution of a branch instruction (RTE or RTFI) that shifts operation to the user mode.

16.2.4 Background Access Control Register (MPBAC)

Address(es): 0008 6504h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	UBAC[2:0]			—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3 to b1	UBAC[2:0]	Background Access Control Bits in User Mode	b3 0: Reading prohibited 1: Reading permitted b2 0: Writing prohibited 1: Writing permitted b1 0: Execution prohibited 1: Execution permitted	R/W
b31 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

UBAC[2:0] Bits (Background Access Control Bits in User Mode)

These bits specify the background access control in user mode.

16.2.5 Memory-Protection Error Status-Clearing Register (MPECLR)

Address(es): 0008 6508h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CLR	Error Status-Clearing	[Reading] 0: Fixed value for reading [Writing] 0: Nothing is done. 1: The DRW, DMPER and IMPER bits in MPESTS are set to 0.	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CLR Bit (Error Status-Clearing)

This bit clears the data read/write bit (DRW), the data memory-protection error generation bit (DMPER), and the instruction memory-protection error generation bit (IMPER) in the memory-protection error status register (MPESTS) to 0.

16.2.6 Memory-Protection Error Status Register (MPESTS)

Address(es): 0008 650Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	DRW	DMPER	IMPER
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IMPER	Instruction Memory-Protection Error Generation	0: No instruction memory-protection error was generated. 1: Instruction memory-protection error was generated.	R
b1	DMPER	Data Memory-Protection Error Generation	0: No data memory-protection error was generated. 1: Data memory-protection error was generated.	R
b2	DRW	Data Read/Write	0: Data were read. 1: Data were written.	R
b31 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

IMPER Bit (Instruction Memory-Protection Error Generation)

This bit indicates the state of memory-protection error generation by instruction execution.

Setting the error status-clearing bit (CLR) in the memory-protection error status-clearing register (MPECLR) to 1 clears this bit to 0.

DMPER Bit (Data Memory-Protection Error Generation)

This bit indicates the state of memory-protection error generation by operand access.

Setting the error status-clearing bit (CLR) in the memory-protection error status-clearing register (MPECLR) to 1 clears this bit to 0.

DRW Bit (Data Read/Write)

For a memory-protection error produced by operand access, this bit indicates the read/write attribute of the access operation. This bit is only valid when the DMPER bit is 1.

Setting the error status-clearing bit (CLR) in the memory-protection error status-clearing register (MPECLR) to 1 clears this bit to 0.

16.2.7 Data Memory-Protection Error Address Register (MPDEA)

Address(es): 0008 6514h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DEA[31:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DEA[31:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	DEA[31:0]	Data Memory-Protection Error Address	Data memory-protection error address	R

DEA[31:0] Bits (Data Memory-Protection Error Address)

These bits retain the address for which operand access generated a memory-protection error.

16.2.8 Region Search Address Register (MPSA)

Address(es): 0008 6520h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SA[31:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SA[31:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	SA[31:0]	Region Search Address	Address for region searching	R/W

SA[31:0] Bits (Region Search Address)

These bits specify the address for use in comparison with region-start addresses in the region-n start page number registers (RSPAGEn) and region-end addresses in the region-n end page number registers (REPAGEn).

16.2.9 Region Search Operation Register (MPOPS)

Address(es): 0008 6524h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	S
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	S	Region Search Operation	[Reading] 0: Fixed value for reading [Writing] 0: Nothing is done. 1: A region-search operation proceeds.	R/W
b15 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

S Bit (Region Search Operation)

Setting this bit to 1 makes the memory-protection unit perform a region-search operation. The address specified in the region search address register (MPSA) is compared with the address information for individual regions to search for a hitting region.

The result of searching is stored in the data-hit region bits (HITD[7:0]) of the data-hit region register (MHITD). Moreover, the logical OR of the respective access control bits for hitting regions is stored in the data-hit region access control bits (UHACD[2:0]) in user mode.

16.2.10 Region Invalidation Operation Register (MPOPI)

Address(es): 0008 6526h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	INV
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	INV	Region Invalidate Start	[Reading] 0: Fixed value for reading [Writing] 0: Nothing is done. 1: All access-controlled areas are invalidated.	R/W
b15 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

INV Bit (Region Invalidate Start)

Setting this bit to 1 clears the valid (V) bits in all of the region-n end page number registers (REPAGEn) to 0. After a V bit is set to 0, all settings other than background access-control settings are invalid.

16.2.11 Instruction-Hit Region Register (MHITI)

Address(es): 0008 6528h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	HITI[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	UHACI[2:0]			—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3 to b1	UHACI[2:0]	Instruction-Hit Region Access Control Bits in User Mode	b3 0: Reading prohibited 1: Read permitted b2 0: Writing prohibited 1: Writing permitted b1 0: Execution prohibited 1: Execution is permitted.	R
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23 to b16	HITI[7:0]	Instruction-Hit Region	When the instruction memory-protection error generation bit (MPESTS.IMPER) = 1, [b23:b16] = 0000 0000b indicates that attempted access to the background region led to an instruction memory-protection error. Other than above b23 0: Instruction memory-protection error was not generated in region 7. 1: Instruction memory-protection error was generated in region 7. b22 0: Instruction memory-protection error was not generated in region 6. 1: Instruction memory-protection error was generated in region 6. b21 0: Instruction memory-protection error was not generated in region 5. 1: Instruction memory-protection error was generated in region 5. b20 0: Instruction memory-protection error was not generated in region 4. 1: Instruction memory-protection error was generated in region 4. b19 0: Instruction memory-protection error was not generated in region 3. 1: Instruction memory-protection error was generated in region 3. b18 0: Instruction memory-protection error was not generated in region 2. 1: Instruction memory-protection error was generated in region 2. b17 0: Instruction memory-protection error was not generated in region 1. 1: Instruction memory-protection error was generated in region 1. b16 0: Instruction memory-protection error was not generated in region 0. 1: Instruction memory-protection error was generated in region 0.	R
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

UHACI[2:0] Bits (Instruction-Hit Region Access Control Bits in User Mode)

These bits hold the user-mode access control bits (REPAGEn.UAC[2:0]) for the region where the instruction memory-protection error was generated.

If the error was generated in an overlap between regions, the value stored here is the logical OR of the user-mode access control bits for the corresponding regions (including the background region).

HITI[7:0] Bits (Instruction-Hit Region)

These bits indicate the region where an instruction memory-protection error was generated. These bits are set to 0000 0000b in response to the generation of an instruction memory-protection error in the background region.

16.2.12 Data-Hit Region Register (MHITD)

Address(es): 0008 652Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	HITD[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	UHACD[2:0]			—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3 to b1	UHACD[2:0]	Data-Hit Region Access Control Bits in User Mode	b3 0: Reading prohibited 1: Reading permitted b2 0: Writing prohibited 1: Writing permitted b1 0: Execution prohibited 1: Execution permitted	R
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23 to b16	HITD[7:0]	Data-Hit Region	When the data memory-protection error generation bit (MPESTS.DMPER) = 1, [b23:b16] = 0000 0000b indicates that attempted access to the background region led to a data memory-protection error. Other than above b23 0: Neither a data memory-protection error nor a search hit was generated in region 7. 1: A data memory-protection error or search hit was generated in region 7. b22 0: Neither a data memory-protection error nor a search hit was generated in region 6. 1: A data memory-protection error or search hit was generated in region 6. b21 0: Neither a data memory-protection error nor a search hit was generated in region 5. 1: A data memory-protection error or search hit was generated in region 5. b20 0: Neither a data memory-protection error nor a search hit was generated in region 4. 1: A data memory-protection error or search hit was generated in region 4. b19 0: Neither a data memory-protection error nor a search hit was generated in region 3. 1: A data memory-protection error or search hit was generated in region 3. b18 0: Neither a data memory-protection error nor a search hit was generated in region 2. 1: A data memory-protection error or search hit was generated in region 2. b17 0: Neither a data memory-protection error nor a search hit was generated in region 1. 1: A data memory-protection error or search hit was generated in region 1. b16 0: Neither a data memory-protection error nor a search hit was generated in region 0. 1: A data memory-protection error or search hit was generated in region 0.	R
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

UHACD[2:0] Bits (Data-Hit Region Access Control Bits in User Mode)

These bits hold the user-mode access control bits (REPAGEn.UAC[2:0]) that have been set for the region where a data memory-protection error was generated or the region that produced a hit in region searching.
When an error is generated in an overlap between regions or a hit was generated in region searching, the value stored here is the logical OR of the user-mode access control bits for the corresponding regions (including the background region).

HITD[7:0] Bits (Data-Hit Region)

These bits indicate the region where a data memory-protection error was generated or the region that produced a hit in a region search. These bits are set to 0000 0000b for a data memory-protection error generated in the background region.

Note: When access to a register of memory protection unit in user mode generates a data memory-protection error, the value in this register is set to 0000 0000h.

16.3 Functions

16.3.1 Memory Protection

Memory protection means monitoring, in accord with the access-control information that has been set for the individual access-control regions and the background region, whether or not access by programs running in user mode violates the access-control settings. The memory-protection unit notifies the CPU of access-control violations (or memory-protection errors) when they are detected, causing the CPU to start access-exception processing.
Memory protection is enabled by setting the memory-protection enable (MPEN) bit in the memory-protection enable (MPEN) register to 1.
An instruction memory-protection error is generated on detection of an instruction-execution violation and a data memory-protection error is generated on detection of an operand-access reading or writing violation. Operand access that leads to a data memory-protection error is not actually executed.

16.3.2 Region Search

Region search means enquiry as to which of the eight specified access regions was “hit” and how the access-control information (permission to execute, to read, and to write) is set.
When the region search operation (S) bit in the region-search operation (MPOPS) register is set to 1, the address specified in the region search address (MPSA) register is compared with the addresses for the individual regions. After a region search is executed, the data-hit region register (MHITD) indicates the logical OR of the access-control information for the region which was “hit” and for the other regions.

16.3.3 Protection of Registers Related to the Memory-Protection Unit

Registers related to the memory-protection unit are not accessible through means of access other than operand access by the CPU (i.e. by instruction fetching or DMA). The registers related to the memory-protection unit are only accessible in supervisor mode. Attempted access to registers related to the memory-protection unit in user mode through operand access by the CPU leads to a data memory-protection error regardless of whether or not memory protection is in effect at the given location.

16.3.4 Flow for Determination of Access by the Memory-Protection Function

Figure 16.2 shows the flow of determination in the case of data access and Figure 16.3 shows the flow of determination in the case of instruction access.

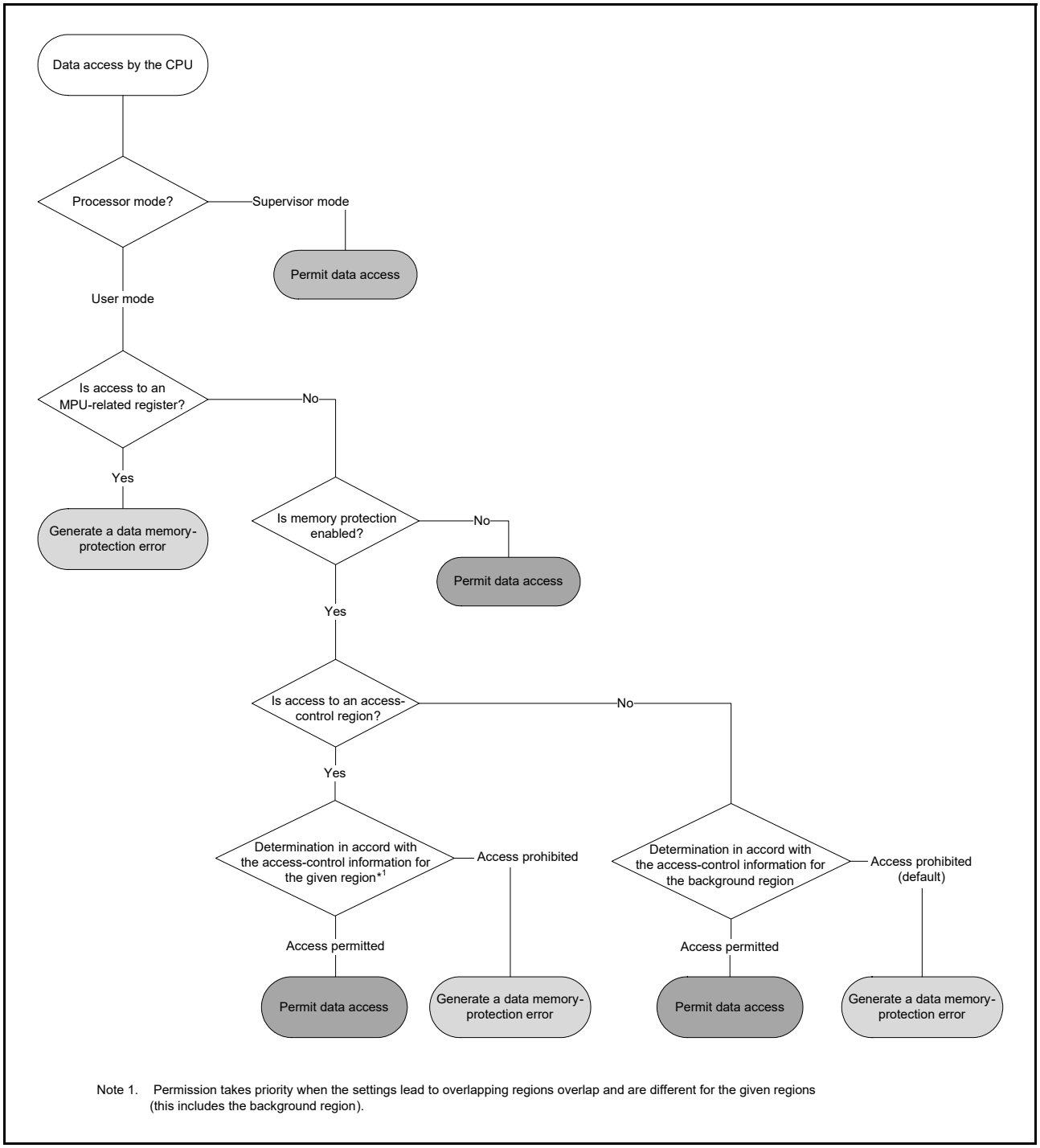
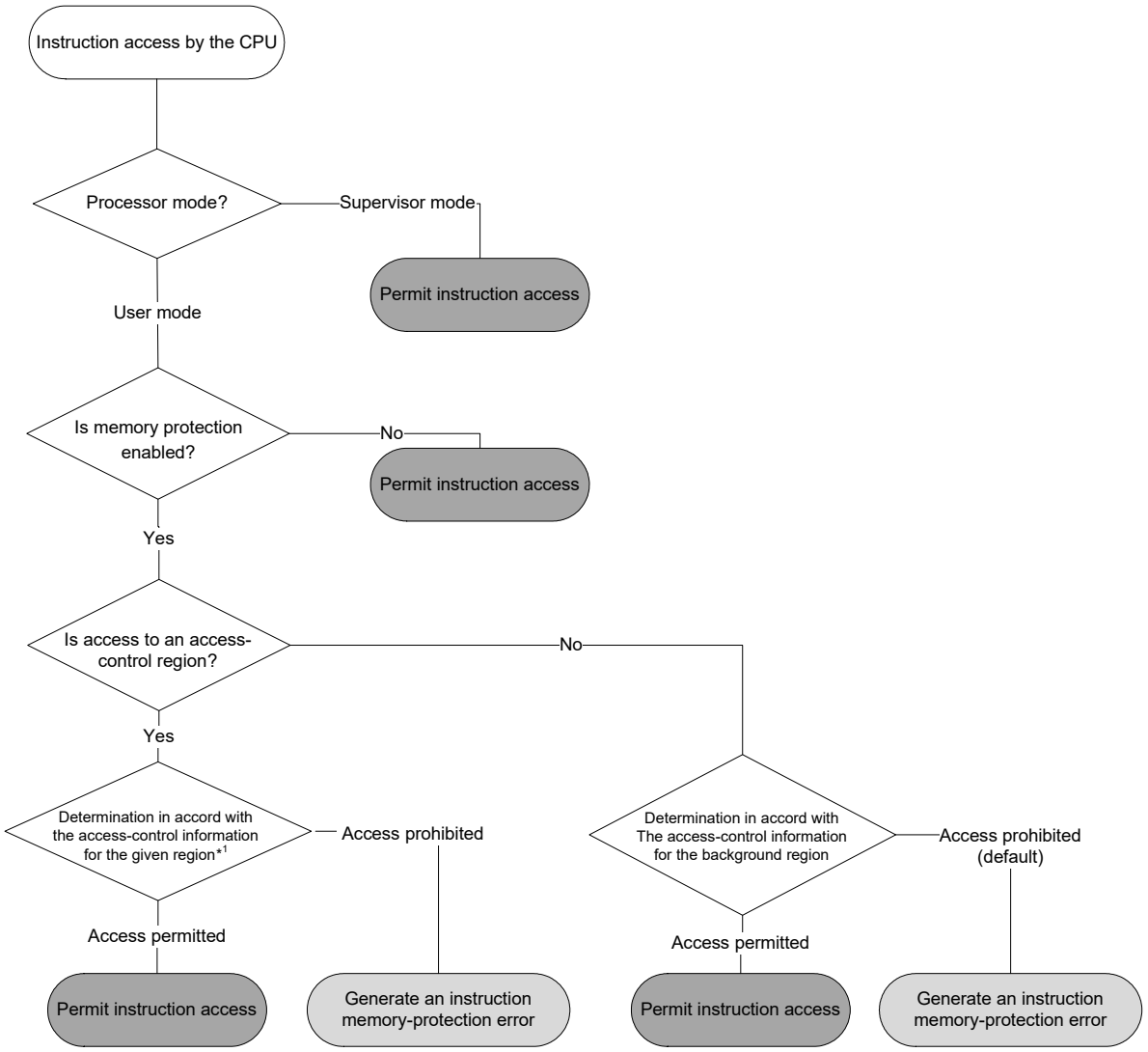


Figure 16.2 Flow of Determination for Data Access



Note 1. Permission takes priority when the settings lead to overlapping regions overlap and are not the same for the given regions (this includes the background region).

Figure 16.3 Flow of Determination for Instruction Access

16.4 Procedures for Using Memory Protection

16.4.1 Setting Access-Control Information

Access-control information for the various regions is set in supervisor mode.
Settings for up to eight access-control regions are made in the region-n start page number registers (RSPAGEn) and region-n end page number registers (REPAGEn), where n = 0 to 7.
Settings for the background access-control region are made in the background access-control register (MPBAC).

16.4.2 Enabling Memory Protection

Setting the memory-protection enable (MPEN) bit in the memory-protection enable (MPEN) register to 1 while operation is in supervisor mode enables memory protection.

16.4.3 Transition to User Mode

After updating the registers related to the memory-protection unit, read any of these registers and check that the settings have been made before the transition to user mode.
Either of the methods below can be used for the transition from supervisor mode to user mode.

- Set the processor mode setting (PM) bit in the copy of the processor status word (PSW) saved in the stack area to 1 (the setting for user mode) and then execute an RTE instruction.
- Set the PM bit in the backup processor status word (BPSW) to 1 and then execute an RTFI instruction.

Note: Using an MVTC or POPC instruction to write to the PSW.PM bit is invalid. Use an RTE or RTFI instruction to update the value of the PSW.PM bit.

The memory-protection unit starts checking instruction-execution access and operand access by the CPU on the transition to user mode.

16.4.4 Processing in Response to Memory-Protection Errors

The CPU starts access-exception processing on detection of a violation of protection set up by the access-control information (i.e. a memory-protection error). For details on CPU operations in access-exception processing, refer to section 13, Exception Handling.
To determine whether an instruction memory-protection error or data memory-protection error has been generated, check the values of the instruction memory-protection error generation (IMPER) and data memory-protection error generation (DMPER) bits in the memory-protection error status register (MPESTS) from within the exception-processing routine. After confirming the type of error, clear the memory-protection error status register (MPESTS) by writing 1 to the status clearing (CLR) bit in the memory-protection error status clearing register (MPECLR).

(1) When a data memory-protection error is generated

Access-exception processing by the CPU saves the address of the instruction that led to the memory-protection error on the stack. Furthermore, the address of the operand for which access led to a memory-protection error is stored in the data memory-protection error address register (MPDEA) and the region information for the region where the memory-protection error was generated is stored in the data-hit region register (MHITD).

- Violations of access control in access to valid regions 0 to 7

The data-hit region bits (MHITD.HITD[7:0]) with the same region number as the region where the error occurred is set to 1. In user mode, the logical “or” of the region access-control information for the location where the error occurred is set in the data-hit region access-control bits (MHITD.UHACD[2:0]).

- Violations of access control for the background region, besides access to outside valid regions 0 to 7

The data-hit region bits (MHITD.HITD[7:0]) are set to 0000 0000b. In user mode, the access-control information for the background region is set in the data-hit region access-control bits (MHITD.UHACD[2:0]).

Referring to this information can pinpoint the sources of errors.

(2) When an instruction memory-protection error is generated

Access-exception processing by the CPU saves the address of the instruction that led to the memory-protection error on the stack. Furthermore, the region information for the region where the memory-protection error was generated is stored in the instruction-hit region register (MHITI).

- Violations of access control in access to valid regions 0 to 7

The instruction-hit region bit (MHITI.HITI[7:0]) with the same region number as the region where the error occurred is set to 1. In user mode, the logical “or” of the region access-control information for the location where the error occurred is set in the instruction-hit region access-control bits (MHITI.UHACI[2:0]).

- Violations of access control for the background region, besides access to outside valid regions 0 to 7

The instruction-hit region bits (MHITI.HITI[7:0]) are set to 0000 0000b. In user mode, the access-control information for the background region is set in the instruction-hit region access-control bits (MHITI.UHACI[2:0]).

Referring to this information can pinpoint the sources of errors.

17. DMA Controller (DMACAa)

This MCU incorporates an 8-channel direct memory access controller (DMAC).
The DMAC is a module to transfer data without the CPU. When a DMA transfer request is generated, the DMAC transfers data stored at the transfer source address to the transfer destination address.

17.1 Overview

Table 17.1 lists the specifications of the DMAC, and Figure 17.1 shows a block diagram of the DMAC.

Table 17.1 Specifications of DMAC

Item		Description
Number of channels		8 (DMACm (m = 0 to 7))
Transfer space		512 Mbytes (0000 0000h to 0FFF FFFFh and F000 0000h to FFFF FFFFh excluding reserved areas)
Maximum transfer volume		64 Mbytes (Maximum number of transfers in block transfer mode: 1,024 data × 65,536 blocks)
DMA request source		• Request source selectable for each channel Software trigger Interrupt requests from peripheral modules or trigger input to external interrupt input pins*1
Channel priority		Channel 0 > Channel 1 > Channel 2 > Channel 3... > Channel 7 (Channel 0: Highest)
Transfer data	Single data	Bit length: 8, 16, 32 bits
	Block size	Number of data: 1 to 1,024
Transfer mode	Normal transfer mode	• One data transfer by one DMA transfer request • Free running mode (setting in which total number of data transfers is not specified) settable
	Repeat transfer mode	• One data transfer by one DMA transfer request • Program returns to the transfer start address on completion of the repeat size of data transfer specified for the transfer source or destination. • Maximum settable repeat size: 1,024
	Block transfer mode	• One block data transfer by one DMA transfer request • Maximum settable block size: 1,024 data
Selective functions	Extended repeat area function	• Function in which data can be transferred by repeating the address values in the specified range with the upper bit values in the transfer address register fixed • Area of 2 bytes to 128 Mbytes separately settable as extended repeat area for transfer source and destination
Interrupt request	Transfer end interrupt	Generated when the specified number of transfers is completed in normal transfer mode Generated when the specified repeat count of transfers is completed in repeat transfer mode Generated when the specified block count of transfers is completed in block transfer mode
	Transfer escape end interrupt	Generated when the repeat size of data transfer is completed or the extended repeat area overflows.
Event link function		An event link request is generated after each data transfer (for block transfer, after each block is transferred).
Power consumption reduction function		Module-stop state can be set.

Note 1. For details on DMA request sources, see Table 14.4, Interrupt Vector Table in section 14, Interrupt Controller (ICUG).

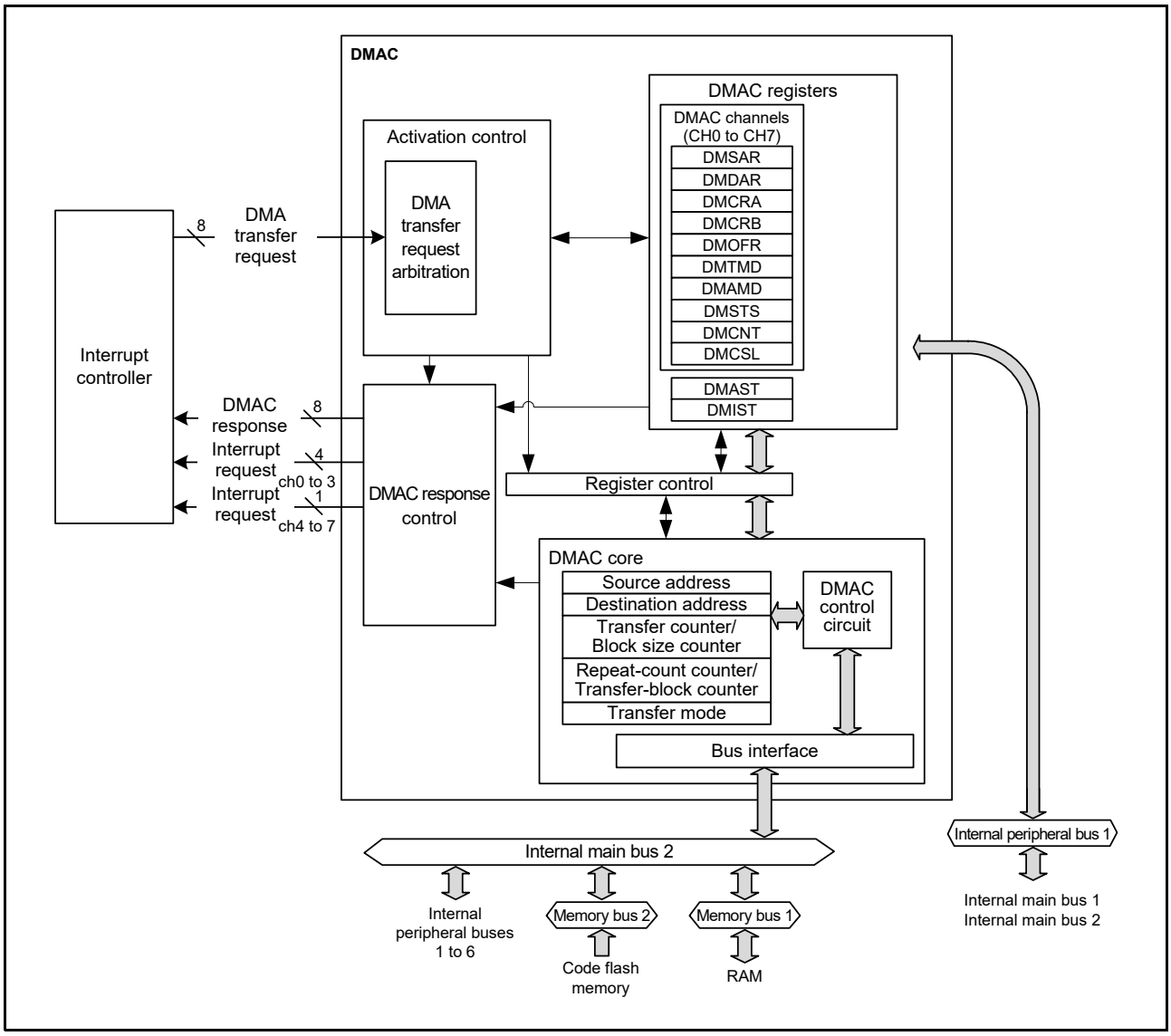


Figure 17.1 Block Diagram of DMAC

17.2 Register Descriptions

17.2.1 DMA Source Address Register (DMSAR)

Address(es): DMAC0.DMSAR 0008 2000h, DMAC1.DMSAR 0008 2040h, DMAC2.DMSAR 0008 2080h, DMAC3.DMSAR 0008 20C0h, DMAC4.DMSAR 0008 2100h, DMAC5.DMSAR 0008 2140h, DMAC6.DMSAR 0008 2180h, DMAC7.DMSAR 0008 21C0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Description	Setting Range	R/W
b31 to b0	Specifies the transfer source start address.	0000 0000h to 0FFF FFFFh (256 Mbytes) F000 0000h to FFFF FFFFh (256 Mbytes)	R/W

Set DMSAR while DMAC stops (the DMST bit in DMAST = 0) or DMA transfer is disabled (the DTE bit in DMCNT = 0).

Setting bits 31 to 29 is invalid; a value of bit 28 is extended to bits 31 to 29. Reading DMSAR returns the extended value.

17.2.2 DMA Destination Address Register (DMDAR)

Address(es): DMAC0.DMDAR 0008 2004h, DMAC1.DMDAR 0008 2044h, DMAC2.DMDAR 0008 2084h, DMAC3.DMDAR 0008 20C4h, DMAC4.DMDAR 0008 2104h, DMAC5.DMDAR 0008 2144h, DMAC6.DMDAR 0008 2184h, DMAC7.DMDAR 0008 21C4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Description	Setting Range	R/W
b31 to b0	Specifies the transfer destination start address.	0000 0000h to 0FFF FFFFh (256 Mbytes) F000 0000h to FFFF FFFFh (256 Mbytes)	R/W

Set DMDAR while DMAC stops (the DMST bit in DMAST = 0) or DMA transfer is disabled (the DTE bit in DMCNT = 0).

Setting bits 31 to 29 is invalid; a value of bit 28 is extended to bits 31 to 29. Reading DMDAR returns the extended value.

17.2.3 DMA Transfer Count Register (DMCRA)

Address(es): DMAC0.DMCRA 0008 2008h, DMAC1.DMCRA 0008 2048h, DMAC2.DMCRA 0008 2088h, DMAC3.DMCRA 0008 20C8h, DMAC4.DMCRA 0008 2108h, DMAC5.DMCRA 0008 2148h, DMAC6.DMCRA 0008 2188h, DMAC7.DMCRA 0008 21C8h

- Normal transfer mode

DMCRAH

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
—	—	—	—	—	—										
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Value after reset:

DMCRAL

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Value after reset:

- Repeat transfer mode, block transfer mode

DMCRAH

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
—	—	—	—	—	—										
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Value after reset:

DMCRAL

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	0										
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Value after reset:

Symbol	Bit Name	Description	R/W
DMCRAL	Lower bits of transfer count	Specifies the number of transfer operations	R/W
DMCRAH	Upper bits of transfer count		R/W

Note: Set the same value for DMCRAH and DMCRAL in repeat transfer mode and block transfer mode.

(1) Normal Transfer Mode (MD[1:0] Bits in DMACm.DMTMD = 00b)

DMCRAL functions as a 16-bit transfer counter.

The number of transfer operations is one when the setting is 0001h, and 65,535 when it is FFFFh. The value is decremented by one each time data is transferred.

When the setting is 0000h, no specific number of transfer operations is set; data transfer is performed with the transfer counter stopped (free running mode).

DMCRAH is not used in normal transfer mode. Write 0000h to DMCRAH.

(2) Repeat Transfer Mode (MD[1:0] Bits in DMACm.DMTMD = 01b)

DMCRAH specifies the repeat size and DMCRAL functions as a 10-bit transfer counter.
The number of transfer operations is one when the setting is 001h, 1023 when it is 3FFh, and 1024 when it is 000h. In repeat transfer mode, a value in the range of 000h to 3FFh (1 to 1024) can be set for DMCRAH and DMCRAL.
Setting bits 15 to 10 in DMCRAL is invalid. Write 0 to these bits.
The value in DMCRAL is decremented by one each time data is transferred until it reaches 000h, at which the value in DMCRAH is loaded into DMCRAL.

(3) Block Transfer Mode (MD[1:0] Bits in DMACm.DMTMD = 10b)

DMCRAH specifies the block size and DMCRAL functions as a 10-bit block size counter.
The block size is one when the setting is 001h, 1023 when it is 3FFh, and 1024 when it is 000h. In block transfer mode, a value in the range of 000h to 3FFh can be set for DMCRAH and DMCRAL.
Setting bits 15 to 10 in DMCRAL is invalid. Write 0 to these bits.
The value in DMCRAL is decremented by one each time data is transferred until it reaches 000h, at which the value in DMCRAH is loaded into DMCRAL.

17.2.4 DMA Block Transfer Count Register (DMCRB)

Address(es): DMAC0.DMCRB 0008 200Ch, DMAC1.DMCRB 0008 204Ch, DMAC2.DMCRB 0008 208Ch, DMAC3.DMCRB 0008 20CCh, DMAC4.DMCRB 0008 210Ch, DMAC5.DMCRB 0008 214Ch, DMAC6.DMCRB 0008 218Ch, DMAC7.DMCRB 0008 21CCh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Description	Setting Range	R/W
b15 to b0	Specifies the block count or repeat count of transfers.	0001h to FFFFh (1 to 65,535) 0000h (65,536)	R/W

This register specifies the transfer block count in block transfer mode and the repeat count in repeat transfer mode.
In repeat transfer mode, the value is decremented by one when the final data of one repeat size is transferred.
In block transfer mode, the value is decremented by one when the final data of one block size is transferred.
In normal transfer mode, DMCRB is not used. The setting is invalid.

17.2.5 DMA Transfer Mode Register (DMTMD)

Address(es): DMAC0.DMTMD 0008 2010h, DMAC1.DMTMD 0008 2050h, DMAC2.DMTMD 0008 2090h, DMAC3.DMTMD 0008 20D0h, DMAC4.DMTMD 0008 2110h, DMAC5.DMTMD 0008 2150h, DMAC6.DMTMD 0008 2190h, DMAC7.DMTMD 0008 21D0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MD[1:0]		DTS[1:0]		—	—	SZ[1:0]		—	—	—	—	—	—	DCTG[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	DCTG[1:0]	Transfer Request Source Select	b1 b0 0 0: Software 0 1: Interrupts*1 from peripheral modules or external interrupt input pins 1 0: Setting prohibited 1 1: Setting prohibited	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	SZ[1:0]	Transfer Data Size Select	b9 b8 0 0: 8 bits 0 1: 16 bits 1 0: 32 bits 1 1: Setting prohibited	R/W
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13, b12	DTS[1:0]	Repeat Area Select	b13 b12 0 0: The destination is specified as the repeat area or block area. 0 1: The source is specified as the repeat area or block area. 1 0: The repeat area or block area is not specified. 1 1: Setting prohibited	R/W
b15, b14	MD[1:0]	Transfer Mode Select	b15 b14 0 0: Normal transfer 0 1: Repeat transfer 1 0: Block transfer 1 1: Setting prohibited	R/W

Note 1. DMA request source is selected using the DMRSRm registers of the ICU. For details on DMA request sources, refer to Table 14.4, Interrupt Vector Table in section 14, Interrupt Controller (ICUG).

DTS[1:0] Bits (Repeat Area Select)

DTS[1:0] select either the source or destination as the repeat area in repeat or block transfer mode. In normal transfer mode, setting these bits is invalid.

17.2.6 DMA Interrupt Setting Register (DMINT)

Address(es): DMAC0.DMINT 0008 2013h, DMAC1.DMINT 0008 2053h, DMAC2.DMINT 0008 2093h, DMAC3.DMINT 0008 20D3h, DMAC4.DMINT 0008 2113h, DMAC5.DMINT 0008 2153h, DMAC6.DMINT 0008 2193h, DMAC7.DMINT 0008 21D3h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	DTIE	ESIE	RPTIE	SARIE	DARIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DARIE	Destination Address Extended Repeat Area Overflow Interrupt Enable	0: Disables an interrupt request for an extended repeat area overflow on the destination address 1: Enables an interrupt request for an extended repeat area overflow on the destination address	R/W
b1	SARIE	Source Address Extended Repeat Area Overflow Interrupt Enable	0: Disables an interrupt request for an extended repeat area overflow on the source address 1: Enables an interrupt request for an extended repeat area overflow on the source address	R/W
b2	RPTIE	Repeat Size End Interrupt Enable	0: Disables the repeat size end interrupt request. 1: Enables the repeat size end interrupt request.	R/W
b3	ESIE	Transfer Escape End Interrupt Enable	0: Disables the transfer escape end interrupt request. 1: Enables the transfer escape end interrupt request.	R/W
b4	DTIE	Transfer End Interrupt Enable	0: Disables the transfer end interrupt request. 1: Enables the transfer end interrupt request.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

DARIE Bit (Destination Address Extended Repeat Area Overflow Interrupt Enable)

When an extended repeat area overflow on the destination address occurs while this bit is set to 1, the DTE bit in DMCNT is set to 0. At the same time, the ESIF flag in DMSTS is set to 1 to indicate that an interrupt by an extended repeat area overflow on the destination address is requested.

When block transfer mode is used with the extended repeat area function, an interrupt is requested after completion of a 1-block size transfer. When setting 1 in the DTE bit in DMACm.DMCNT of the channel for which a transfer has been stopped, the transfer is resumed from the state when the transfer is stopped.

When the extended repeat area is not specified for the destination address, this bit is ignored.

SARIE Bit (Source Address Extended Repeat Area Overflow Interrupt Enable)

When an extended repeat area overflow on the source address occurs while this bit is set to 1, the DTE bit in DMCNT is set to 0. At the same time, the ESIF flag in DMSTS is set to 1 to indicate that an interrupt by an extended repeat area overflow on the source address is requested.

When block transfer mode is used with the extended repeat area function, an interrupt is requested after completion of a 1-block size transfer. When setting 1 in the DTE bit in DMACm.DMCNT of the channel for which a transfer has been stopped, the transfer is resumed from the state when the transfer is stopped.

When the extended repeat area is not specified for the source address, this bit is ignored.

RPTIE Bit (Repeat Size End Interrupt Enable)

When this bit is set to 1 in repeat transfer mode, the DTE bit in DMCNT is set to 0 after completion of a 1-repeat size data transfer. At the same time, the ESIF flag in DMSTS is set to 1 to indicate that the repeat size end interrupt request has been generated. The repeat size end interrupt request can be generated even when the DTS[1:0] bits in DMTMD are 10b (= repeat area or block area is not specified).

When this bit is set to 1 in block transfer mode, the DTE bit in DMCNT is set to 0 after completion of a 1-block data transfer in the same way as repeat transfer mode. At the same time, the ESIF flag in DMSTS is set to 1 to indicate that the repeat size end interrupt request has been generated. The repeat size end interrupt request can be generated even when the DTS[1:0] bits in DMTMD are 10b (= repeat area or block area is not specified).

ESIE Bit (Transfer Escape End Interrupt Enable)

This bit enables or disables the transfer escape end interrupt requests (repeat size end interrupt request and extended repeat area overflow interrupt request) that are generated during DMA transfer.

The transfer escape end interrupt is generated when the ESIF flag in DMSTS is set to 1 with this bit set to 1. The transfer escape end interrupt is cleared by setting this bit or the ESIF flag in DMSTS to 0.

DTIE Bit (Transfer End Interrupt Enable)

This bit enables or disables the transfer end interrupt request to be generated on completion of a specified number of data transfers.

The transfer end interrupt is generated when the DTIF bit in DMSTS is set to 1 with this bit set to 1. The transfer end interrupt is cleared by setting this bit or the DTIF bit in DMSTS to 0.

17.2.7 DMA Address Mode Register (DMAMD)

Address(es): DMAC0.DMAMD 0008 2014h, DMAC1.DMAMD 0008 2054h, DMAC2.DMAMD 0008 2094h, DMAC3.DMAMD 0008 20D4h, DMAC4.DMAMD 0008 2114h, DMAC5.DMAMD 0008 2154h, DMAC6.DMAMD 0008 2194h, DMAC7.DMAMD 0008 21D4h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SM[1:0]		—	SARA[4:0]					DM[1:0]		—	DARA[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	DARA[4:0]	Destination Address Extended Repeat Area	Specifies the extended repeat area on the destination address. For details on the settings, see Table 17.2.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7, b6	DM[1:0]	Destination Address Update Mode	b7 b6 0 0: Destination address is fixed. 0 1: Offset addition*1 1 0: Destination address is incremented. 1 1: Destination address is decremented.	R/W
b12 to b8	SARA[4:0]	Source Address Extended Repeat Area	Specifies the extended repeat area on the source address. For details on the settings, see Table 17.2.	R/W
b13	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b15, b14	SM[1:0]	Source Address Update Mode	b15 b14 0 0: Source address is fixed. 0 1: Offset addition*1 1 0: Source address is incremented. 1 1: Source address is decremented.	R/W

Note 1. Offset addition can be specified only for DMAC0.

DARA[4:0] Bits (Destination Address Extended Repeat Area)

These bits specify the extended repeat area on the destination address. The extended repeat area function is realized by updating the specified lower address bits with the remaining upper address bits fixed. The size of the extended repeat area can be any power of two between 2 bytes and 128 Mbytes.

When the lower address overflows the extended repeat area by address increment, the start address of the extended repeat area is set. Similarly, when the lower address underflows the extended repeat area by address decrement, the end address of the extended repeat area is set.

When the repeat area or block area is specified as a transfer destination, do not specify the extended repeat area on the destination address. When repeat transfer or block transfer is selected, and when DMACm.DMTMD.DTS[1:0] = 00b (the transfer destination is specified as the repeat area or block area), write 00000b in the DARA[4:0] bits.

An interrupt can be requested when an overflow or underflow occurs in the extended repeat area with the DARIE bit in DMINT set to 1. Table 17.2 lists the settings and the corresponding extended repeat areas.

DM[1:0] Bits (Destination Address Update Mode)

These bits select the mode of updating the destination address.

When increment is selected and the SZ[1:0] bits in DMTMD are set to 00b, 01b, and 10b, the destination address is incremented by 1, 2, and 4, respectively.

When decrement is selected and the SZ[1:0] bits in DMTMD are set to 00b, 01b, and 10b, the destination address is decremented by 1, 2, and 4, respectively.

When offset addition is selected, the offset specified by the DMAC0.DMOFR register is added to the address.

Offset addition can be specified only for DMAC0.

SARA[4:0] Bits (Source Address Extended Repeat Area)

These bits specify the extended repeat area on the source address. The extended repeat area function is realized by updating the specified lower address bits with the remaining upper address bits fixed. The size of the extended repeat area can be any power of two between 2 bytes and 128 Mbytes.

When the lower address overflows the extended repeat area by address increment, the start address of the extended repeat area is set. Similarly, when the lower address underflows the extended repeat area by address decrement, the end address of the extended repeat area is set.

When the repeat area or block area is specified as a transfer source, do not specify the extended repeat area on the source address. When repeat transfer or block transfer is selected, and when DMACm.DMTMD.DTS[1:0] = 01b (the transfer source is specified as the repeat area or block area), write 00000b in the SARA[4:0] bits.

An interrupt can be requested when an overflow or underflow occurs in the extended repeat area with the SARIE bit in DMINT set to 1. Table 17.2 lists the settings and the corresponding extended repeat areas.

SM[1:0] (Source Address Update Mode)

These bits select the mode of updating the source address.

When increment is selected and the SZ[1:0] bits in DMTMD are set to 00b, 01b, and 10b, the source address is incremented by 1, 2, and 4, respectively.

When decrement is selected and the SZ[1:0] bits in DMTMD are set to 00b, 01b, and 10b, the source address is decremented by 1, 2, and 4, respectively.

When offset addition is selected, the offset specified by the DMAC0.DMOFR register is added to the address. Offset addition can be specified only for DMAC0.

Table 17.2 SARA[4:0] or DARA[4:0] Settings and Corresponding Repeat Areas

SARA[4:0] or DARA[4:0]	Extended Repeat Area
00000b	Not specified
00001b	2 bytes specified as extended repeat area by the lower 1 bit of the address
00010b	4 bytes specified as extended repeat area by the lower 2 bits of the address
00011b	8 bytes specified as extended repeat area by the lower 3 bits of the address
00100b	16 bytes specified as extended repeat area by the lower 4 bits of the address
00101b	32 bytes specified as extended repeat area by the lower 5 bits of the address
00110b	64 bytes specified as extended repeat area by the lower 6 bits of the address
00111b	128 bytes specified as extended repeat area by the lower 7 bits of the address
01000b	256 bytes specified as extended repeat area by the lower 8 bits of the address
01001b	512 bytes specified as extended repeat area by the lower 9 bits of the address
01010b	1 Kbyte specified as extended repeat area by the lower 10 bits of the address
01011b	2 Kbytes specified as extended repeat area by the lower 11 bits of the address
01100b	4 Kbytes specified as extended repeat area by the lower 12 bits of the address
01101b	8 Kbytes specified as extended repeat area by the lower 13 bits of the address
01110b	16 Kbytes specified as extended repeat area by the lower 14 bits of the address
01111b	32 Kbytes specified as extended repeat area by the lower 15 bits of the address
10000b	64 Kbytes specified as extended repeat area by the lower 16 bits of the address
10001b	128 Kbytes specified as extended repeat area by the lower 17 bits of the address
10010b	256 Kbytes specified as extended repeat area by the lower 18 bits of the address
10011b	512 Kbytes specified as extended repeat area by the lower 19 bits of the address
10100b	1 Mbyte specified as extended repeat area by the lower 20 bits of the address
10101b	2 Mbytes specified as extended repeat area by the lower 21 bits of the address
10110b	4 Mbytes specified as extended repeat area by the lower 22 bits of the address
10111b	8 Mbytes specified as extended repeat area by the lower 23 bits of the address
11000b	16 Mbytes specified as extended repeat area by the lower 24 bits of the address
11001b	32 Mbytes specified as extended repeat area by the lower 25 bits of the address
11010b	64 Mbytes specified as extended repeat area by the lower 26 bits of the address
11011b	128 Mbytes specified as extended repeat area by the lower 27 bits of the address
11100b to 11111b	Setting prohibited.

17.2.8 DMA Offset Register (DMOFR)

Address(es): DMAC0.DMOFR 0008 2018h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Description	Setting Range	R/W
b31 to b0	Specifies the offset when offset addition is selected as the address update mode for transfer source or destination.	0000 0000h to 00FF FFFFh (0 bytes to (16 M – 1) bytes) FF00 0000h to FFFF FFFFh (–16 Mbytes to –1 byte)	R/W

Write to this register while the DMAC operation is stopped or DMA transfer is disabled (not during data transfer).
Setting bits 31 to 25 is invalid; a value of bit 24 is extended to bits 31 to 25. Reading DMOFR returns the extended value.

17.2.9 DMA Transfer Enable Register (DMCNT)

Address(es): DMAC0.DMCNT 0008 201Ch, DMAC1.DMCNT 0008 205Ch, DMAC2.DMCNT 0008 209Ch, DMAC3.DMCNT 0008 20DCh, DMAC4.DMCNT 0008 211Ch, DMAC5.DMCNT 0008 215Ch, DMAC6.DMCNT 0008 219Ch, DMAC7.DMCNT 0008 21DCh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DTE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DTE	DMA Transfer Enable	0: Disables DMA transfer. 1: Enables DMA transfer.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

DTE Bit (DMA Transfer Enable)

When the DMST bit in DMAST is set to 1 (DMAC module start) and this bit is set to 1 (DMA transfer is enabled), DMA transfer can be started for the corresponding channel.

[Setting condition]

- When 1 is written to this bit.

[Clearing conditions]

- When 0 is written to this bit.
- When the specified total volume of data transfer is completed.
- When DMA transfer is stopped by the repeat size end interrupt.
- When DMA transfer is stopped by the extended repeat area overflow interrupt.

17.2.10 DMA Software Start Register (DMREQ)

Address(es): DMAC0.DMREQ 0008 201Dh, DMAC1.DMREQ 0008 205Dh, DMAC2.DMREQ 0008 209Dh, DMAC3.DMREQ 0008 20DDh, DMAC4.DMREQ 0008 211Dh, DMAC5.DMREQ 0008 215Dh, DMAC6.DMREQ 0008 219Dh, DMAC7.DMREQ 0008 21DDh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	CLRS	—	—	—	SWREQ
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SWREQ	DMA Software Start	0: DMA transfer is not requested. 1: DMA transfer is requested.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	CLRS	DMA Software Start Bit Auto Clear Select	0: SWREQ bit is cleared after DMA transfer is started by software. 1: SWREQ bit is not cleared after DMA transfer is started by software.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SWREQ Bit (DMA Software Start)

When 1 is written to this bit, a DMA transfer request is generated. After DMA transfer is started in response to the request, this bit is set to 0 if the CLRS bit is set to 0. This bit is not set to 0 while the CLRS bit is set to 1. In this case, a DMA transfer request can be issued again after completion of a transfer.

Note that, however, setting this bit is valid and DMA transfer by software is enabled only when the DCTG[1:0] bits in DMTMD are set to 00b (DMA request source is software).

Setting this bit is invalid when the DCTG[1:0] bits in DMTMD are set to a value other than 00b.

To start DMA transfer by software with the CLRS bit being 0, ensure that the SWREQ bit is 0, and then write 1 to the SWREQ bit.

[Setting condition]

- When 1 is written to this bit.

[Clearing conditions]

- When a DMA transfer request by software is accepted and DMA transfer is started while the CLRS bit is set to 0 (the SWREQ bit is cleared after DMA transfer is started by software).
- When 0 is written to this bit.

CLRS Bit (DMA Software Start Bit Auto Clear Select)

This bit specifies whether to clear the SWREQ bit to 0 after DMA transfer is started in response to the DMA transfer request generated by setting the SWREQ bit to 1. With this bit set to 0, the SWREQ bit is set to 0 after DMA transfer is started. With this bit set to 1, the SWREQ bit is not set to 0. In this case, a DMA transfer request can be issued again after completion of a transfer.

17.2.11 DMA Status Register (DMSTS)

Address(es): DMAC0.DMSTS 0008 201Eh, DMAC1.DMSTS 0008 205Eh, DMAC2.DMSTS 0008 209Eh, DMAC3.DMSTS 0008 20DEh, DMAC4.DMSTS 0008 211Eh, DMAC5.DMSTS 0008 215Eh, DMAC6.DMSTS 0008 219Eh, DMAC7.DMSTS 0008 21DEh

	b7	b6	b5	b4	b3	b2	b1	b0
	ACT	—	—	DTIF	—	—	—	ESIF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ESIF	Transfer Escape End Interrupt Flag	0: A transfer escape end interrupt has not been generated. 1: A transfer escape end interrupt has been generated.	R/W*1
b3 to b1	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b4	DTIF	Transfer End Interrupt Flag	0: A transfer end interrupt has not been generated. 1: A transfer end interrupt has been generated.	R/W*1
b6, b5	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b7	ACT	DMA Active Flag	0: DMAC operation is suspended. 1: DMAC is operating.	R

Note 1. Only 0 can be written to clear the flag.

ESIF Flag (Transfer Escape End Interrupt Flag)

This flag indicates that the transfer escape end interrupt has been generated.

[Setting conditions]

- When 1-repeat size data transfer is completed in repeat transfer mode with the RPTIE bit in DMINT set to 1.
- When 1-block data transfer is completed in block transfer mode with the RPTIE bit in DMINT set to 1.
- When an extended repeat area overflow on the source address occurs while the SARIE bit in DMINT is set to 1 and the SARA[4:0] bits in DMAMD are set to a value other than 00000b (extended repeat area is specified on the transfer source address)
- When an extended repeat area overflow on the destination address occurs while the DARIE bit in DMINT is set to 1 and the DARA[4:0] bits in DMAMD are set to a value other than 00000b (extended repeat area is specified on the transfer destination address)

[Clearing conditions]

- When 0 is written to this bit.
- When 1 is written to the DTE bit in DMCNT.

DTIF Flag (Transfer End Interrupt Flag)

This flag indicates that the transfer end interrupt has been generated.
[Setting conditions]

- When the specified number of unit-transfers are completed in normal transfer mode (the value of DMCRAL becoming 0 on completion of transfer)
- When the specified number of repeat transfer operations are completed in repeat transfer mode (the value of DMCRB becoming 0 on completion of transfer)
- When the specified number of blocks have been transferred in block transfer mode (the value of DMCRB becoming 0 on completion of transfer)

[Clearing conditions]

- When 0 is written to this bit
- When 1 is written to the DTE bit in DMCNT

ACT Flag (DMA Active Flag)

This flag indicates whether the DMAC is in the idle or active state.
[Setting condition]

- When the DMAC starts data transfer operation

[Clearing condition]

- When data transfer in response to one transfer request is completed

17.2.12 DMA Request Source Flag Control Register (DMCSL)

Address(es): DMAC0.DMCSL 0008 201Fh, DMAC1.DMCSL 0008 205Fh, DMAC2.DMCSL 0008 209Fh, DMAC3.DMCSL 0008 20DFh, DMAC4.DMCSL 0008 211Fh, DMAC5.DMCSL 0008 215Fh, DMAC6.DMCSL 0008 219Fh, DMAC7.DMCSL 0008 21DFh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DISEL
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DISEL	Interrupt Select	0: At the beginning of transfer, clear the interrupt status flag of the request source to 0. 1: At the end of transfer, the interrupt status flag of the request source issues an interrupt to the CPU.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

DISEL Bit (Interrupt Select)

This bit selects whether the interrupt status flag of the request source is set to 0 or issues an interrupt request to the CPU, at the beginning of DMA transfer.
When DMTMD.DCTG[1:0] = 00b (trigger by software), the setting of the DISEL bit does not affect the operation.

17.2.13 DMAC Module Start Register (DMAST)

Address(es): 0008 2200h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DMST
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DMST	DMAC Module Start	0: DMAC module stop 1: DMAC module start	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

DMST Bit (DMAC Module Start)

When this bit is set to 1, DMAC is ready to accept transfer requests for all channels.

When 1 is written to the DMACm.DMCNT.DTE bit (DMA transfer is enabled) of multiple channels and then this bit is set to 1 (DMAC module start), the corresponding multiple channels can be placed in the transfer request acceptable state at the same time.

When the DMST bit is set to 0 during DMA transfer, DMA transfer is suspended after completion of the current data transfer corresponding to a single transfer request. DMA transfer can be resumed by setting the DMST bit to 1 again.

[Setting condition]

- When 1 is written to this bit

[Clearing condition]

- When 0 is written to this bit

17.2.14 DMAC74 Interrupt Status Monitor Register (DMIST)

Address(es): 0008 2204h

	b7	b6	b5	b4	b3	b2	b1	b0
	DMIS7	DMIS6	DMIS5	DMIS4	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b4	DMIS4	DMAC4 Interrupt Status Flag	0: DMAC4 interrupt is not requested. 1: DMAC4 interrupt is requested.	R
b5	DMIS5	DMAC5 Interrupt Status Flag	0: DMAC5 interrupt is not requested. 1: DMAC5 interrupt is requested.	R
b6	DMIS6	DMAC6 Interrupt Status Flag	0: DMAC6 interrupt is not requested. 1: DMAC6 interrupt is requested.	R
b7	DMIS7	DMAC7 Interrupt Status Flag	0: DMAC7 interrupt is not requested. 1: DMAC7 interrupt is requested.	R

DMISm Flag (DMACm Interrupt Status Flag) (m = 4 to 7)

This bit monitors the DMACm interrupt request. Writing to this bit will be ignored.

While the DMACm.DMINT.DTIE bit is 1 and the DMACm.DMSTS.DTIF bit is 1, or the DMACm.DMINT.ESIE bit is 1 and the DMACm.DMSTS.ESIF bit is 1, the DMIST.DMISm bit is set to 1.

17.3 Operation

17.3.1 Transfer Mode

(1) Normal Transfer Mode

In normal transfer mode, one data is transferred by one transfer request. A maximum of 65535 can be set as the number of transfer operations using the DMCRAL of DMACm. When these bits are set to 0000h, no specific number of transfer operations is set; data transfer is performed with the transfer counter stopped (free running mode). Setting DMCRB of DMACm is invalid in normal transfer mode. Except in free running mode, a transfer end interrupt request can be generated after completion of the specified number of transfer operations.
Table 17.3 summarizes the register update operation in normal transfer mode, and Figure 17.2 shows the operation in normal transfer mode.

Table 17.3 Register Update Operation in Normal Transfer Mode

Register	Function	Update Operation after Completion of a Transfer by One Transfer Request
DMACm.DMSAR	Transfer source address	Increment/decrement/fixed/offset addition*1
DMACm.DMDAR	Transfer destination address	Increment/decrement/fixed/offset addition*1
DMACm.DMCRAL	Transfer counter	Decremented by one/not updated (in free running mode)
DMACm.DMCRAH	—	Not updated (Not used in normal transfer mode)
DMACm.DMCRB	—	Not updated (Not used in normal transfer mode)

Note 1. Offset addition can be specified only for DMAC0.

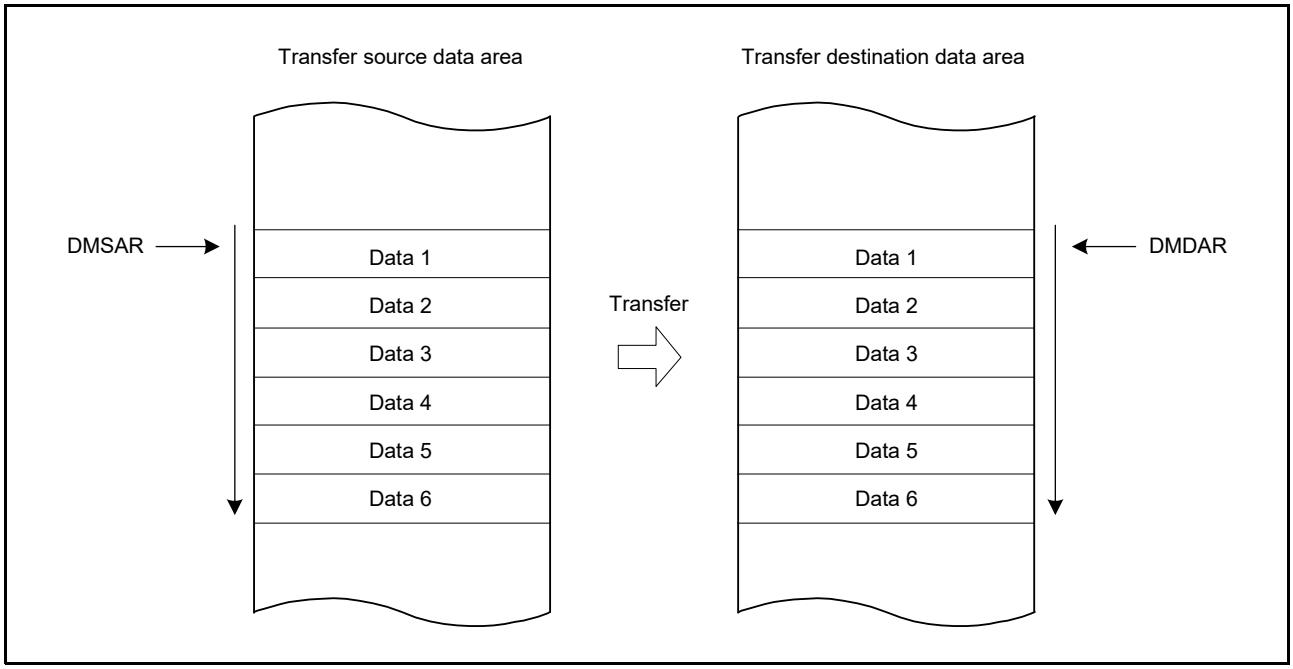


Figure 17.2 Operation in Normal Transfer Mode

(2) Repeat Transfer Mode

In repeat transfer mode, one data is transferred by one transfer request.
A maximum of 1K data can be set as a total repeat transfer size using DMCRA of the DMACm.
A maximum of 64K can be set as the number of repeat transfer operations using DMCRB of the DMACm; therefore, a maximum of 64M data (1K data × 64K counts of repeat transfer operations) can be set as a total data transfer size.
Either the transfer source or transfer destination can be specified as a repeat area. When transfer of the repeat size data is completed, the address of the specified repeat area (DMSAR or DMDAR of the DMACm) returns to the transfer start address. When data of the specified repeat size has all been transferred in repeat transfer mode, DMA transfer can be stopped and the repeat size end interrupt can be requested. DMA transfer can be resumed by writing 1 to the DTE bit in DMCNT of DMACm in the repeat size end interrupt handling.
A transfer end interrupt request can be generated after completion of the specified number of repeat transfer operations.
Table 17.4 summarizes the register update operation in repeat transfer mode, and Figure 17.3 shows the operation in repeat transfer mode.

Table 17.4 Register Update Operation in Repeat Transfer Mode

		Update Operation after Completion of a Transfer by One Transfer Request	
Register	**Function**	**When DMACm.DMCRAL is not 1**	**When DMACm.DMCRAL is 1 (Transfer of the Last Data in Repeat Size)**
DMACm.DMSAR	Transfer source address	Increment/decrement/fixed/offset addition*1	• DMACm.DMTMD.DTS[1:0] = 00b Increment/decrement/fixed/offset addition*1 • DMACm.DMTMD.DTS[1:0] = 01b Initial value of DMACm.DMSAR • DMACm.DMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition*1
DMACm.DMDAR	Transfer destination address	Increment/decrement/fixed/offset addition*1	• DMACm.DMTMD.DTS[1:0] = 00b Initial value of DMACm.DMDAR • DMACm.DMTMD.DTS[1:0] = 01b Increment/decrement/fixed/offset addition*1 • DMACm.DMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition*1
DMACm.DMCRAH	Repeat size	Not updated	Not updated
DMACm.DMCRAL	Transfer counter	Decremented by one	DMACm.DMCRAH
DMACm.DMCRB	Repeat-count counter	Not updated	Decremented by one

Note 1. Offset addition can be specified only for DMAC0.

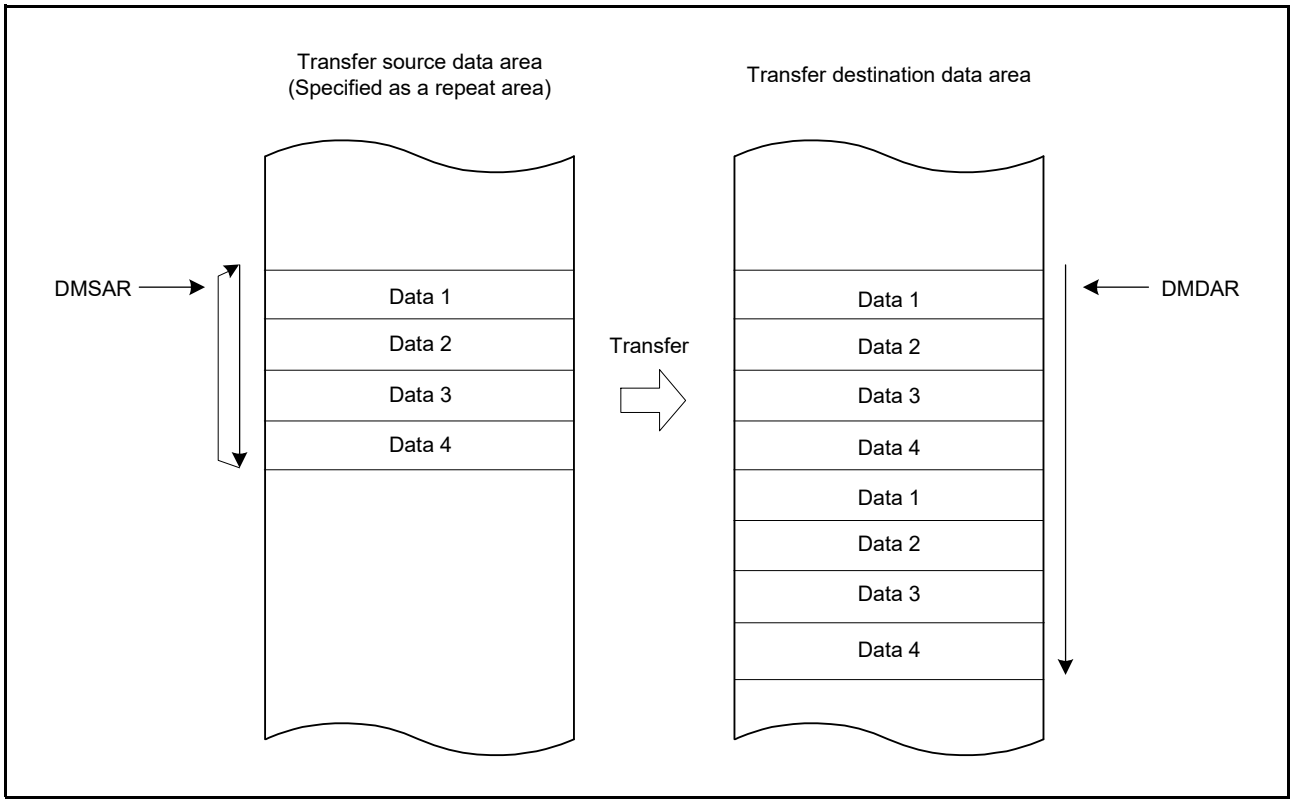


Figure 17.3 Operation in Repeat Transfer Mode

(3) Block Transfer Mode

In block transfer mode, a single block data is transferred by one transfer request.
A maximum of 1K data can be set as a total block transfer size using DMCRA of the DMACm.
A maximum of 64K can be set as the number of block transfer operations using DMCRB of the DMACm; therefore, a maximum of 64M data (1K data × 64K counts of block transfer operations) can be set as a total data transfer size.
Either the transfer source or transfer destination can be specified as a block area. When transfer of a single block data is completed, the address of the specified block area (DMSAR or DMDAR of the DMACm) returns to the transfer start address. When a single block data has all been transferred in block transfer mode, DMA transfer can be stopped and the repeat size end interrupt can be requested. DMA transfer can be resumed by writing 1 to the DTE bit in DMCNT of DMACm in the repeat size end interrupt handling.
Transfer end interrupt request can be generated after completion of the specified number of block transfer operations.
Table 17.5 summarizes the register update operation in block transfer mode, and Figure 17.4 shows the operation in block transfer mode.

Table 17.5 Register Update Operation in Block Transfer Mode

Register	Function	Update Operation after Completion of Single-Block Transfer by One Transfer Request
DMACm.DMSAR	Transfer source address	• DMACm.DMTMD.DTS[1:0] = 00b Increment/decrement/fixed/offset addition*1 • DMACm.DMTMD.DTS[1:0] = 01b Initial value of DMACm.DMSAR • DMACm.DMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition*1
DMACm.DMDAR	Transfer destination address	• DMACm.DMTMD.DTS[1:0] = 00 b Initial value of DMACm.DMDAR • DMACm.DMTMD.DTS[1:0] = 01 b Increment/decrement/fixed/offset addition*1 • DMACm.DMTMD.DTS[1:0] = 10 b Increment/decrement/fixed/offset addition*1
DMACm.DMCRAH	Block size	Not updated
DMACm.DMCRAL	Block size counter	DMACm.DMCRAH
DMACm.DMCRB	Transfer-block counter	Decremented by one

Note 1. Offset addition can be specified only for DMAC0.

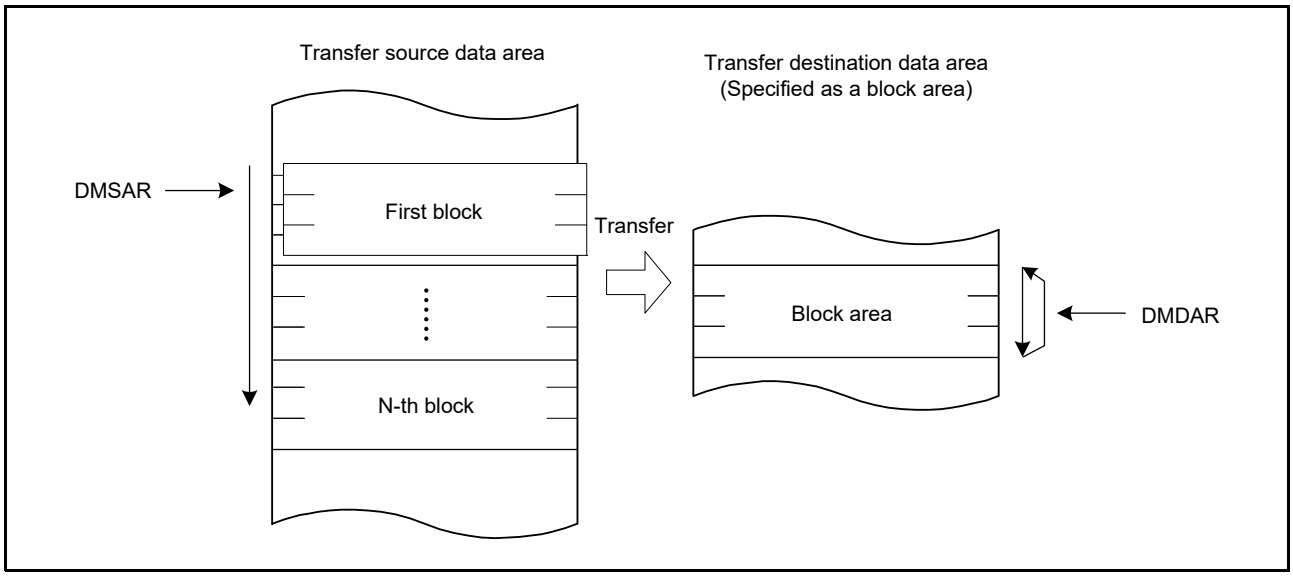


Figure 17.4 Operation in Block Transfer Mode

17.3.2 Extended Repeat Area Function

The DMAC supports a function to specify the extended repeat areas on the transfer source and destination addresses. With the extended repeat areas set, the address registers repeatedly indicate the addresses of the specified extended repeat areas.

The extended repeat areas can be specified separately to the transfer source address register (DMSAR) and transfer destination address register (DMDAR) of DMACm.

The extended repeat area on the source address is specified by the SARA[4:0] bits in DMAMD of DMACm. The extended repeat area on the destination address is specified by the DARA[4:0] bits in DMAMD of DMACm. The size can be specified separately for the source and destination sides.

However, the area (of transfer source or transfer destination) which is specified as the repeat area or block area should not be specified as the extended repeat area.

When the address register value reaches the end address of the extended repeat area and the extended repeat area overflows, DMA transfer is stopped and an interrupt by an extended repeat area overflow can be requested. When an overflow occurs in the extended repeat area on the transfer source while the SARIE bit in DMINT of DMACm is set to 1, the ESIF flag in DMSTS of DMACm is set to 1 and the DTE bit in DMCNT of DMACm is set to 0 to stop DMA transfer. At this time, if the ESIE bit in DMINT of DMACm is set to 1, an interrupt by an extended repeat area overflow is requested. When the DARIE bit in DMINT of DMACm is set to 1, the destination address register becomes a target to apply the function. DMA transfer can be resumed by writing 1 to the DTE bit in DMCNT of DMACm in the interrupt handling.

Figure 17.5 shows an example of the extended repeat area operation.

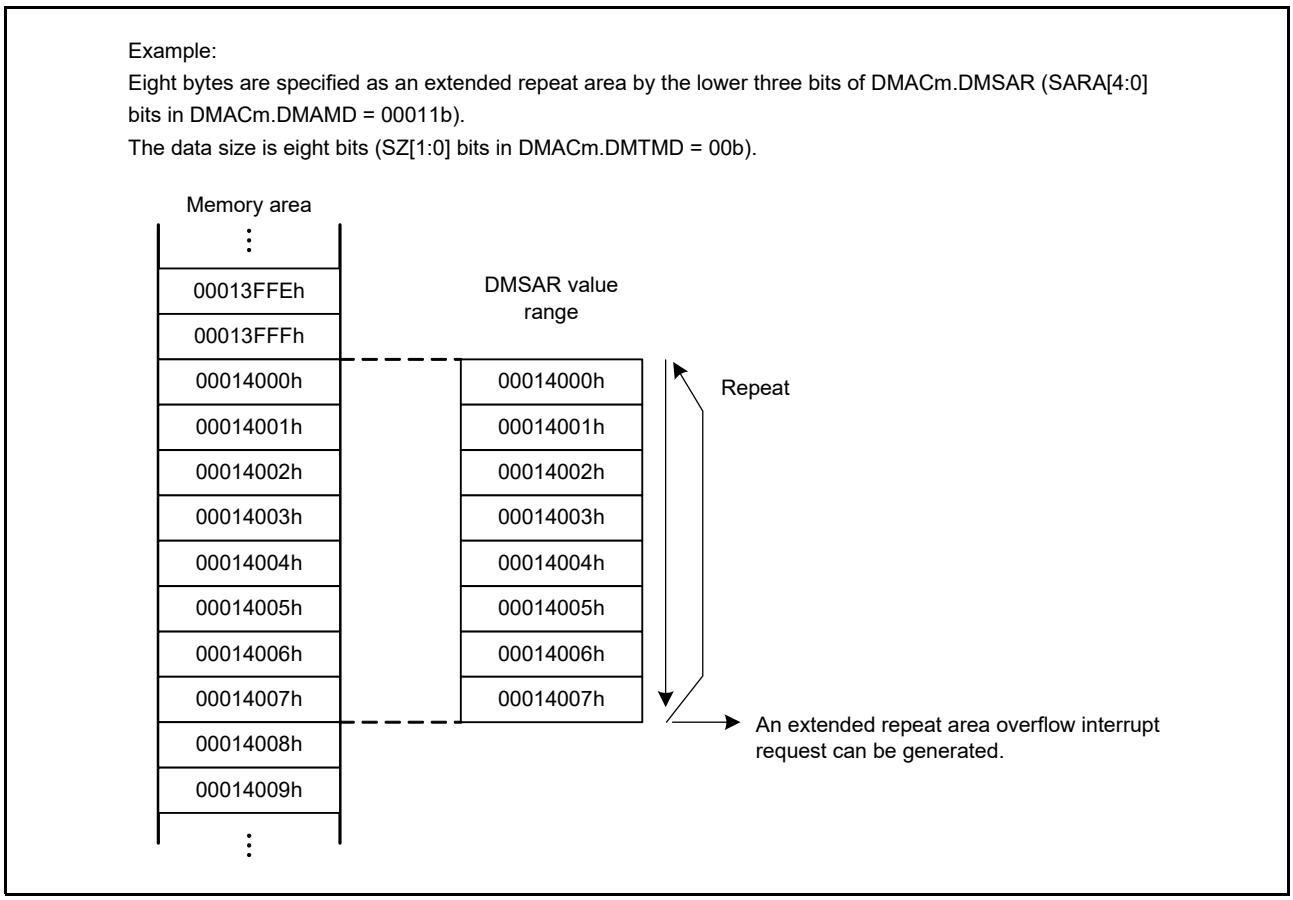


Figure 17.5 Example of Extended Repeat Area Operation

When an interrupt by an extended repeat area overflow is used in block transfer mode, the following should be taken into consideration.
When a transfer is stopped by an interrupt by an extended repeat area overflow, the address register must be set so that the block size is a power of 2 or the block size boundary is aligned with the extended repeat area boundary. When an overflow on the extended repeat area occurs during a transfer of one block, the interrupt by the overflow is suspended until transfer of the block is completed, and the transfer overruns.

Figure 17.6 shows an example when the extended repeat area function is used in block transfer mode.

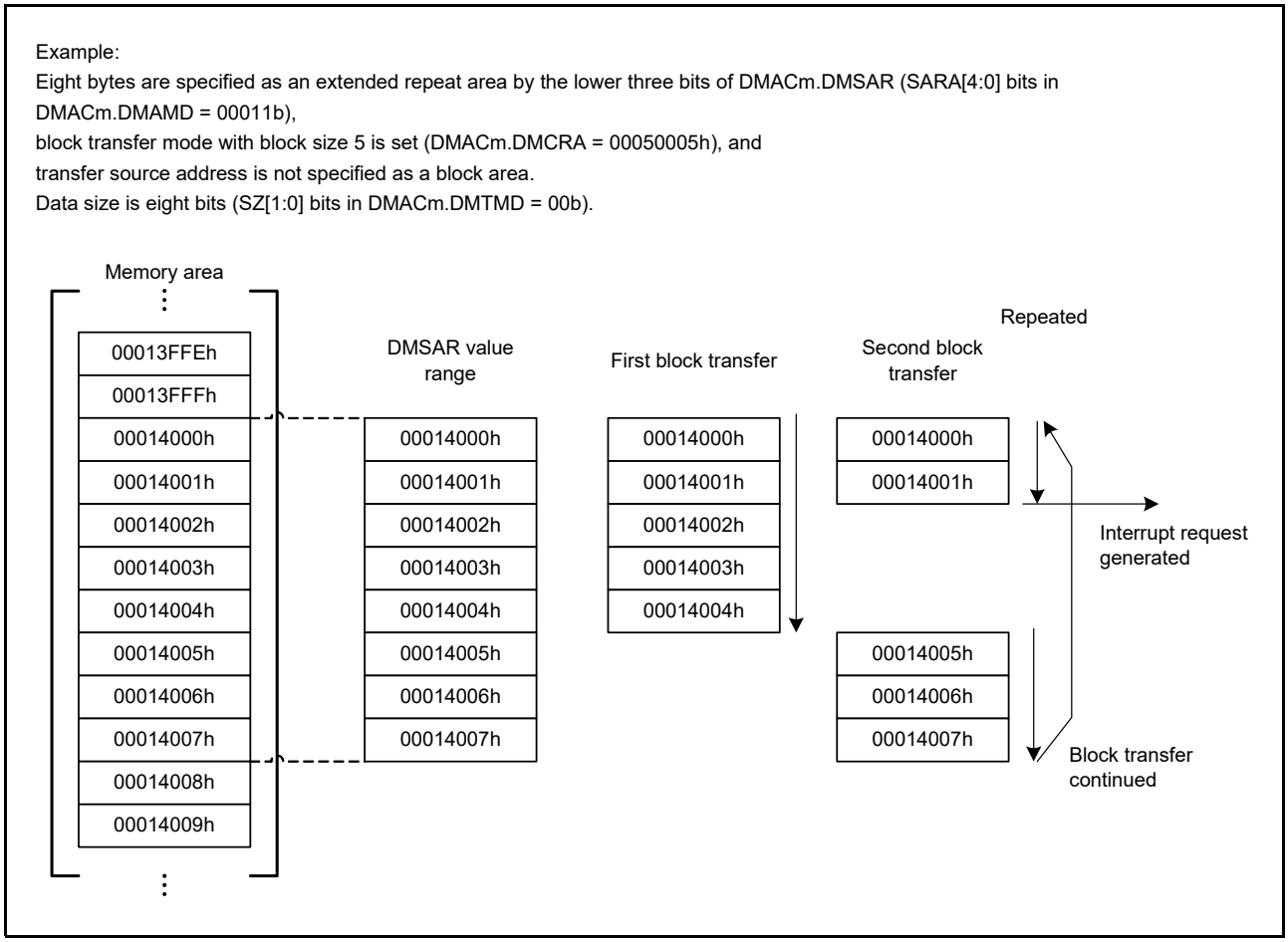


Figure 17.6 Example of Extended Repeat Area Function in Block Transfer Mode

17.3.3 Address Update Function using Offset

The source and destination addresses can be updated by fixing, increment, decrement, or offset addition. When the offset addition is selected, the offset specified by the DMA offset register (DMOFR of DMAC0) is added to the address every time the DMAC performs one data transfer. This function realizes a data transfer where addresses are allocated to separated areas.

Offset subtraction can also be realized by setting a negative value in DMOFR register of DMAC0. In this case, the negative value must be 2's complement.

Address update function using offset can be specified only for the DMAC0 channel.

Table 17.6 shows the address update method in each address update mode.

Table 17.6 Address Update Method in Each Address Update Mode

Address Update Mode	Settings of DMACm.DMAMD.SM[1:0] and DMACm.DMAMD.DM[1:0] for Address Update Modes	Address Update Method (for Different SZ[1:0] Settings in DMTMD of DMACm)		
		SZ[1:0] = 00b	SZ[1:0] = 01b	SZ[1:0] = 10b
Address fixed	00b	Fixed		
Offset addition	01b	+DMACm.DMOFR*1		
Increment	10b	+1	+2	+4
Decrement	11b	–1	–2	–4

Note 1. When setting a negative value in the DMA offset register, the value must be 2's complement. The 2's complement is obtained by the following formula.
2's complement of a negative offset value = ~ (offset) + 1 (~: bit inversion)

(1) Basic Transfer Using Offset Addition

Figure 17.7 shows an example of address updating using offset addition.

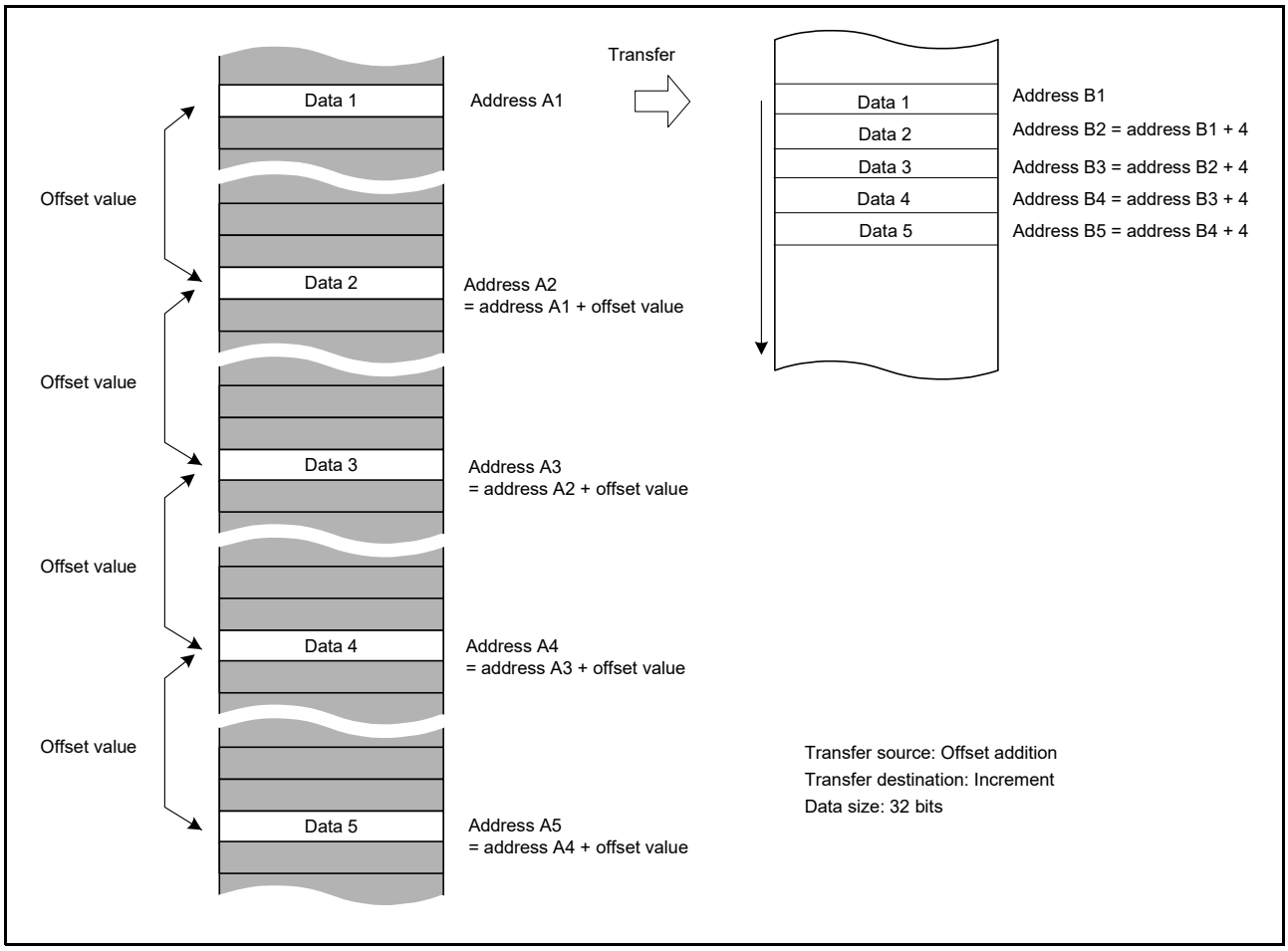


Figure 17.7 Example of Address Updating by Offset Addition

In Figure 17.7, the transfer data is 32 bits long, and offset addition and increment are set as the transfer source address update mode and transfer destination address update mode, respectively. The second and subsequent data is each read from the transfer source address obtained by adding the offset value to the previous address. The data read from the addresses at the specified intervals is written to the continuous locations on the destination.

(2) Example of XY Conversion Using Offset Addition

Figure 17.8 shows the XY conversion using offset addition in repeat transfer mode.
Settings are as follows:

- DMAC0.DMAMD: Transfer source address update mode: Offset addition
- DMAC0.DMAMD: Transfer destination address update mode: Destination address is incremented.
- DMAC0.DMTMD: Transfer data size select: 32 bits
- DMAC0.DMTMD: Transfer mode select: Repeat transfer
- DMAC0.DMTMD: Repeat area select: The source is specified as the repeat area.
- DMAC0.DMOFR: Offset address: 10h
- DMAC0.DMCRA: Repeat size: 4h
- DMAC0.DMINT: The repeat size end interrupt is enabled.

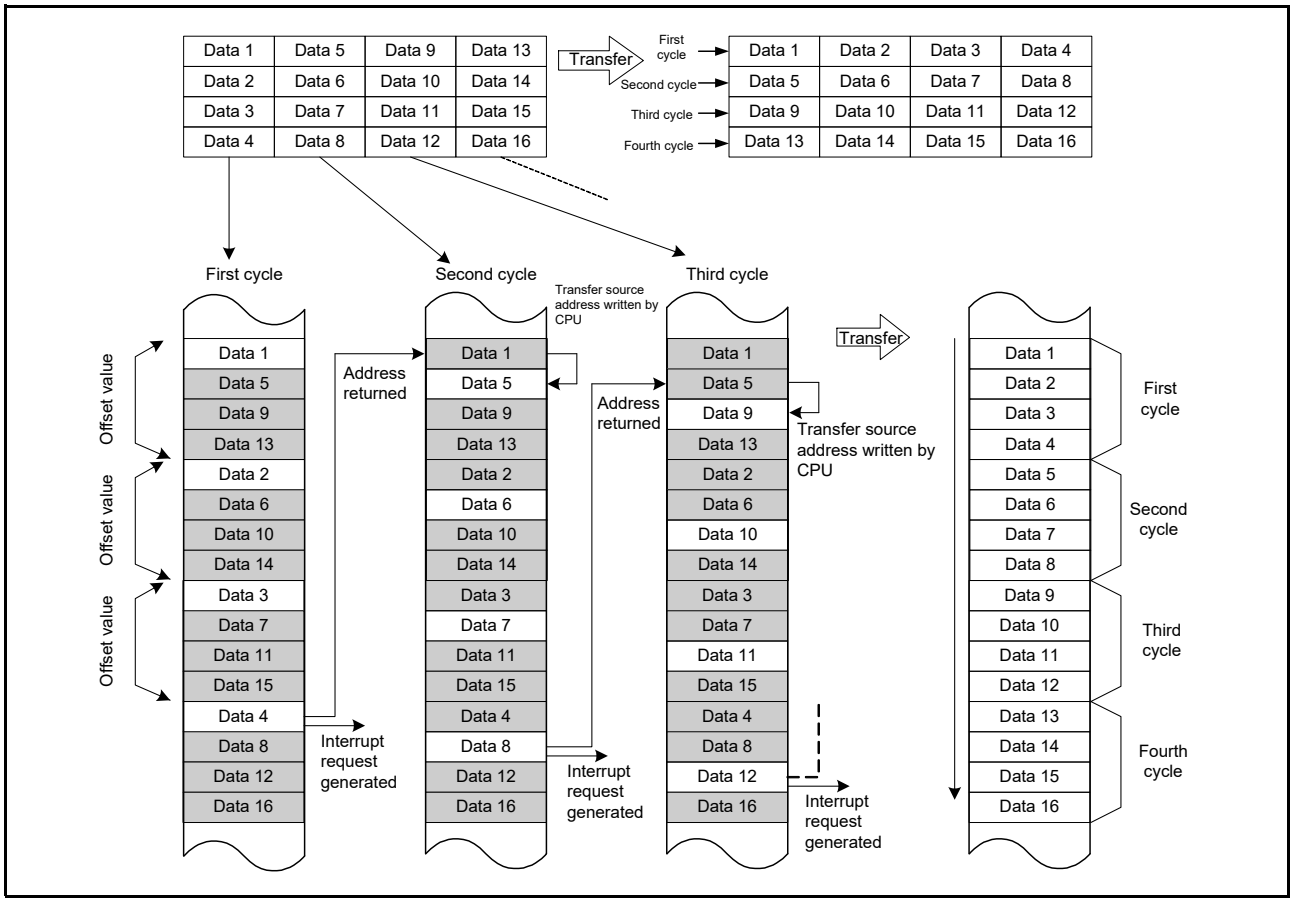


Figure 17.8 XY Conversion Operation Using Offset Addition in Repeat Transfer Mode

When a transfer starts, the offset value is added to the transfer source address every time data is transferred. The transfer data is written to the destination continuous addresses. When data 4 is transferred, which means that the repeat size of transfers is completed, the transfer source address returns to the transfer start address (address of data 1 on the transfer source) and a repeat size end interrupt is requested. While this interrupt stops the transfer temporarily, perform the following.

- DMAC0.DMSAR: Rewrite the DMA transfer source address to the address of data 5 (with the above example, the data 1 address + 4).
- DMAC0.DMCNT: Set the DTE bit to 1.

The DMA transfer is resumed from the state when the DMA transfer is stopped. After that, the operations described above are repeated until the transfer source data is transposed to the destination area (XY conversion).

Figure 17.9 shows a flowchart of the XY conversion.

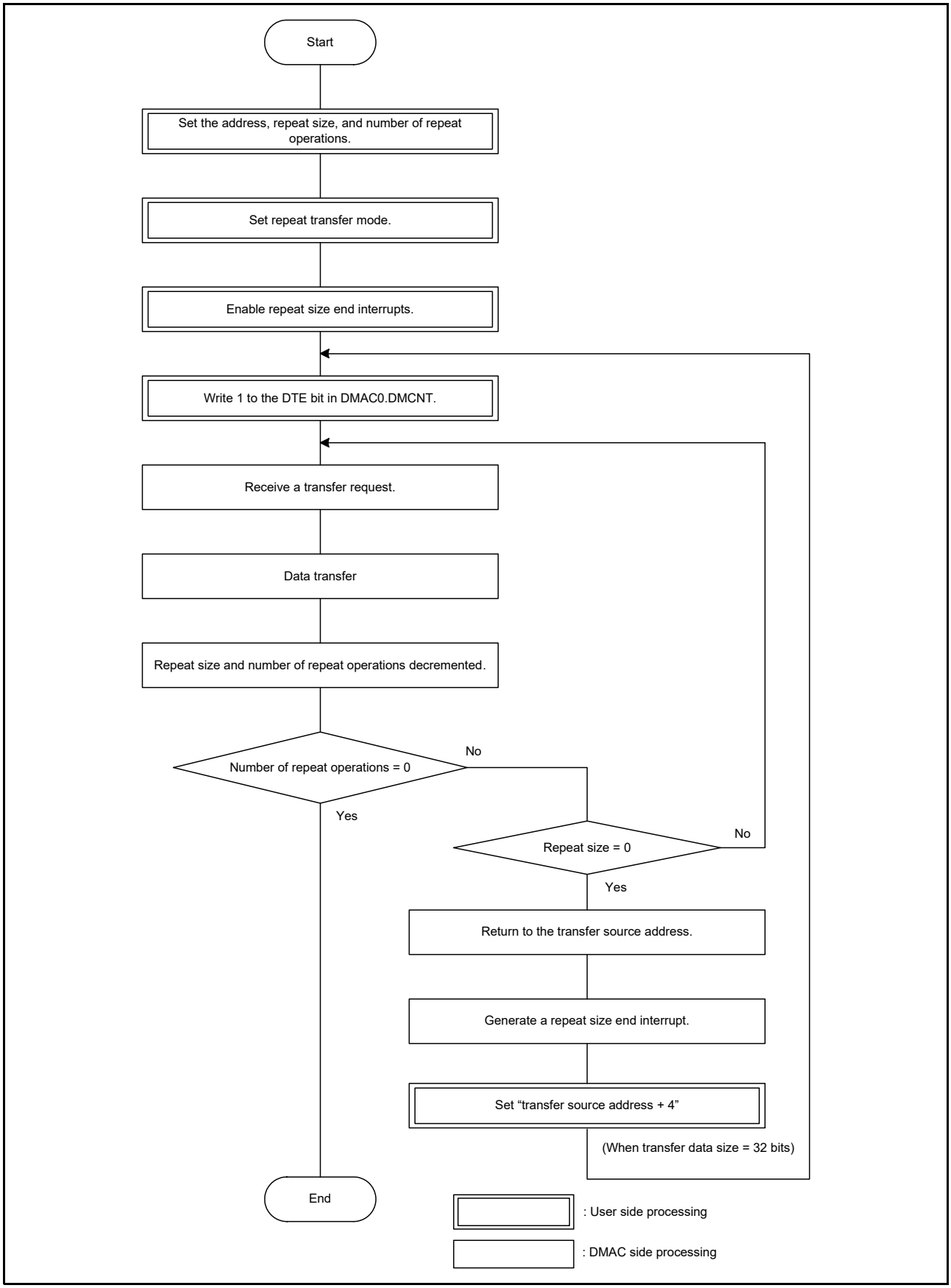


Figure 17.9 XY Conversion Flowchart Using Offset Addition in Repeat Transfer Mode

17.3.4 Request Sources

Software, the interrupt requests from the peripheral modules, and the external interrupt requests can be specified as the DMA request sources. Setting the DCTG[1:0] bits in DMTMD of DMACm selects the request source.

(1) Trigger by Software

Setting the DCTG[1:0] bits in DMTMD of DMACm to 00b enables the trigger by software.
To start DMA transfer by software, set the DCTG[1:0] bits in DMTMD of DMACm to 00b, and then set the DTE bit in DMCNT of DMACm to 1 (DMA transfer is enabled) and the SWREQ bit in DMREQ of DMACm to 1 (DMA transfer is requested) with the DMST bit in DMAST set to 1 (DMAC module start).
When the DMAC is triggered by software while the CLRS bit in DMREQ of DMACm is 0, the SWREQ bit in DMREQ of DMACm is set to 0 after data transfer is started in response to a DMA transfer request.
When the DMAC is triggered by software while the CLRS bit is 1, the SWREQ bit is not set to 0 after data transfer is started. In this case, a DMA transfer request is issued again after completion of a transfer.

(2) Trigger by Interrupt Requests from On-Chip Peripheral Modules or External Interrupt Requests

Interrupt requests from the on-chip peripheral modules and external interrupt requests can be specified as the DMA request sources. The request source can be selected separately for each channel using the DMRSRm registers (m = 0 to 7) of the ICU.
The DMA transfer is triggered when an interrupt request from the on-chip peripheral module or an external interrupt request is generated while the DCTG[1:0] bits in DMTMD of DMACm are set to 01b (interrupts from the peripheral modules and the external interrupt pins are selected), the DTE bit in DMCNT of DMACm is set to 1 (DMA transfer is enabled), and the DMST bit in DMAST is set to 1 (DMAC module start).

For interrupt requests specified as DMA request sources, refer to Table 14.4, Interrupt Vector Table, in section 14, Interrupt Controller (ICUG).

17.3.5 Operation Timing

Figure 17.10 and Figure 17.11 show DMAC operation timing examples.

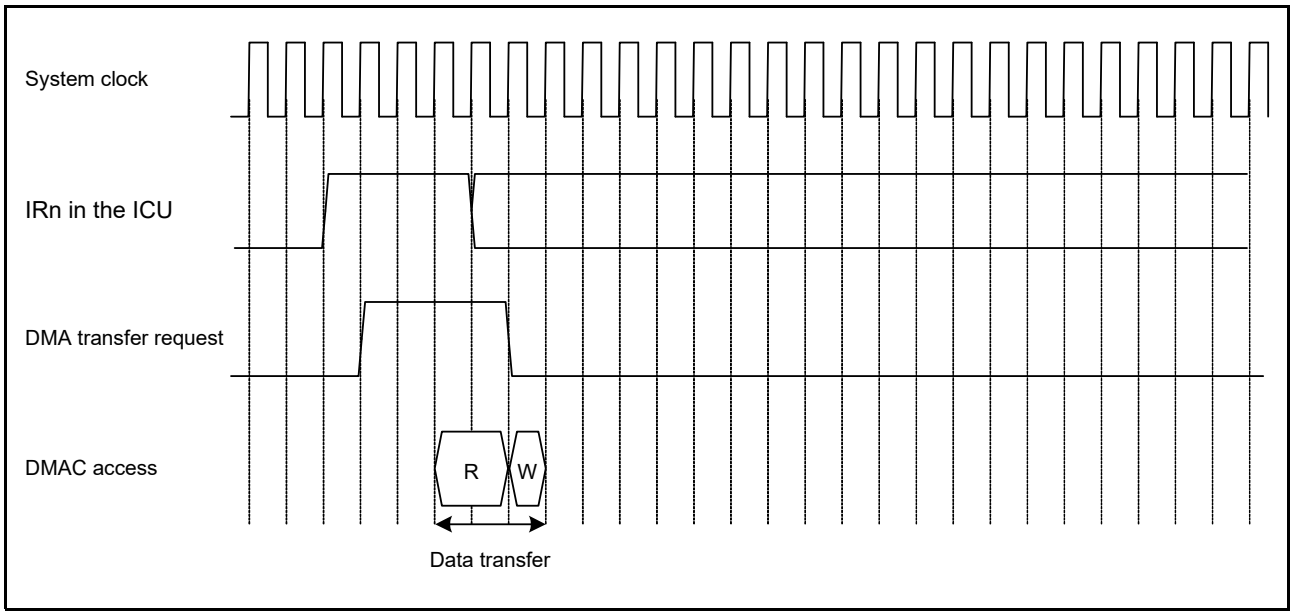


Figure 17.10 DMAC Operation Timing Example (1) (Trigger by Interrupt from Peripheral Module/External Interrupt Input Pin, Normal Transfer Mode, Repeat Transfer Mode)

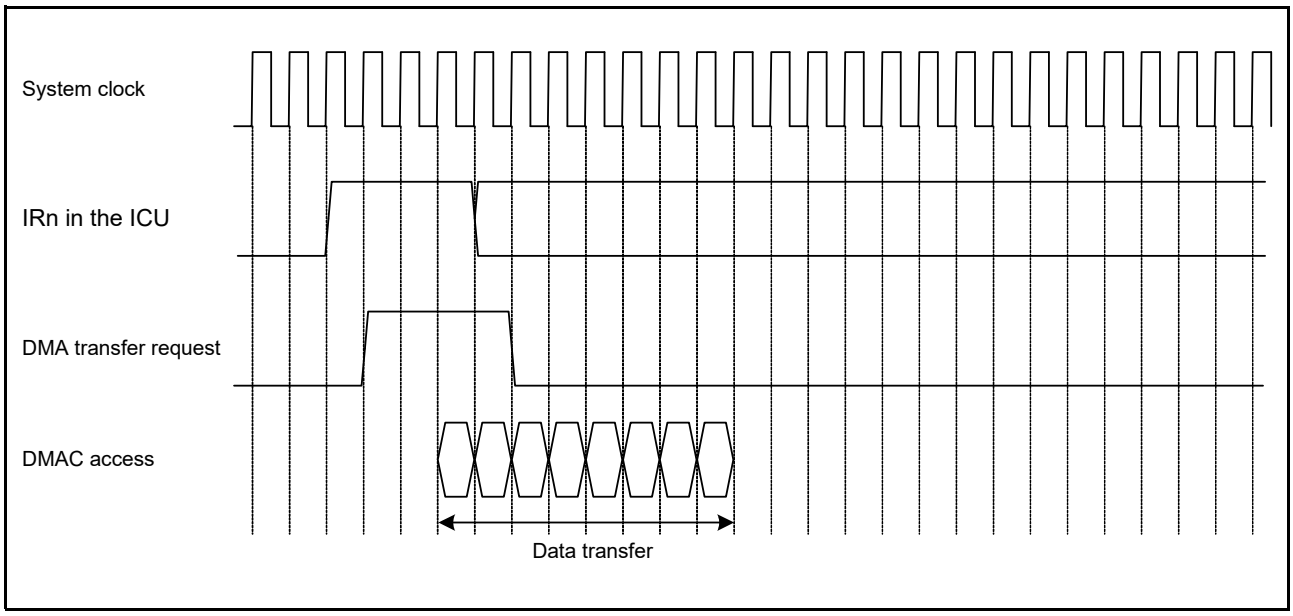


Figure 17.11 DMAC Operation Timing Example (2) (Trigger by Interrupt from Peripheral Module/External Interrupt Input Pin, Block Transfer Mode, Block Size = 4)

17.3.6 DMAC Execution Cycles

Table 17.7 lists execution cycles in one DMAC data transfer operation.

Table 17.7 DMAC Execution Cycles

Transfer Mode	Data Transfer (Read)	Data Transfer (Write)
Normal	Cr+1	Cw
Repeat	Cr+1	Cw
Block*1	P × Cr	P × Cw

Note 1. This is the case when the block size is 2 or more. When the block size is 1, normal transfer cycle is applied.

P: Block size (DMCRAH register setting)
Cr: Data read destination access cycle
Cw: Data write destination access cycle

Cr and Cw depend on the access destination. For the number of cycles for each access destination, see section 47, RAM, section 48, Flash Memory (FLASH), and section 5, I/O Registers.
The unit for +1 in “Data Transfer (Read)” column is one system clock cycle (ICLK).
For the operation example, see section 17.3.5, Operation Timing.

17.3.7 Activating the DMAC

Figure 17.12 shows the register setting procedure.

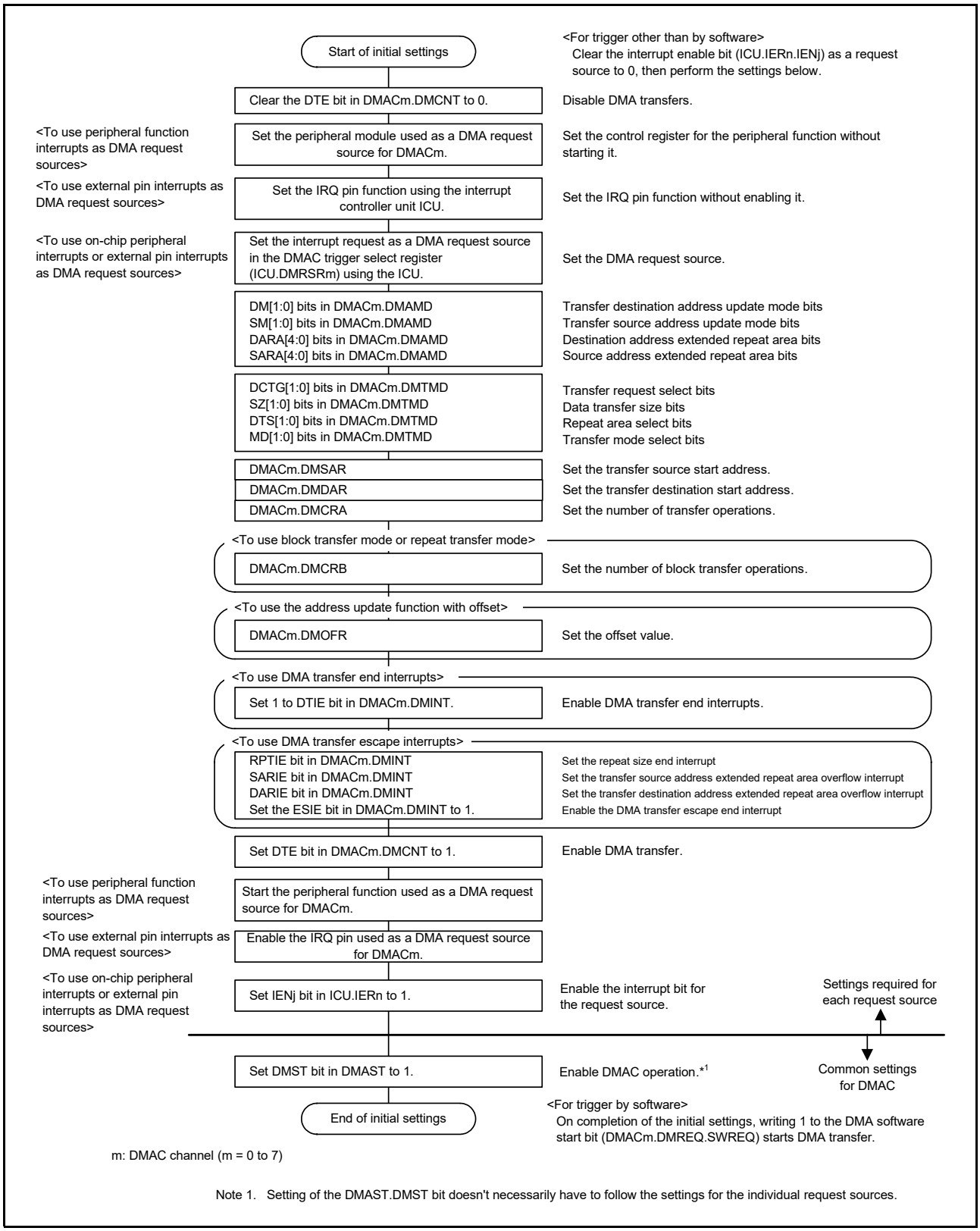


Figure 17.12 Register Setting Procedure

17.3.8 Starting DMA Transfer

Setting the DTE bit in DMCNT of DMACm to 1 (DMA transfer enabled) and setting the DMST bit in DMAST to 1 (DMAC module start) enable DMA transfer of channel m (m = 0 to 7).
Another transfer request cannot be accepted during the transfer of other DMAC channel or DTC. When the proceeding transfer is completed, channel arbitration is performed where a DMA transfer request of the highest priority channel is accepted and DMA transfer of the channel starts. When DMA transfer starts, the ACT bit in DMSTS of DMACm is set to 1 (the DMAC is in the active state).

17.3.9 Registers during DMA Transfer

The DMAC registers are updated by a DMA transfer. The value to be updated differs according to the other settings and the transfer state. The registers to be updated are DMSAR, DMDAR, DMCRA, DMCRB, DMCNT, and DMSTS of DMACm.

(1) DMA Source Address Register (DMACm.DMSAR)

When data has been transferred in response to one transfer request, the contents of DMSAR are updated to the address to be accessed by the next transfer request.
For details on register update operation in each transfer mode, refer to Table 17.3 to Table 17.5.

(2) DMA Destination Address Register (DMACm.DMDAR)

When data has been transferred in response to one transfer request, the contents of DMDAR are updated to the address to be accessed by the next transfer request.
For details on register update operation in each transfer mode, refer to Table 17.3 to Table 17.5.

(3) DMA Transfer Count Register (DMACm.DMCRA)

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected.
For details on register update operation in each transfer mode, refer to Table 17.3 to Table 17.5.

(4) DMA Block Transfer Count Register (DMACm.DMCRB)

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected.
For details on register update operation in each transfer mode, refer to Table 17.3 to Table 17.5.

(5) DMA Transfer Enable Bit (DMACm.DMCNT.DTE)

Although the DMACm.DMCNT.DTE bit enables or disables data transfer by the register write access, it is automatically set to 0 by the DMAC according to the DMA transfer state.
The conditions for clearing this bit by the DMAC are as follows:

- When the specified total volume of data transfer is completed
- When DMA transfer is stopped by the repeat size end interrupt
- When DMA transfer is stopped by the extended repeat area overflow interrupt

Writing to the registers for the channels when the corresponding DMACm.DMCNT.DTE bit is set to 1 is prohibited (except for DMACm.DMCNT). In this case, writing must be performed after the bit is set to 0.

(6) DMA Active Flag (DMACm.DMSTS.ACT)

The ACT bit in DMSTS of DMACm indicates whether the DMACm is in the idle or active state.
This flag is set to 1 when the DMAC starts data transfer, and is set to 0 when data transfer in response to one transfer request is completed.
Even when DMA transfer is stopped by writing 0 to the DTE bit in DMCNT of DMACm during DMA transfer, this flag remains 1 until DMA transfer is completed.

(7) Transfer End Interrupt Flag (DMACm.DMSTS.DTIF)

The DTIF flag in DMSTS of DMACm is set to 1 after DMA transfer of the total transfer size of data is completed.
When both this flag and the DTIE bit in DMINT of DMACm are set to 1, a transfer end interrupt is requested.
This flag is set to 1 when the DMA transfer bus cycle is completed and the ACT flag in DMSTS of DMACm is set to 0 indicating the DMA transfer end.
This flag is automatically set to 0 when the DTE bit in DMCNT of DMACm is set to 1 during the interrupt handling.

(8) Transfer Escape End Interrupt Flag (DMACm.DMSTS.ESIF)

The ESIF flag in DMSTS of DMACm is set to 1 when a repeat size end interrupt or extended repeat area overflow interrupt is requested. When this bit and the ESIE bit in DMINT of DMACm are set to 1, a transfer escape end interrupt is requested.
This flag is set to 1 when the bus cycle of the DMA transfer having caused the interrupt request is completed and the ACT flag in DMSTS of DMACm is set to 0 indicating the DMA transfer end.
This flag is automatically set to 0 when the DTE bit in DMCNT of DMACm is set to 1 during an interrupt handling.
Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set. For details, see section 14, Interrupt Controller (ICUG).

17.3.10 Channel Priority

When multiple DMA transfer requests are present, the DMAC determines the priority of channels that have DMA transfer requests.
The channel priority is fixed as channel 0 > channel 1 > channel 2 > channel 3 > channel 4 > channel 5 > channel 6 > channel 7 (channel 0: highest).
When a DMA transfer request is generated during data transfer, channel arbitration is started after the final data has been transferred, and DMA transfer of the higher-priority channel starts.

17.4 Ending DMA Transfer

The operation for ending DMA transfer depends on the transfer end conditions. When DMA transfer ends, the DTE bit in DMCNT and the ACT flag in DMSTS of DMACm are changed from 1 to 0, indicating that DMA transfer has ended.

17.4.1 Transfer End by Completion of Specified Total Number of Transfer Operations

(1) In Normal Transfer Mode (DMACm.DMTMD.MD[1:0] = 00b)

When the value of DMCRAL of DMACm changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DTE bit in DMCNT of DMACm is set to 0 and the DTIF bit in DMSTS of DMACm is set to 1 at the same time. If the DTIE bit in DMINT of DMACm is 1 at this time, a transfer end interrupt request is issued to the CPU or the DTC.

(2) In Repeat Transfer Mode (DMACm.DMTMD.MD[1:0] = 01b)

When the value of DMCRB of DMACm changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DTE bit in DMCNT of DMACm is set to 0 and the DTIF bit in DMSTS of DMACm is set to 1 at the same time. If the DTIE bit in DMINT of DMACm is 1 at this time, an interrupt request is issued to the CPU or the DTC.

(3) In Block Transfer Mode (DMACm.DMTMD.MD[1:0] = 10b)

When the value of DMCRB of DMACm changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DTE bit in DMCNT of DMACm is set to 0 and the DTIF bit in DMSTS of DMACm is set to 1 at the same time. If the DTIE bit in DMINT of DMACm is 1 at this time, an interrupt request is issued to the CPU or the DTC.
Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set. For details, see section 14, Interrupt Controller (ICUG).

17.4.2 Transfer End by Repeat Size End Interrupt

In repeat transfer mode, a repeat size end interrupt is requested when transfer of a 1-repeat size of data is completed while the RPTIE bit in DMINT of DMACm is set to 1. When the interrupt is requested to complete DMA transfer, the DTE bit in DMCNT of DMACm is set to 0 and the ESIF flag in DMSTS of DMACm is set to 1. If the ESIE bit in DMINT of DMACm is 1 at this time, an interrupt request is issued to the CPU or the DTC. Here, the transfer can be resumed by writing 1 to the DTE bit in DMCNT of DMACm.
A repeat size end interrupt can be requested also in block transfer mode. In block transfer mode, the interrupt is requested in the same way as in repeat transfer mode when transfer of a 1-block size data is completed.
Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set. For details, see section 14, Interrupt Controller (ICUG).

17.4.3 Transfer End by Interrupt on Extended Repeat Area Overflow

When an overflow on the extended repeat area occurs while the extended repeat area is specified and the SARIE or DARIE bit in DMINT of DMACm is set to 1, an interrupt by an extended repeat area overflow is requested. When the interrupt is requested, the DMA transfer is terminated, the DTE bit in DMCNT of DMACm is set to 0, and the ESIF flag in DMSTS of DMACm is set to 1. If the ESIE bit in DMINT of DMACm is 1 at this time, an interrupt request is issued to the CPU or the DTC.
Even if an interrupt by an extended repeat area overflow is requested during a read cycle, the following write cycle is performed.
In block transfer mode, even if an interrupt by an extended repeat area overflow is requested during a 1-block transfer, the remaining data in the block is transferred; transfer is terminated after a block transfer.
Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set. For details, see section 14, Interrupt Controller (ICUG).

17.5 Interrupts

Each DMAC channel can output an interrupt request to the CPU or the DTC after transfer in response to one request is completed. When the transfer destination is the on-chip peripheral bus, an interrupt request is generated upon completion of data write to the write buffer not to the actual transfer destination.

Table 17.8 lists the relation among the interrupt sources, the interrupt status flags, and the interrupt enable bits. Figure 17.13 shows the schematic logic diagram of interrupt outputs (DMAC0 to DMAC3). Figure 17.14 shows the schematic logic diagram of interrupt outputs (DMAC4 to DMAC7). Figure 17.15 shows the DMAC interrupt handling routine to resume/terminate DMA transfer.

Table 17.8 Relation among Interrupt Sources, Interrupt Status Flags, and Interrupt Enable Bits

Interrupt Sources		Interrupt Enable Bits	Interrupt Status Flags	Request Output Enable Bits
Transfer end		—	DMACm.DMSTS.DTIF	DMACm.DMINT.DTIE
Escape transfer end	Repeat size end	DMACm.DMINT.RPTIE	DMACm.DMSTS.ESIF	DMACm.DMINT.ESIE
	Source address extended repeat area overflow	DMACm.DMINT.SARIE		
	Destination address extended repeat area overflow	DMACm.DMINT.DARIE		

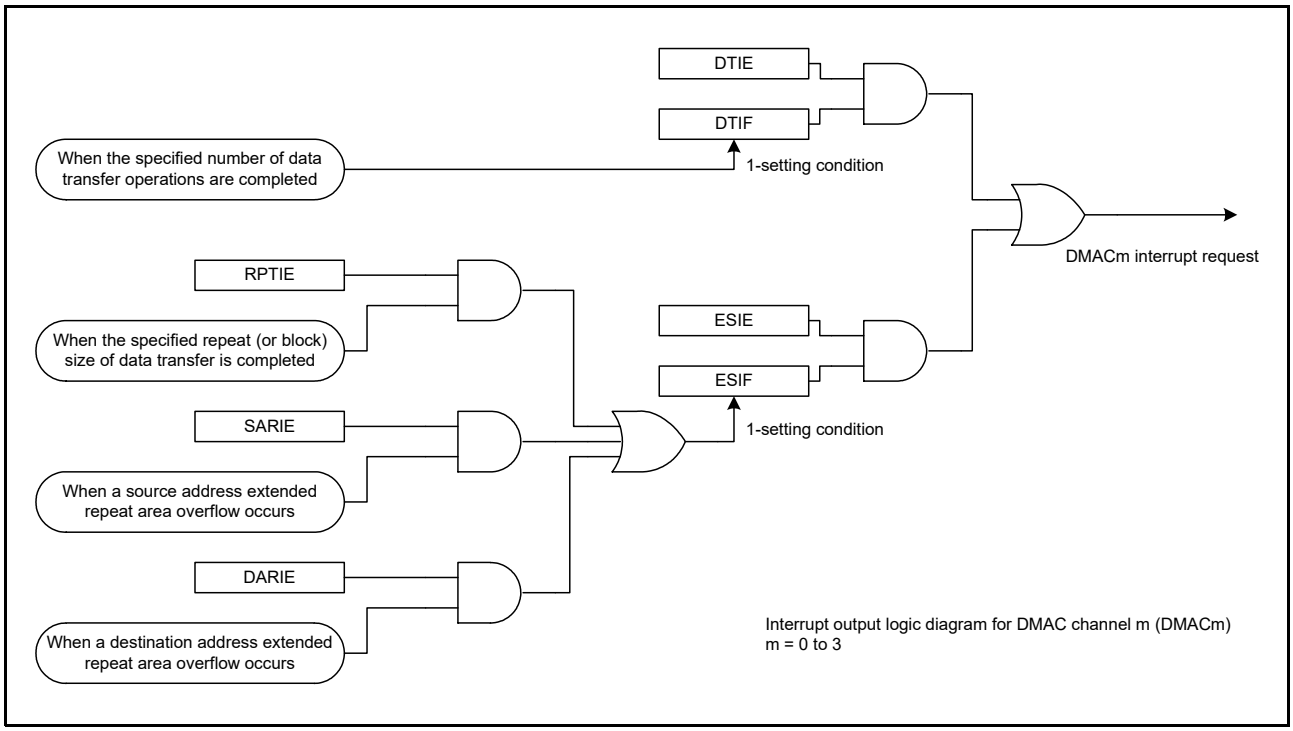


Figure 17.13 Schematic Logic Diagram of Interrupt Outputs (DMAC0 to DMAC3)

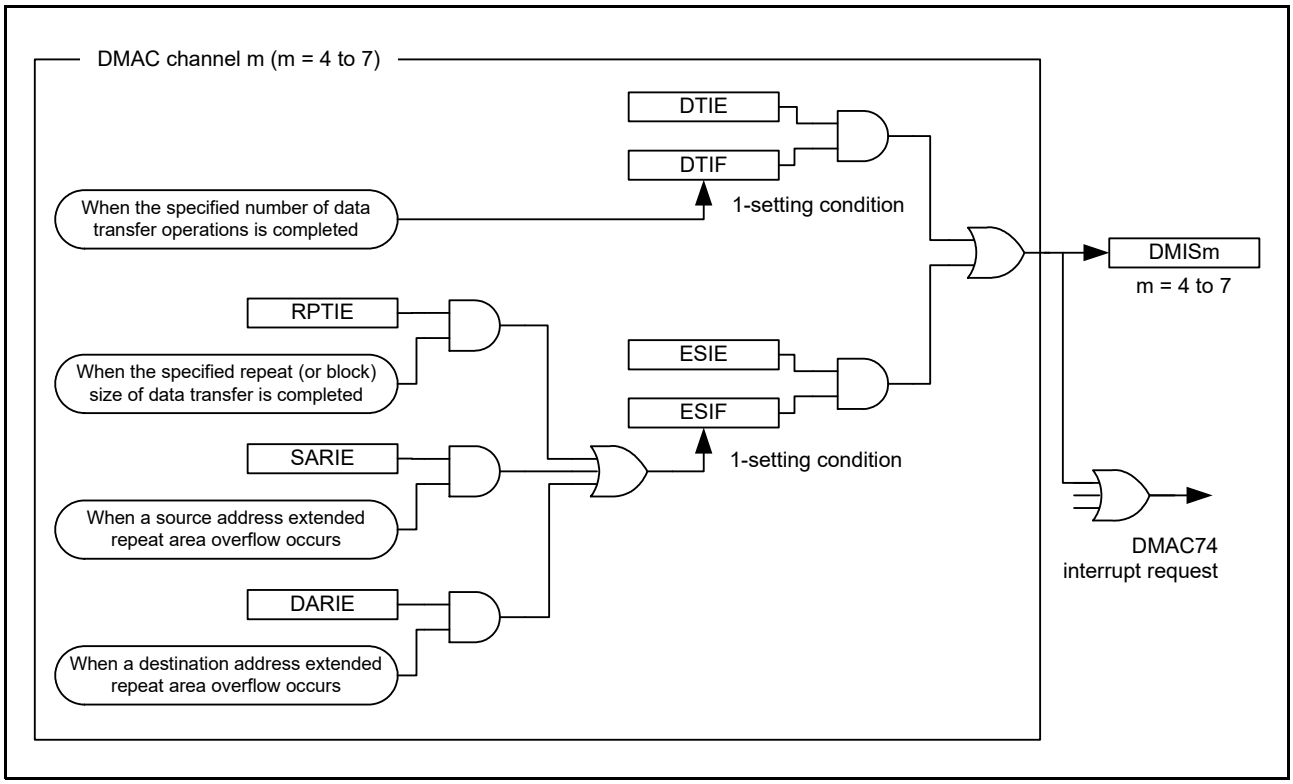


Figure 17.14 Schematic Logic Diagram of Interrupt Outputs (DMAC4 to DMAC7)

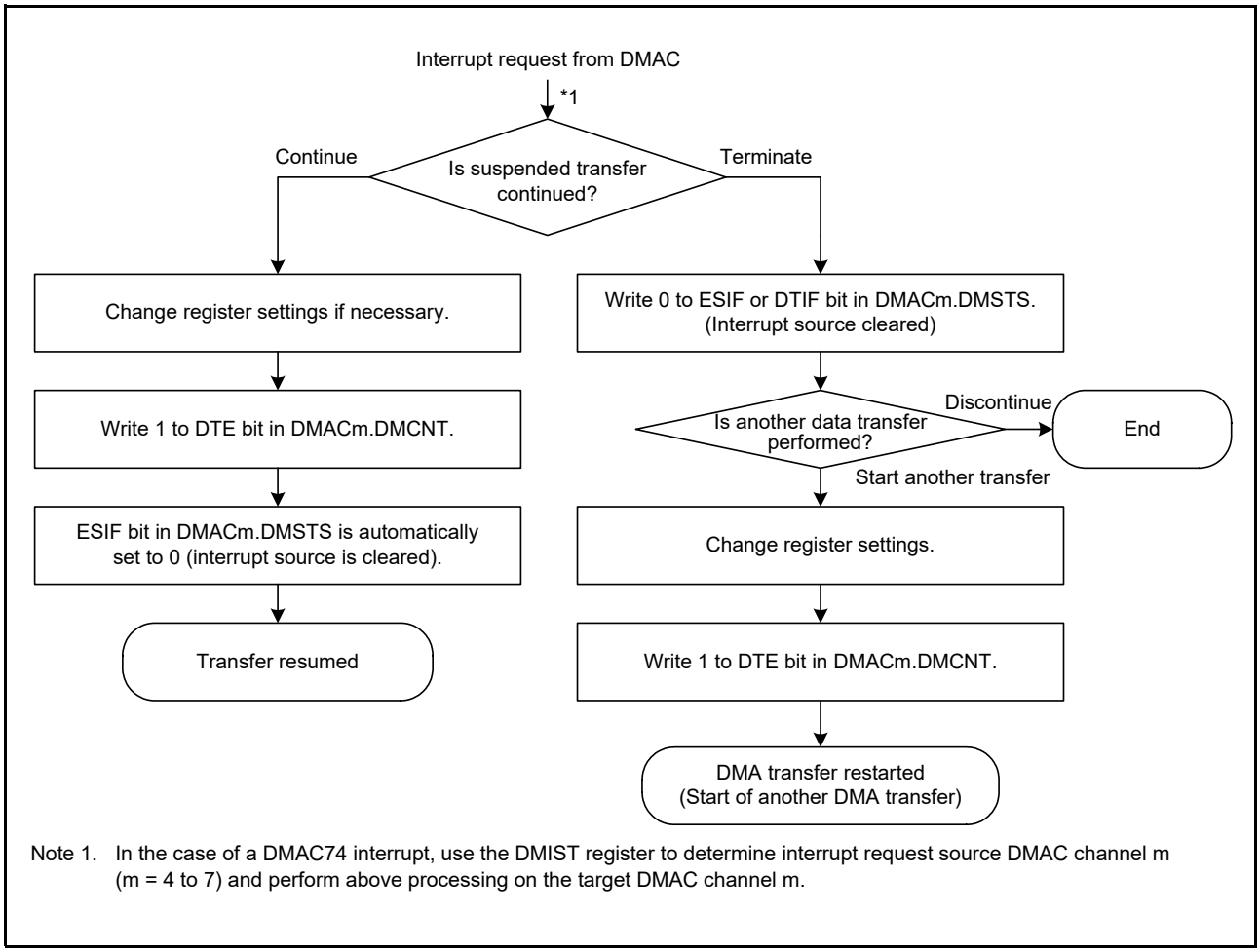
Specifically, the different procedures are used for canceling an interrupt to restart DMA transfer in the following two cases: (1) discontinuing or terminating DMA transfer and (2) continuing DMA transfer.

(1) When Discontinuing or Terminating DMA Transfer

Write 0 to the DTIF bit in DMSTS of DMACm to clear a transfer end interrupt, and to the ESIF bit in DMSTS of DMACm to clear a repeat size interrupt and an extended repeat area overflow interrupt. The DMACm remains in the stop state. When starting another DMA transfer after that, set the appropriate registers, and set the DTE bit in DMCNT of DMACm to 1 (DMA transfer enabled).

(2) When Continuing DMA Transfer

Write 1 to the DTE bit in DMCNT of DMACm. The ESIF bit in DMSTS of DMACm is automatically set to 0 (interrupt source cleared), and DMA transfer is resumed.



Note 1. In the case of a DMAC74 interrupt, use the DMIST register to determine interrupt request source DMAC channel m (m = 4 to 7) and perform above processing on the target DMAC channel m.

Figure 17.15 DMAC Interrupt Handling Routine to Resume/Terminate DMA Transfer

17.6 Event Link

Each DMAC channel outputs an event link request signal each time the channel completes data transfer (or block transfer in block transfer mode). However, when the transfer destination is the internal peripheral bus, an event link request signal is generated when the writing to the write buffer is accepted.

17.7 Low-Power Consumption Function

Before transition to the module-stop state, all-module clock stop mode, or software standby mode, set the DMST bit in DMAST to 0 (the DMAC suspended), and then perform the following.

(1) Module-Stop Function

Writing 1 to the MSTPA28 bit (transition to the module-stop state) in MSTPCRA enables the module-stop function of the DMAC. If DMA transfer is in progress at the time a 1 is written to the MSTPA28 bit, the transition to the module-stop state proceeds after DMA transfer has ended. While the MSTPA28 bit is 1, accessing the DMAC registers are prohibited.
Writing 0 to the MSTPA28 bit releases the DMAC from the module-stop state.

(2) All-Module Clock Stop Mode

Make settings in accord with the procedure under section 11.5.2.1, Transition to All-Module Clock Stop Mode, in section 11, Low Power Consumption.
If DMA transfer operations are in progress at the time the WAIT instruction is executed, the transition to all-module clock stop mode follows the completion of DMA transfer.
The DMAC is released from the module-stop state by writing 0 to the MSTPCRA.MSTPA28 bit following recovery from all-module clock stop mode.

(3) Software Standby Mode

Make settings in accord with the procedure under section 11.5.3.1, Transition to Software Standby Mode, in section 11, Low Power Consumption.
If DMA transfer operations are in progress at the time the WAIT instruction is executed, the transition to software standby follows the completion of DMA transfer.

(4) Note on Low-Power Consumption Function

For the WAIT instruction and the register setting procedure, see section 11.6.5, Timing of WAIT Instruction in section 11, Low Power Consumption.
To perform DMA transfer after returning from low-power consumption mode, set the DMST bit in DMAST to 1 again.
To use a request that is generated in all-module clock-stop mode and software standby mode as an interrupt request to the CPU but not as a DMAC startup request, specify the CPU as the interrupt request destination in accordance with the description in section 14, Interrupt Controller (ICUG), and then execute the WAIT instruction.

17.8 Usage Notes

17.8.1 DMA Transfer to Peripheral Modules

In DMA transfer to a peripheral module, the ACT bit in DMSTS of DMACm may be set to 0 (DMAC transfer suspended) during the period from the beginning of the final data write to the end of the peripheral bus access.

17.8.2 Access to the Registers during DMA Transfer

Do not write to the DMSAR, DMDAR, DMCRA, DMCRB, DMTMD, DMINT, DMAMD, DMOFR, and DMCSL registers of DMACm while the ACT bit in DMSTS of the same channel is set to 1 (DMAC active state) or the DTE bit in DMCNT of the same channel is set to 1 (DMA transfer enabled).

17.8.3 DMA Transfer to Reserved Areas

DMA transfer to the reserved areas is prohibited. If such an access is made, transfer results are not guaranteed. For details on the reserved areas, see section 4, Address Space.

17.8.4 Interrupt Request by the DMA Request Source Flag Control Register (DMCSL) at the End of Each Transfer

While the DMACm.DMCSL.DISEL bit is 1, an interrupt request is issued to the CPU at the end of each transfer that has been triggered by one DMA request. Unlike the transfer end interrupt that the DMAC outputs or the escape end interrupt, the interrupt of this type is issued to the CPU at the end of DMA transfer without clearing the interrupt status flag of the DMA request source to 0 by changing the interrupt request destination to the CPU. In this case, since the interrupt status flag is not set to 0 at the end of DMAC transfer, it should be set to 0 by the CPU interrupt routine.
The interrupt flag is cleared when the CPU interrupt is accepted.
For the change of the settings on the interrupt flag or the interrupt request destination, see section 14, Interrupt Controller (ICUG). For the DMACm.DMCSL.DISEL bit setting, see section 17.2.12, DMA Request Source Flag Control Register (DMCSL).

17.8.5 Setting of DMAC Trigger Select Register of the Interrupt Controller (ICU.DMRSRm)

The DMAC trigger select register (ICU.DMRSRm) should be set while the DMA transfer enable bit (DMACm.DMCNT.DTE) is 0 (DMA transfer is disabled). Moreover, the DTC transfer request enable register (ICU.DTCERm) that corresponds to the same vector number that has been set by the ICU.DMRSRm register should not be set to 1. For details on the ICU.DTCERn and ICU.DMRSRm, see section 14, Interrupt Controller (ICUG).

17.8.6 Suspending or Restarting DMA Transfer

To suspend a DMA transfer request, write 0 to the interrupt enable bit for the request source (ICU.IERn.IENj bit). To restart the DMA transfer, write 1 to the ICU.IERn.IENj bit with the setting shown in section 17.3.7, Activating the DMAC.

18. Data Transfer Controller (DTCb)

This MCU incorporates a data transfer controller (DTC).
The DTC is triggered by an interrupt request to perform data transfers.
In addition to the conventional methods of DTC transfer (normal, repeat, block, and chain), DTCb supports sequential transfer, in which it handles a series of transfers made up of a combination of the other methods. In sequential transfer, the data that is initially transferred selects one from possible 256 sequences for execution. The DTCb can divide one sequence into several transfers depending on how the parts of the sequence are combined.

18.1 Overview

Table 18.1 lists the specifications of the DTC, and Figure 18.1 shows a block diagram of the DTC.

Table 18.1 DTC Specifications

Item	Description
Number of transfer channels	• The same number as all interrupt sources that can start the DTC transfer.
Transfer modes	• Normal transfer mode A single transfer request leads to a single data transfer. • Repeat transfer mode A single transfer request leads to a single data transfer. The transfer address is returned to the transfer start address after the number of data transfers corresponding to “repeat size”. The maximum number of repeat transfers is 256, and the maximum data transfer size is 256 × 32 bits, 1024 bytes. • Block transfer mode A single transfer request leads to the transfer of a single block. The maximum block size is 256 × 32 bits = 1024 bytes.
Chain transfer	• Multiple types of data transfers can sequentially be executed in response to a single request. • Either “performed only when the transfer counter becomes 0” or “every time” can be selected.
Sequence transfer	A series of complicated transfers can be registered as a sequence. Any sequence can be selected by the transfer data and executed. • Only one trigger source can be set at a time. • Up to 256 sequences for a single trigger source • The data that is initially transferred in response to a transfer request determines a sequence • The whole sequence can be executed on a single request, or be suspended in the middle of the sequence and resumed on the next transfer request (division of sequence).
Transfer space	• In short-address mode: 16 Mbytes (Areas from 0000 0000h to 007F FFFFh and FF80 0000h to FFFF FFFFh except reserved areas) • In full-address mode: 4 Gbytes (Area from 0000 0000h to FFFF FFFFh except reserved areas)
Data transfer units	• Single data: 1 byte (8 bits), 1 word (16 bits), 1 longword (32 bits) • Single block size: 1 to 256 data
CPU interrupt source	• An interrupt request can be generated to the CPU on a request source for a data transfer. • An interrupt request can be generated to the CPU after a single data transfer. • An interrupt request can be generated to the CPU after data transfer of specified volume.
Event link function	An event link request is generated after one data transfer (for block, after one block transfer).
Read skip	Reading of the transfer information can be skipped when the same transfer is repeated.
Write-back skip	Write-back of the transferred data that is not updated can be skipped when the address of the transfer source or destination is fixed.
Write-back disable	Allows disabling the write-back of transfer information.
Displacement addition	The displacement value can be added to the transfer source address (for each transfer information)
Low power consumption function	Module stop state can be set.

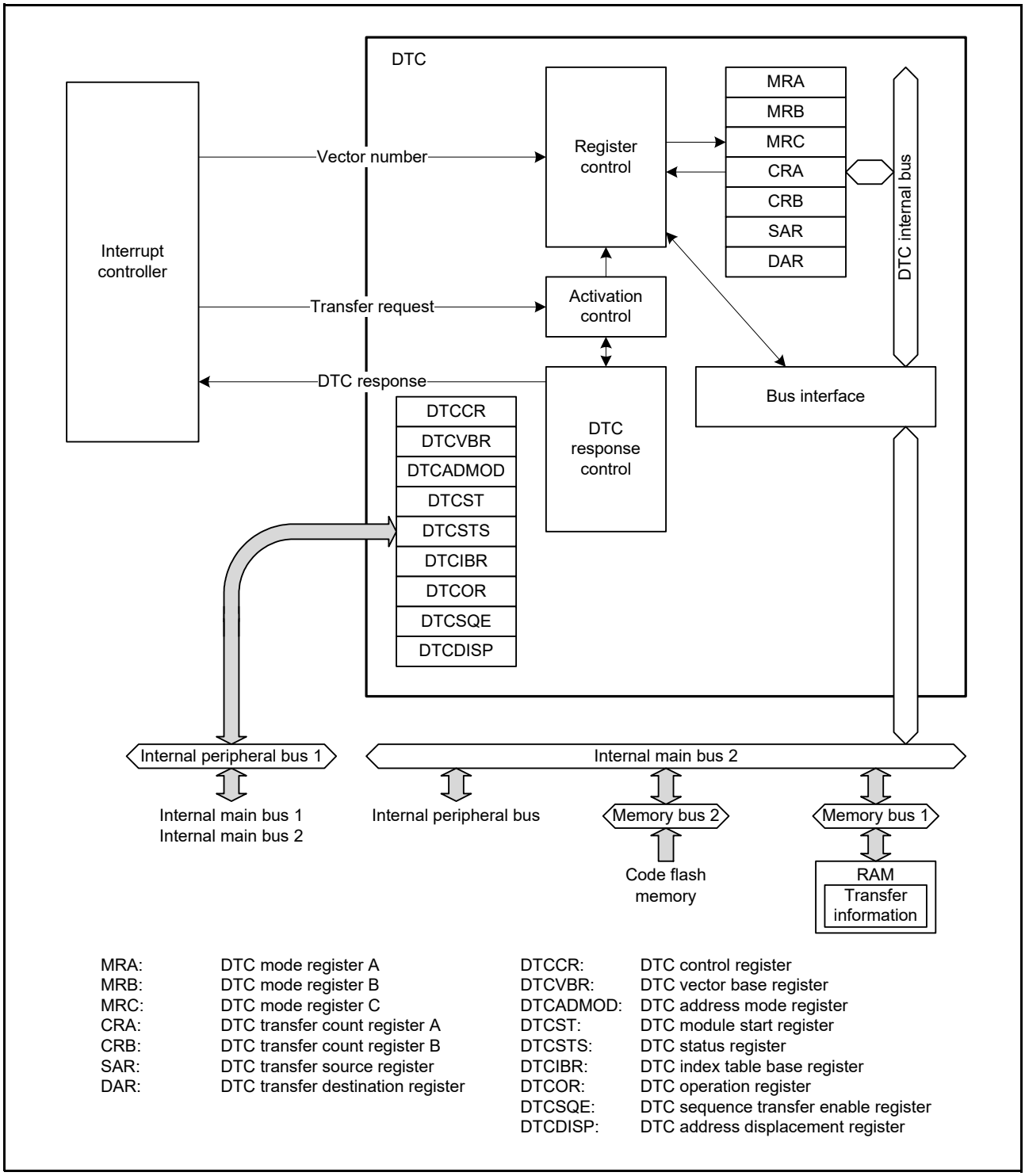


Figure 18.1 DTC Block Diagram

18.2 Register Descriptions

Registers MRA, MRB, MRC, SAR, DAR, CRA, and CRB are DTC internal registers, which cannot be directly accessed from the CPU. Values to be set in these DTC internal registers are placed in the RAM area as transfer information. When accepting a transfer request, the DTC reads the transfer information from the RAM area and sets it in the internal registers. After the data transfer ends, the values of the updated internal register are written back to the RAM area as transfer information.

18.2.1 DTC Mode Register A (MRA)

Address(es): (inaccessible directly from the CPU)

	b7	b6	b5	b4	b3	b2	b1	b0
	MD[1:0]		SZ[1:0]		SM[1:0]		—	WBDIS
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	WBDIS	Write-back Disable	0: Writes back the transfer information on completion of the data transfer 1: Does not write back the transfer information on completion of the data transfer	—
b1	—	Reserved	Set this bit to 0.	—
b3, b2	SM[1:0]	Transfer Source Address Addressing Mode	b3 b2 0 0: The address in the SAR register is fixed. (write-back to SAR is skipped.) 0 1: The address in the SAR register is fixed. (write-back to SAR is skipped.) 1 0: The SAR value is incremented after a data transfer. (+1 when the SZ[1:0] bits are 00b, +2 when 01b, +4 when 10b) 1 1: The SAR value is decremented after a data transfer. (–1 when the SZ[1:0] bits are 00b, –2 when 01b, –4 when 10b)	—
b5, b4	SZ[1:0]	DTC Data Transfer Size	b5 b4 0 0: Byte (8-bit) transfer 0 1: Word (16-bit) transfer 1 0: Longword (32-bit) transfer 1 1: Setting prohibited	—
b7, b6	MD[1:0]	DTC Transfer Mode Select	b7 b6 0 0: Normal transfer mode 0 1: Repeat transfer mode 1 0: Block transfer mode 1 1: Setting prohibited	—

MRA register is used to select the DTC operating mode and cannot be accessed directly from the CPU.

WBDIS Bit (Write-back Disable)

The WBDIS bit selects whether to write back the transfer information.

When the bit is 0, updated transfer information is written back.

When the bit is 1, updated transfer information is not written back even with the setting of that address is incremented after a transfer, and the same data transfer is executed every time for each transfer request. The transfer information can be stored in ROM because the transfer information is not written back.

While the WBDIS bit is 1, operation for each transfer mode is as follows:

(1) Normal transfer and repeat transfer modes
1-byte, 1-word, or 1-longword of data is transferred on a single transfer request. The transfer address and transfer count are not updated so that the same transfer is repeated on each transfer request. When the transfer count is 1, the ICU.DTCERn.DTCE bit is not set to 0, and data transfer continues in response to the next transfer request.

(2) Block transfer mode
1-block of data is transferred on a single transfer request. The transfer address and transfer count are not updated so that the same block transfer is repeated on each transfer request. When the block transfer count is 1, the ICU.DTCERn.DTCE bit is not set to 0, and data transfer continues in response to the next transfer request.

When setting the DISPE bit to 1, set the MRA.WBDIS bit to 1 (does not write back the transfer information). If the value of the WBDIS bit in any transfer information is 1, set the DTCCR.RRS bit to 0 (so that reading of the transfer information is not skipped).

18.2.2 DTC Mode Register B (MRB)

Address(es): (inaccessible directly from the CPU)

	b7	b6	b5	b4	b3	b2	b1	b0
	CHNE	CHNS	DISEL	DTS	DM[1:0]		INDX	SQEND
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	SQEND	Sequence Transfer End	0: Continue the sequence transfer 1: End the sequence transfer	—
b1	INDX	Index Table Reference	0: Does not refer to the index table 1: Refers the index table based on the transferred data*1	—
b3, b2	DM[1:0]	Transfer Destination Address Addressing Mode	b3 b2 0 0: The address in the DAR register is fixed. (Write-back to DAR is skipped.) 0 1: The address in the DAR register is fixed. (Write-back to DAR is skipped.) 1 0: The DAR value is incremented after data transfer. (+1 when the MRA.SZ[1:0] bits are 00b, +2 when 01b, +4 when 10b) 1 1: The DAR value is decremented after data transfer. (–1 when the MRA.SZ[1:0] bits are 00b, –2 when 01b, –4 when 10b)	—
b4	DTS	DTC Transfer Mode Select	0: Transfer destination side is repeat area or block area. 1: Transfer source side is repeat area or block area.	—
b5	DISEL	DTC Interrupt Select	0: An interrupt request to the CPU is generated on completion of the specified number of data transfers. 1: An interrupt request to the CPU is generated for each data transfer.	—
b6	CHNS	DTC Chain Transfer Select	0: Chain transfer is performed on completion of each transfer. 1: Chain transfer is performed only when the transfer counter is changed from 1 to 0 or 1 to CRAH.	—
b7	CHNE	DTC Chain Transfer Enable	0: Chain transfer is disabled. 1: Chain transfer is enabled.	—

Note 1. Set the MRA.MD[1:0] bits to 00b (normal transfer mode) when setting the INDX bit to 1.

MRB register is used to select the DTC operating mode and cannot be accessed directly from the CPU.

SQEND Bit (Sequence Transfer End)

The SQEND bit selects whether to continue or end sequence transfer. Refer to Table 18.2 for details.
This bit can only be set to 1 for transfer information referred to by the DTC index table. Set this bit to 0 for transfer information referred to by the DTC vector table.

INDX Bit (Index Table Reference)

When the value of the INDX bit in transfer information that is read is 1, a sequence transfer proceeds. Refer to Table 18.2 for details.
Set this bit to 0 for transfer information which is not associated with sequence transfer or is not intended to start sequence transfer. Do not allow transfer requests to be generated by the sources different from that specified in the DTCSQE register but having the INDX bit set to 1.

Table 18.2 Values of Bits CHNE, SQEND, and INDX in the Sequence Transfer and DTC Operation

CHNE Bit	SQEND Bit	INDX Bit	Operation	Usage
0	0	1	Start sequence transfer	Use this setting for the transfer information that is first read in response to a transfer request from the source specified in the DTCSQE register.
1	0	0	Continue sequence transfer	Use this setting for the first or intermediate transfer information in a sequence.
0	0	0	Suspend sequence transfer	Use this setting for the first or intermediate transfer information in a sequence.
0	1	0	End sequence transfer	Use this setting with the last transfer information in a sequence.
0	1	1	End current sequence transfer and start new sequence transfer	Use this setting with the last transfer information in a sequence.

Note: Do not set the values other than listed above.

DTS Bit (DTC Transfer Mode Select)

The DTS bit specifies the side (transfer source or destination) to be a repeat area or block area in repeat transfer mode or block transfer mode.

CHNS Bit (DTC Chain Transfer Select)

The CHNS bit selects the chain transfer condition.

When the CHNE bit is 0, setting of the CHNS bit is ignored. For details on the conditions to select the chain transfer, refer to Table 18.4, Chain Transfer Conditions.

When the next transfer is chain transfer, completion of the specified number of transfers is not determined, the interrupt status flag for the request source is not cleared, and an interrupt request to the CPU is not generated.

CHNE Bit (DTC Chain Transfer Enable)

The CHNE bit enables or disables chain transfer.

The chain transfer condition is selected by the CHNS bit.

For details of chain transfer, refer to section 18.4.6, Chain Transfer.

Refer to Table 18.2 for the setting value to be used in the sequence transfer.

18.2.3 DTC Mode Register C (MRC)

Address(es): (inaccessible directly from the CPU)

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DISPE
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	DISPE	Displacement Addition	0: The displacement value is not added to the transfer source address. 1: The displacement value is added to the transfer source address.	—
b7 to b1	—	Reserved	Set these bits to 0.	—

The MRC register is used to select DTC operating mode and cannot be accessed directly from the CPU.
This register can only be used in full-address mode, but not in short-address mode. Therefore, set the DTCADMOD.SHORT bit to 0 (full-address mode) when using the displacement addition function.

DISPE Bit (Displacement Addition)

This bit specifies whether to use the SAR + DTCDISP value as the transfer source address.
When setting the DISPE bit to 1, set the MRA.WBDIS bit to 1 (does not write back the transfer information) and set the DTCCR.RRS bit to 0 (transfer information read is not skipped).

18.2.4 DTC Transfer Source Register (SAR)

Address(es): (inaccessible directly from the CPU)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

SAR register is used to set the transfer source start address.
In full-address mode, 32 bits are valid.
In short-address mode, lower 24 bits are valid and upper 8 bits (b31 to b24) are ignored. The address of this register is extended by the value specified by b23.
SAR register cannot be accessed directly from the CPU.

18.2.5 DTC Transfer Destination Register (DAR)

Address(es): (inaccessible directly from the CPU)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

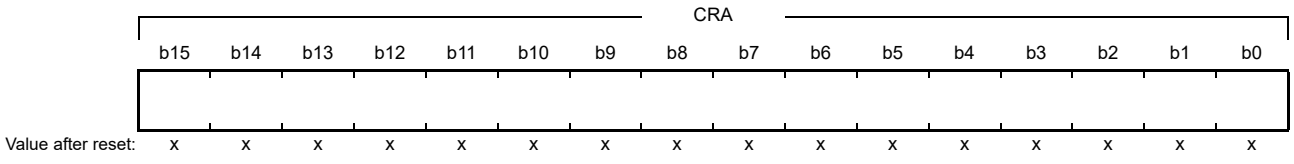
x: Undefined

DAR register is used to set the transfer destination start address.
In full-address mode, 32 bits are valid.
In short-address mode, lower 24 bits are valid and upper 8 bits (b31 to b24) are ignored. The address of this register is extended by the value specified by b23.
DAR register cannot be accessed directly from the CPU.

18.2.6 DTC Transfer Count Register A (CRA)

- Normal transfer mode

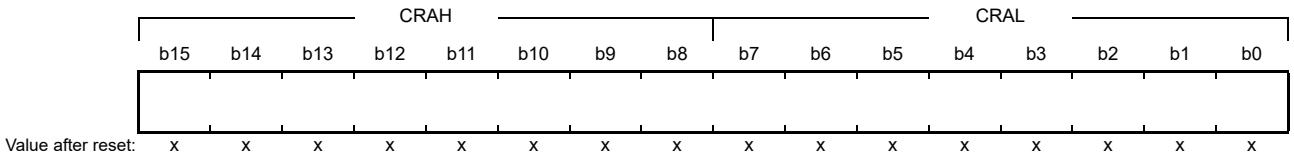
Address(es): (inaccessible directly from the CPU)



x: Undefined

- Repeat transfer mode/block transfer mode

Address(es): (inaccessible directly from the CPU)



x: Undefined

Symbol	Register Name	Description	R/W
CRAL	Transfer Counter A Lower Register	Set transfer count. This register functions as a transfer counter during data transfer.	—
CRAH	Transfer Counter A Upper Register	Set transfer count. This register functions as a reload register during data transfer.	—

Note: The function depends on transfer mode.
Note: Set CRAH and CRAL to the same value in repeat transfer mode and block transfer mode.

This register is for counting the number of transfers and cannot be accessed directly from the CPU.

(1) Normal transfer mode (MRA.MD[1:0] bits = 00b)

CRA register functions as a 16-bit transfer counter in normal transfer mode.
The transfer count is 1, 65535, and 65536 when the set value is 0001h, FFFFh, and 0000h, respectively.
The CRA value is decremented (–1) at each data transfer.

(2) Repeat transfer mode (MRA.MD[1:0] bits = 01b)

The CRAH register retains the transfer count and the CRAL register functions as an 8-bit transfer counter.
The transfer count is 1, 255, and 256 when the set value is 01h, FFh, and 00h, respectively.
The CRAL value is decremented (–1) at each data transfer. When it reaches 00h, the CRAH value is reloaded to the CRAL register.

(3) Block transfer mode (MRA.MD[1:0] bits = 10b)

The CRAH register retains the block size and the CRAL register functions as an 8-bit block size counter.
The transfer count is 1, 255, and 256 when the set value is 01h, FFh, and 00h, respectively.
The CRAL value is decremented (–1) at each data transfer. When it reaches 00h, the CRAH value is reloaded to the CRAL register.

18.2.7 DTC Transfer Count Register B (CRB)

Address(es): (inaccessible directly from the CPU)

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

CRB register is used to set the block transfer count for block transfer mode and cannot be accessed directly from the CPU.
The transfer count is 1, 65535, and 65536 when the set value is 0001h, FFFFh, and 0000h, respectively.
The CRB value is decremented (–1) when the final data of a single block size is transferred.
When normal transfer mode or repeat transfer mode is selected, this register is not used and the set value is ignored.

18.2.8 DTC Control Register (DTCCR)

Address(es): DTC.DTCCR 0008 2400h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	RRS	—	—	—	—
Value after reset:	0	0	0	0	1	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b4	RRS	DTC Transfer Information Read Skip Enable*1	0: Transfer information read is not skipped. 1: Transfer information read is skipped when vector numbers match.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Set this bit to 0 when using the sequence transfer.

DTCCR register is used to control the DTC operation.

RRS Bit (DTC Transfer Information Read Skip Enable)

The DTC vector number is compared with the vector number in the previous data transfer.
When these vector numbers match and the RRS bit is set to 1, DTC data transfer is performed without reading the transferred information. However, when the previous transfer was chain transfer, the transferred information is read regardless of the value of the RRS bit.
Furthermore, when the transfer counter (CRA register) became 0 during the previous normal transfer and when the transfer counter (CRB register) became 0 during the previous block transfer, the transferred information is read regardless of the RRS bit value.
If the value of the MRA.WBDIS bit in any transfer information is 1, set the RRS bit to 0. Note that the MRA.WBDIS bit should be set to 1 when the MRC.DISPE bit is set to 1.
Like chain transfer, sequence transfer handles sequences of multiple types of data transfer. When sequence transfer is to be used, set the RRS bit to 0 so that the previous data transfer will not be repeated.

18.2.9 DTC Vector Base Register (DTCVBR)

Address(es): DTC.DTCVBR 0008 2404h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The DTCVBR register is used to set the base address for calculating the address to which the DTC vector is allocated. Values for the upper 4 bits (b31 to b28) cannot be written but reflect the value written to b27. The lower 10 bits are reserved and the values are fixed to 0. Write 0 to the lower 10 bits if necessary.
It can be set in the range of 0000 0000h to 07FF FC00h and F800 0000h to FFFF FC00h in 1-Kbyte units.

18.2.10 DTC Address Mode Register (DTCADMOD)

Address(es): DTC.DTCADMOD 0008 2408h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SHORT
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SHORT	Short-Address Mode Set*1	0: Full-address mode 1: Short-address mode	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Set this bit to 0 (full-address mode) when using the sequence transfer.

DTCADMOD register is used to specify the area accessible by the DTC.

SHORT Bit (Short-Address Mode Set)

This bit is used to select address mode of registers SAR and DAR.
Full-address mode allows the DTC to access to a 4-Gbyte space (0000 0000h to FFFF FFFFh).
Short-address mode allows the DTC to access to a 16-Mbyte space (0000 0000h to 007F FFFFh and FF80 0000h to FFFF FFFFh).

18.2.11 DTC Module Start Register (DTCST)

Address(es): DTC.DTCST 0008 240Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DTCST
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DTCST	DTC Module Start	0: DTC module stop 1: DTC module start	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

DTCST Bit (DTC Module Start)

Set the DTCST bit to 1 to enable the DTC to accept transfer requests. When this bit is set to 0, transfer requests are no longer accepted.

If this bit is set to 0 during data transfer, the accepted transfer request is active until the processing is completed.

Set the DTCST bit to 0 before making a transition to the module stop state, all-module clock stop mode, or software standby mode.

Set the DTCST bit to 1 to resume the data transfer after returning from the module stop state, all-module clock stop mode, or software standby mode.

For details on transitions to the module stop state, all-module clock stop mode, and software standby mode, refer to section 18.9, Low Power Consumption Function, and section 11, Low Power Consumption.

18.2.12 DTC Status Register (DTCSTS)

Address(es): DTC.DTCSTS 0008 240Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ACT	—	—	—	—	—	—	—	VECN[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	VECN[7:0]	DTC Active Vector Number Monitoring Flag	These bits indicate the vector number for the request source when data transfer is in progress. The value is only valid if data transfer is in progress (the value of the ACT flag is 1).	R
b14 to b8	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b15	ACT	DTC Active Flag	0: Data transfer is not in progress. 1: Data transfer is in progress.	R

VECN[7:0] Flags (DTC Active Vector Number Monitoring Flag)

While data transfer is in progress, these bits indicate the vector number corresponding to the request source for the transfer.

When the DTCSTS register is read, the value of the VECN[7:0] flags is valid if the value of the ACT flag was 1 (data transfer is in progress) and invalid if the value of the ACT flag was 0 (data transfer is not in progress).

For the correspondence between the DTC request sources and the vector addresses, refer to Table 14.4, Interrupt Vector Table in section 14, Interrupt Controller (ICUG).

ACT Flag (DTC Active Flag)

This flag indicates the state of data transfer operation.

[Setting condition]

- When the data transfer is started by a transfer request.
- When the sequence transfer is resumed.

[Clearing condition]

- When the data transfer is completed in response to a transfer request.
- When the sequence transfer is suspended.

18.2.13 DTC Index Table Base Register (DTCIBR)

Address(es): DTC.DTCIBR 0008 2410h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The DTCIBR register is used to set the base address for calculating the address to which the DTC index is allocated.
Values for the upper 4 bits (b31 to b28) cannot be written but reflect the value written to b27.
The lower 10 bits (b9 to b0) are reserved bits and fixed to 0. When writing this register, set these bits to 0.
It can be set in the range of 0000 0000h to 07FF FC00h and F800 0000h to FFFF FC00h in 1-Kbyte units.

18.2.14 DTC Operation Register (DTCOR)

Address(es): DTC.DTCOR 0008 2414h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SQTFRL
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SQTFRL	Sequence Transfer Terminate	Writing 1 to this bit terminates the sequence transfer in progress. This bit is read as 0.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R

The DTCOR register sets the operation of the DTC module.

SQTFRL Bit (Sequence Transfer Terminate)

Setting the SQTFRL bit to 1 terminates the sequence transfer in progress.
When the DTCSQE.ESPSEL bit is 1 (Sequence transfer is enabled), follow the procedure shown in Figure 18.2 to terminate the sequence transfer.
Writing 1 to the bit, while no sequence transfer is performed, have no effect.

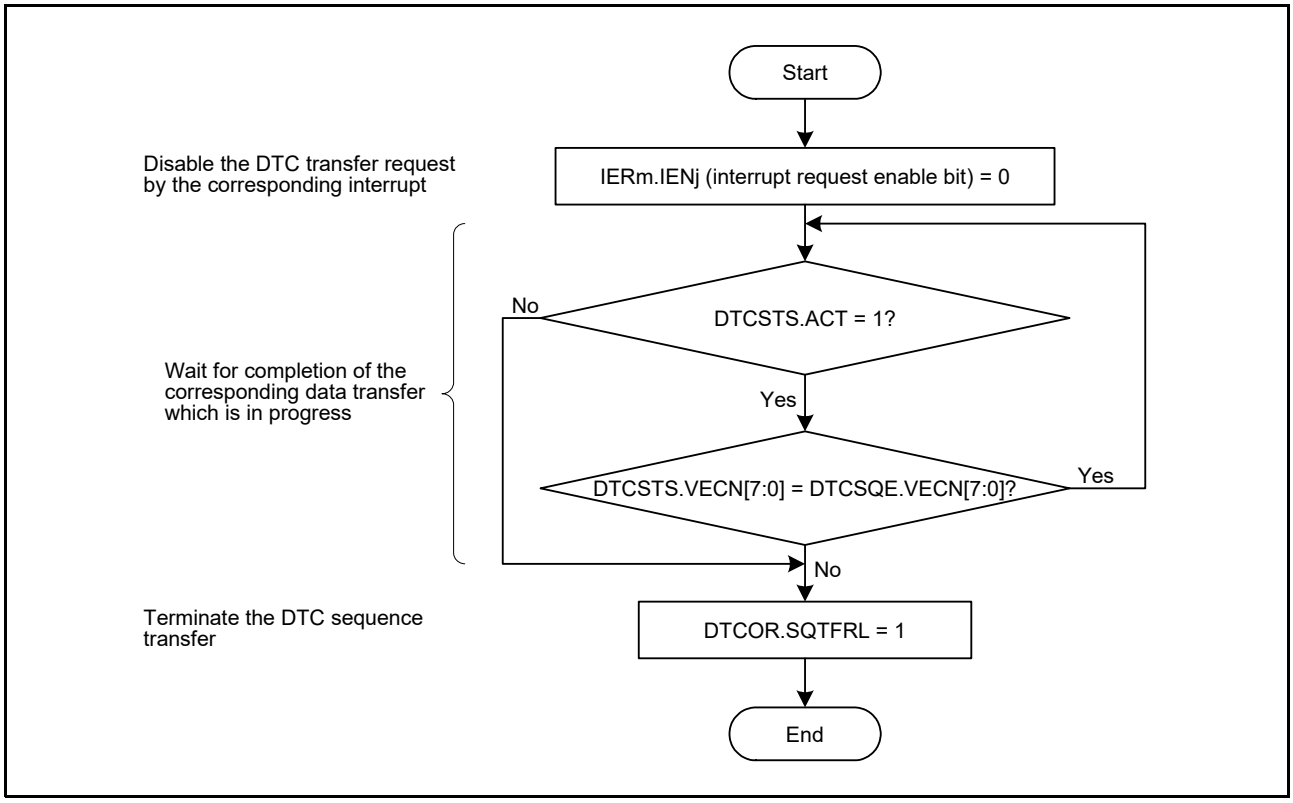


Figure 18.2 Procedure to Terminate Sequence Transfer

18.2.15 DTC Sequence Transfer Enable Register (DTCSQE)

Address(es): DTC.DTCSQE 0008 2416h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ESPSEL	—	—	—	—	—	—	—	VECN[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	VECN[7:0]	Sequence Transfer Vector Number Setting	Specify the vector number by which a sequence transfer is enabled. The value is only valid when the ESPSEL bit is 1.	R/W
b14 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R
b15	ESPSEL	Sequence Transfer Enable	0: Sequence transfer is disabled. 1: Sequence transfer is enabled.	R/W

The DTCSQE register is used to specify sequence transfer. Follow Figure 18.24 for details on the setting procedure.

VECN[7:0] Bit (Sequence Transfer Vector Number Setting)

This bit is used to specify for which vector number to perform sequence transfer. Sequence transfer can occur only for this trigger source.

Table 14.4, Interrupt Vector Table in section 14, Interrupt Controller (ICUG) shows the relationship between the trigger source and the vector number.

ESPSEL Bit (Sequence Transfer Enable)

The ESPSEL bit specifies whether sequence transfer is used.

Set the DTCADMOD.SHORT bit to 0 (full address mode), when setting the ESPSEL bit to 1.

18.2.16 DTC Address Displacement Register (DTCDISP)

Address(es): DTC.DTCDISP 0008 2418h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The DTCDISP register is used to specify the displacement value to add to the DTC transfer source address.

If MRC.DISPE bit is 1, the value SAR + DTCDISP is used as the transfer source address.

18.3 Request Sources

The DTC data transfer is triggered by an interrupt request. Setting the ICU.DTCERn.DTCE bit (n = interrupt vector number) to 1 selects the corresponding interrupt request as a request source for the DTC.
For the correspondence between the DTC request sources and the vector addresses, refer to Table 14.4, Interrupt Vector Table in section 14, Interrupt Controller (ICUG). For request by software, refer to section 14.2.5, Software Interrupt Generation Register (SWINTR) and section 14.2.6, Software Interrupt 2 Generation Register (SWINT2R) in section 14, Interrupt Controller (ICUG).
Once the DTC has accepted a transfer request, it does not accept another transfer request until transfer for that single request is completed, regardless of the priority of the requests.
When multiple transfer requests are generated during data transfer by the DMAC/DTC, the request with the highest priority on completion of the current transfer is accepted. When multiple transfer requests are generated while the DTCST.DTCST bit is 0 (DTC module stop), the request with the highest priority at the moment when the bit is subsequently set to 1 (DTC module start) is accepted.

The DTC performs the following operations at the start of a single data transfer (or the last of the consecutive transfers in the case of a chain transfer).

- On completion of a specified number of data transfer, the ICU.DTCERn.DTCE bit is set to 0 and an interrupt is requested to the CPU.
- If the MRB.DISEL bit is 1, an interrupt is requested to the CPU on completion of data transfer.
- For the other transfers, the interrupt status flag of the request source is set to 0 at the start of data transfer.

18.3.1 Allocating Transfer Information and DTC Vector Table

The DTC reads the start address of the transfer information corresponding to each request source from the vector table and reads the transfer information starting at that address.
The vector table should be located so that the lower 10 bits of the base address (start address) are 0. Use the DTC vector base register (DTCVBR) to set the base address of the DTC vector table.
Transfer information is allocated in the RAM area. Transfer information can be allocated in the ROM area when the MRA.WBDIS bit is set to 1. The start address of the transfer information n with vector number n should be allocated at DTCVBR + 4n.
Transfer information should be aligned on a 4-byte boundary. The size of a transfer information is 12 bytes in short-address mode or 16 bytes in full-address mode. Use the DTCADMOD.SHORT bit to select short-address mode (SHORT bit = 1) or full-address mode (SHORT bit = 0).
Figure 18.3 shows the relationship between the DTC vector table and transfer information.
Figure 18.4 shows the allocation of transfer information in the RAM area. The lower addresses vary according to the endian of the corresponding allocation area. For details, refer to section 18.10.2, Allocating Transfer Information.

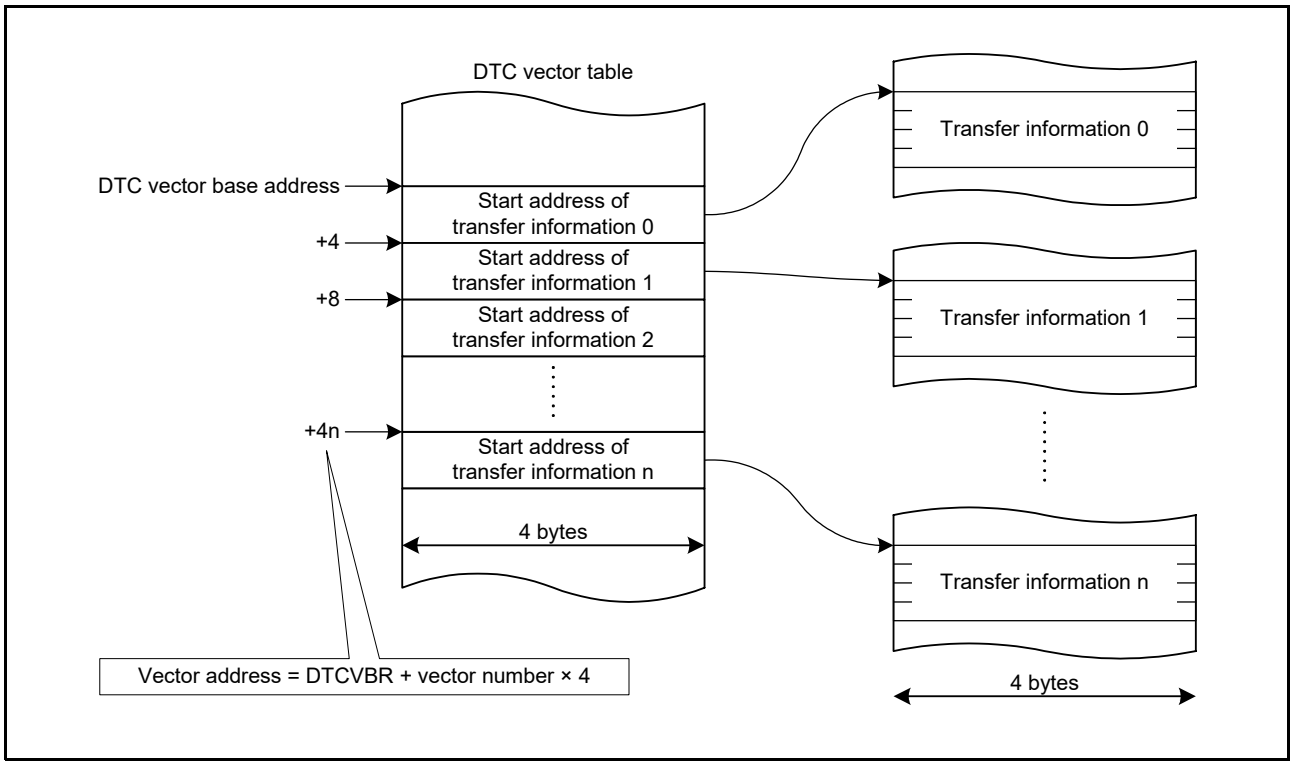


Figure 18.3 DTC Vector Table and Transfer Information

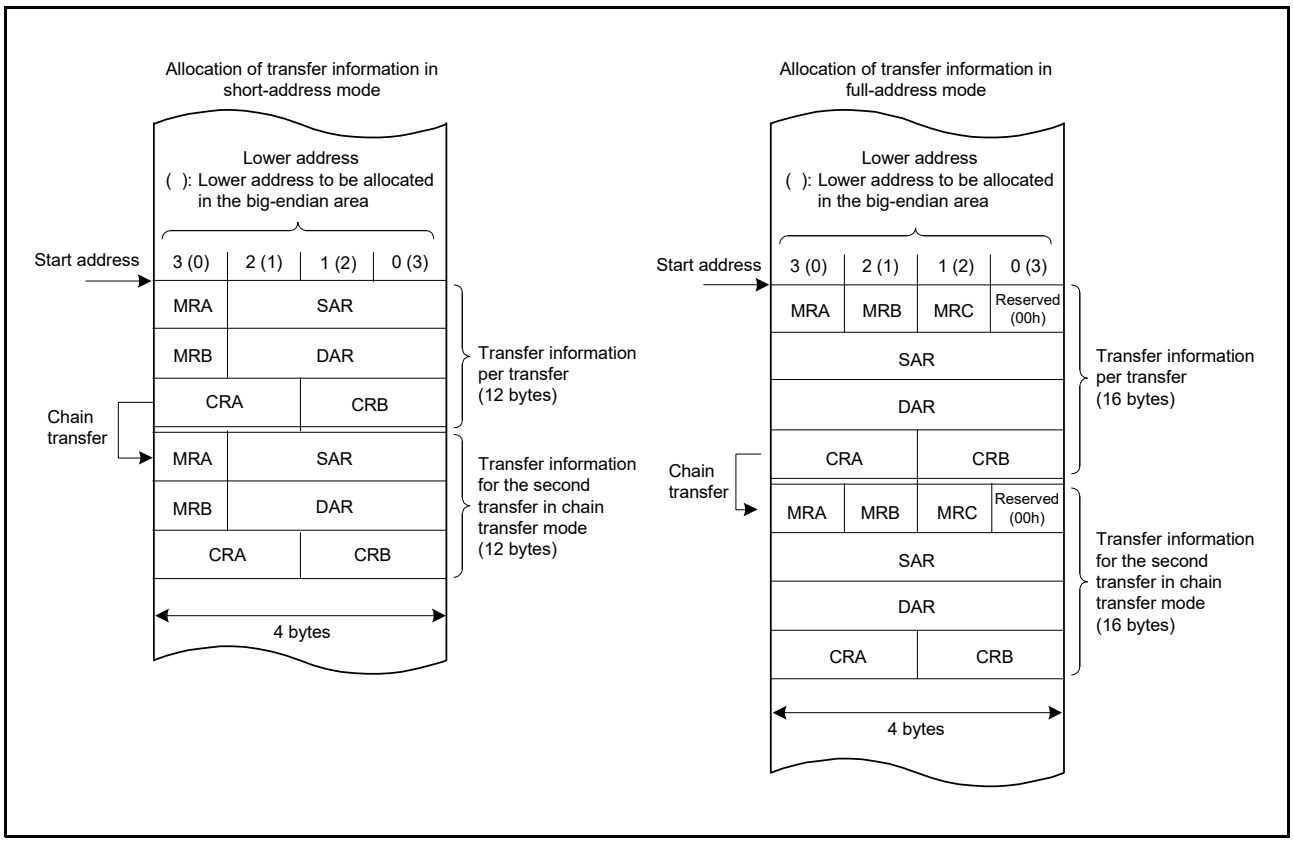


Figure 18.4 Allocation of Transfer Information in the RAM Area

18.4 Operation

The DTC transfers data in accordance with the transfer information. Storage of the transfer information in the RAM area is required before DTC operation.

When the DTC accepts a transfer request, it reads the DTC vector corresponding to the vector number. Next, the DTC reads transfer information from the address pointed by the DTC vector, transfers data, and then writes back the transfer information after the data transfer. Allocating transfer information in the RAM area allows data transfer of arbitrary number of channels.

There are three transfer modes: normal transfer mode, repeat transfer mode, and block transfer mode.

Set a transfer source address in the SAR register and a transfer destination address in the DAR register. The SAR and DAR registers are updated after the transfer according to the respective settings (increment, decrement, or fixed).

Table 18.3 lists transfer modes of the DTC.

Table 18.3 Transfer Modes of the DTC

Transfer Mode	Data Size Transferred on Single Transfer Request	Increment/Decrement of Memory Address	Settable Transfer Count
Normal transfer mode	1 byte/1 word/1 longword	Incremented/decremented by 1, 2, or 4 or address fixed	1 to 65536
Repeat transfer mode*1	1 byte/1 word/1 longword	Incremented/decremented by 1, 2, or 4 or address fixed	1 to 256*3
Block transfer mode*2	Block size specified in CRAH (1 to 256 bytes/1 to 256 words/1 to 256 longwords)	Incremented/decremented by 1, 2, or 4 or address fixed	1 to 65536

Note 1. Set transfer source or transfer destination in the repeat area.
Note 2. Set transfer source or transfer destination in the block area.
Note 3. After data transfer of the specified count, the initial state is restored and the operation is continued (repeated).

Setting the MRB.CHNE bit to 1 allows multiple transfers (chain transfer) on a single transfer request. The setting in combination with the MRB.CHNS bit enables a chain transfer when the specified number of data transfers is completed.

Figure 18.5 shows the operation flowchart of the DTC. Table 18.4 lists chain transfer conditions.

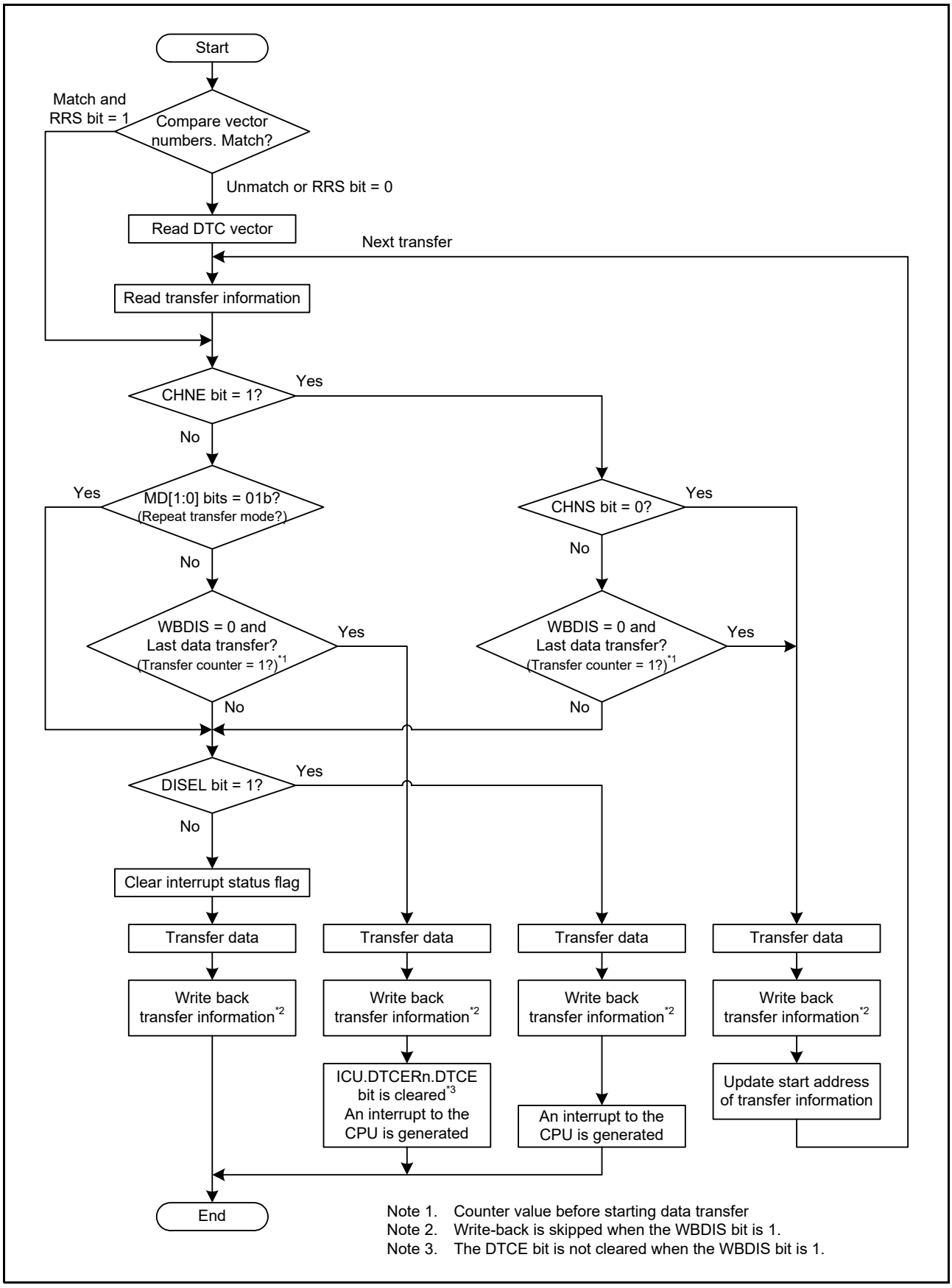


Figure 18.5 Operation Flowchart of the DTC

Table 18.4 Chain Transfer Conditions

First Transfer				Second Transfer*3				
CHNE Bit	CHNS Bit	DISEL Bit	Transfer Counter*1,*2	CHNE Bit	CHNS Bit	DISEL Bit	Transfer Counter*1,*2	Data Transfer
0	—	0	Other than (1 → 0)	—	—	—	—	Ends after the first transfer
0	—	0	(1 → 0)	—	—	—	—	Ends after the first transfer with an interrupt request to the CPU
0	—	1	—	—	—	—	—	
1	0	—	—	0	—	0	Other than (1 → 0)	Ends after the second transfer
				0	—	0	(1 → 0)	Ends after the second transfer with an interrupt request to the CPU
				0	—	1	—	
1	1	0	Other than (1 → ∗)	—	—	—	—	Ends after the first transfer
1	1	—	(1 → ∗)	0	—	0	Other than (1 → 0)	Ends after the second transfer
				0	—	0	(1 → 0)	Ends after the second transfer with an interrupt request to the CPU
				0	—	1	—	
1	1	1	Other than (1 → ∗)	—	—	—	—	Ends after the first transfer with an interrupt request to the CPU

Note 1. The transfer counters used depend on transfer modes as follows:
Normal transfer mode: CRA register
Repeat transfer mode: CRAL register
Block transfer mode: CRB register

Note 2. On completion of data transfer, the counters operate as follows:
1 → 0 in normal and block transfer modes
1 → CRAH in repeat transfer mode
(1 → ∗) in the table indicates both of the two operations above.

Note 3. Chain transfer can be selected for the second or subsequent transfers. The condition combination of “second transfer and the CHNE bit is 1” is omitted.

18.4.1 Transfer Information Read Skip Function

Reading of DTC vector and transfer information can be skipped by the setting of the DTCCR.RRS bit.

When a DTC transfer request is accepted, the current DTC vector number is compared with the DTC vector number in the previous data transfer. When these vector numbers match and the RRS bit is 1, the DTC does not read the DTC vector and transfer information, and transfers data according to the transfer information remained in the DTC.

However, when the previous transfer was chain transfer, the DTC vector and transfer information are read. Furthermore, when the transfer counter (CRA register) became 0 during the previous normal transfer and when the transfer counter (CRB register) became 0 during the previous block transfer, transfer information is read regardless of the value of the RRS bit. Figure 18.14 shows an example of transfer information read skip.

When updating the vector table and transfer information, set the RRS bit to 0, update the vector table and transfer information, and then set the RRS bit to 1. Setting the RRS bit to 0 discards the vector numbers retained in the DTC. The updated DTC vector table and transfer information are read in the next data transfer.

18.4.2 Transfer Information Write-Back Skip Function

18.4.2.1 Write-Back Skip by Fixing Addresses

When the MRA.SM[1:0] bits or the MRB.DM[1:0] bits are set to “address is fixed” (00b or 01b), a part of transfer information is not written back. This function is performed independently of the setting of short-address mode or full-address mode.

Table 18.5 lists transfer information write-back skip conditions and applicable registers. The CRA and CRB registers are written back independently of the setting of short-address mode or full-address mode.

Furthermore, in full-address mode, write-back of registers MRA, MRB, and MRC is skipped.

Table 18.5 Transfer Information Write-Back Skip Conditions and Applicable Registers

MRA.SM[1:0] Bits		MRB.DM[1:0] Bits			
b3	b2	b3	b2	**SAR Register**	**DAR Register**
0	0	0	0	Skip	Skip
0	0	0	1		
0	1	0	0		
0	1	0	1		
0	0	1	0	Skip	Write-back
0	0	1	1		
0	1	1	0		
0	1	1	1		
1	0	0	0	Write-back	Skip
1	0	0	1		
1	1	0	0		
1	1	0	1		
1	0	1	0	Write-back	Write-back
1	0	1	1		
1	1	1	0		
1	1	1	1		

18.4.2.2 Write-Back Skip by the MRA.WBDIS Bit

When the MRA.WBDIS bit is 1, the transfer information (SAR, DAR, CRA, and CRB) is not written back regardless of the settings of the transfer information.

The transfer information on the memory is not updated, data can be transferred by the DTC without copying the transfer information from ROM to RAM. Skipping a write-back reduces time for post-processing of the data transfer.

18.4.3 Normal Transfer Mode

This mode allows 1-byte, 1-word, or 1-longword data transfer on a single transfer request. The transfer count can be set to 1 to 65536.

Transfer source addresses and transfer destination addresses can be set to increment, decrement, or fixed independently.

This mode enables an interrupt request to the CPU to be generated at the end of specified-count transfer.

Table 18.6 lists register functions in normal transfer mode, and Figure 18.6 shows the memory map of normal transfer mode.

Table 18.6 Register Functions in Normal Transfer Mode

Register	Description	Value Written Back by Writing Transfer Information*1
SAR	Transfer source address	Increment/decrement/fixed*2
DAR	Transfer destination address	Increment/decrement/fixed*2
CRA	Transfer counter A	CRA – 1
CRB	Transfer counter B	Not updated

Note 1. Write-back operation is skipped when the MRA.WBDIS bit is 1.
Note 2. Write-back operation is skipped when address is fixed.

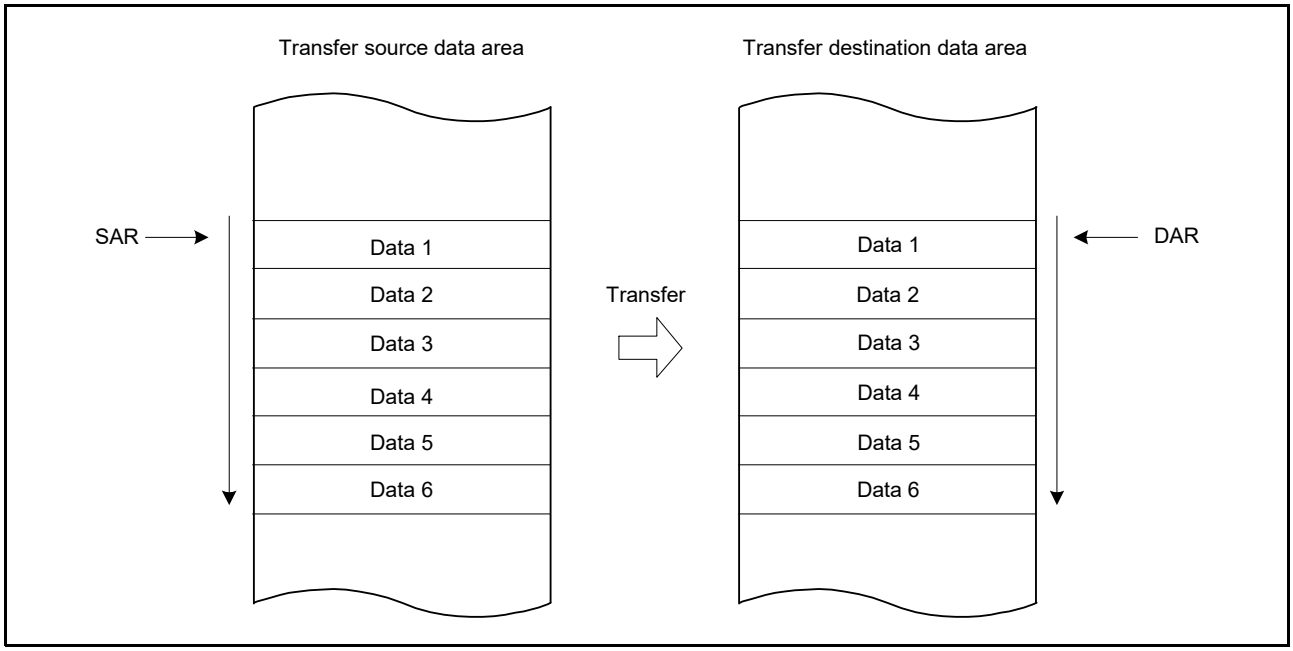


Figure 18.6 Memory Map of Normal Transfer Mode

18.4.4 Repeat Transfer Mode

This mode allows 1-byte, 1-word, or 1-longword data transfer on a single transfer request.
Specify either transfer source or transfer destination for the repeat area by the MRB.DTS bit. The transfer count can be set to 1 to 256. When the specified-count transfer is completed, the initial value of the address register specified in the transfer counter and the repeat area is restored and transfer is repeated. The other address register is incremented or decremented continuously or remains unchanged.
When the transfer counter CRAL is decreased to 00h in repeat transfer mode, the CRAL value is updated to the value set in the CRAH register. Thus the transfer counter does not become 00h, which disables an interrupt request to be generated to the CPU when the MRB.DISEL bit is set to 0 (an interrupt request to the CPU is generated on completion of the specified number of data transfers).
Table 18.7 lists the register functions in repeat transfer mode, and Figure 18.7 shows the memory map of repeat transfer mode.

Table 18.7 Register Functions in Repeat Transfer Mode

		Value Written Back by Writing Transfer Information*1		
			When CRAL = 1	
Register	**Description**	**When CRAL ≠ 1**	**When the MRB.DTS Bit is 0**	**When the MRB.DTS Bit is 1**
SAR	Transfer source address	Increment/decrement/fixed*2	Increment/decrement/fixed*2	SAR register initial value
DAR	Transfer destination address	Increment/decrement/fixed*2	DAR register initial value	Increment/decrement/fixed*2
CRAH	Retains initial value of transfer counter	CRAH	CRAH	
CRAL	Transfer counter A	CRAL – 1	CRAH	
CRB	Transfer counter B	Not updated	Not updated	

Note 1. Write-back operation is skipped when the MRA.WBDIS bit is 1.
Note 2. Write-back operation is skipped when address is fixed.

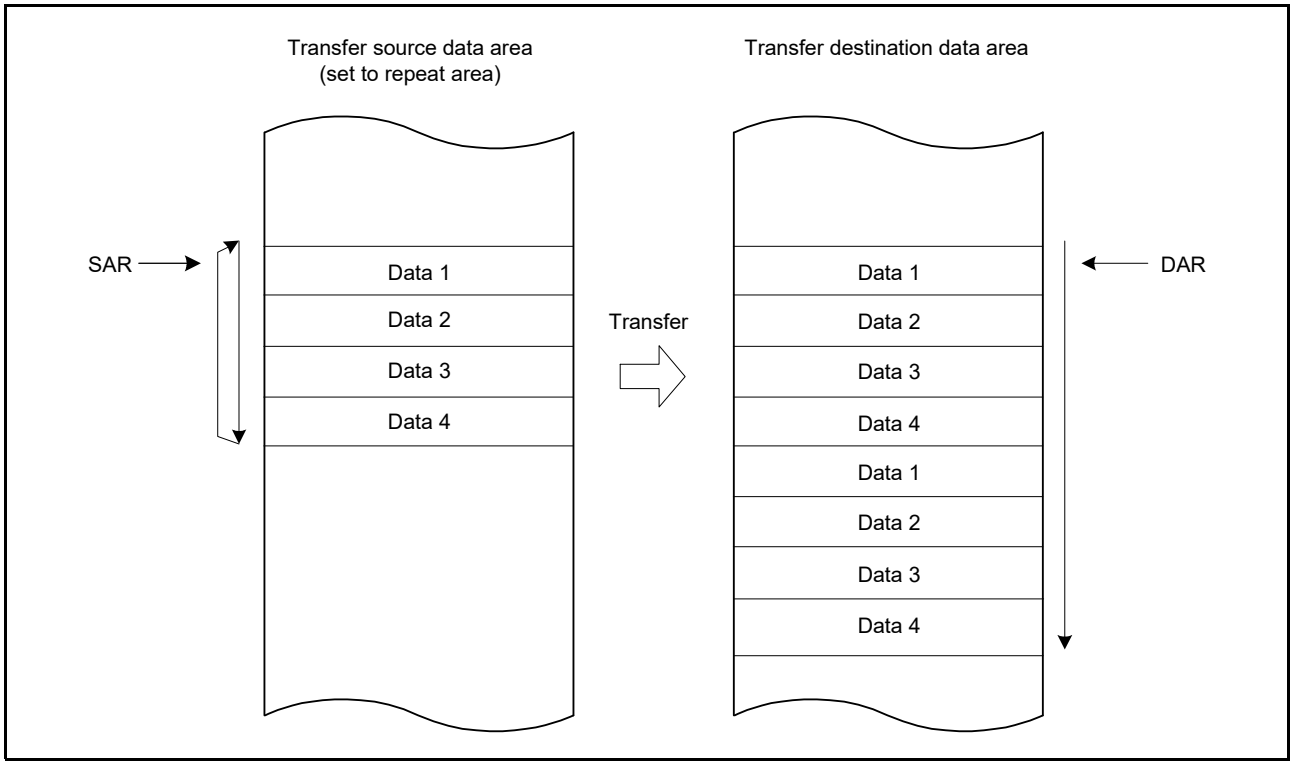


Figure 18.7 Memory Map of Repeat Transfer Mode (Transfer Source: Repeat Area)

18.4.5 Block Transfer Mode

This mode allows single-block data transfer on a single transfer request.
Specify either transfer source or transfer destination for the block area by the MRB.DTS bit. The block size can be set to 1 to 256 bytes, 1 to 256 words, or 1 to 256 longwords.
When transfer of the specified one block is completed, the initial values of the block size counter CRAL and the address register (the SAR register when the MRB.DTS bit is 1 or the DAR register when the DTS bit is 0) specified in the block area are restored. The other address register is incremented or decremented continuously or remains unchanged.
The transfer count (block count) can be set to 1 to 65536. This mode enables an interrupt request to the CPU to be generated at the end of specified-count block transfer.
Table 18.8 lists register functions in block transfer mode, and Figure 18.8 shows the memory map of block transfer mode.

Table 18.8 Register Functions in Block Transfer Mode

		Value Written Back by Writing Transfer Information*1	
Register	**Description**	**When MRB.DTS Bit is 0**	**When MRB.DTS Bit is 1**
SAR	Transfer source address	Increment/decrement/fixed*2	SAR register initial value
DAR	Transfer destination address	DAR register initial value	Increment/decrement/fixed*2
CRAH	Retains initial value of block size	CRAH	
CRAL	Block size counter	CRAH	
CRB	Block transfer counter	CRB – 1	

Note 1. Write-back operation is skipped when the MRA.WBDIS bit is 1.
Note 2. Write-back operation is skipped when address is fixed.

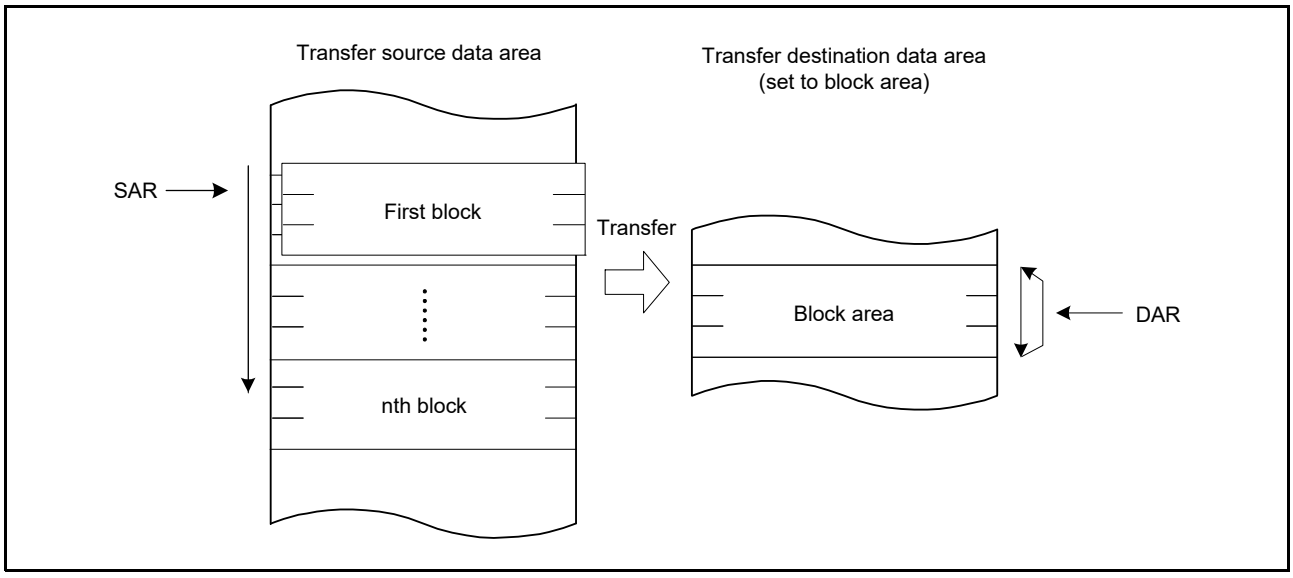


Figure 18.8 Memory Map of Block Transfer Mode (Transfer Destination: Block Area)

18.4.6 Chain Transfer

Setting the MRB.CHNE bit to 1 allows chain transfer to be performed continuously on a single transfer request.
If the MRB.CHNE bit is 1 and the MRB.CHNS bit is 0, an interrupt request to the CPU is not generated when the specified number of data transfers is completed, or while the MRB.DISEL bit is 1 (an interrupt request to the CPU is generated for every data transfer). Data transfer has no effect on the interrupt status flag, which is the request source.
The transfer information (SAR, DAR, CRA, CRB, MRA, MRB, and MRC) that define a data transfer can be specified independently of each other. Figure 18.9 shows chain transfer operation.

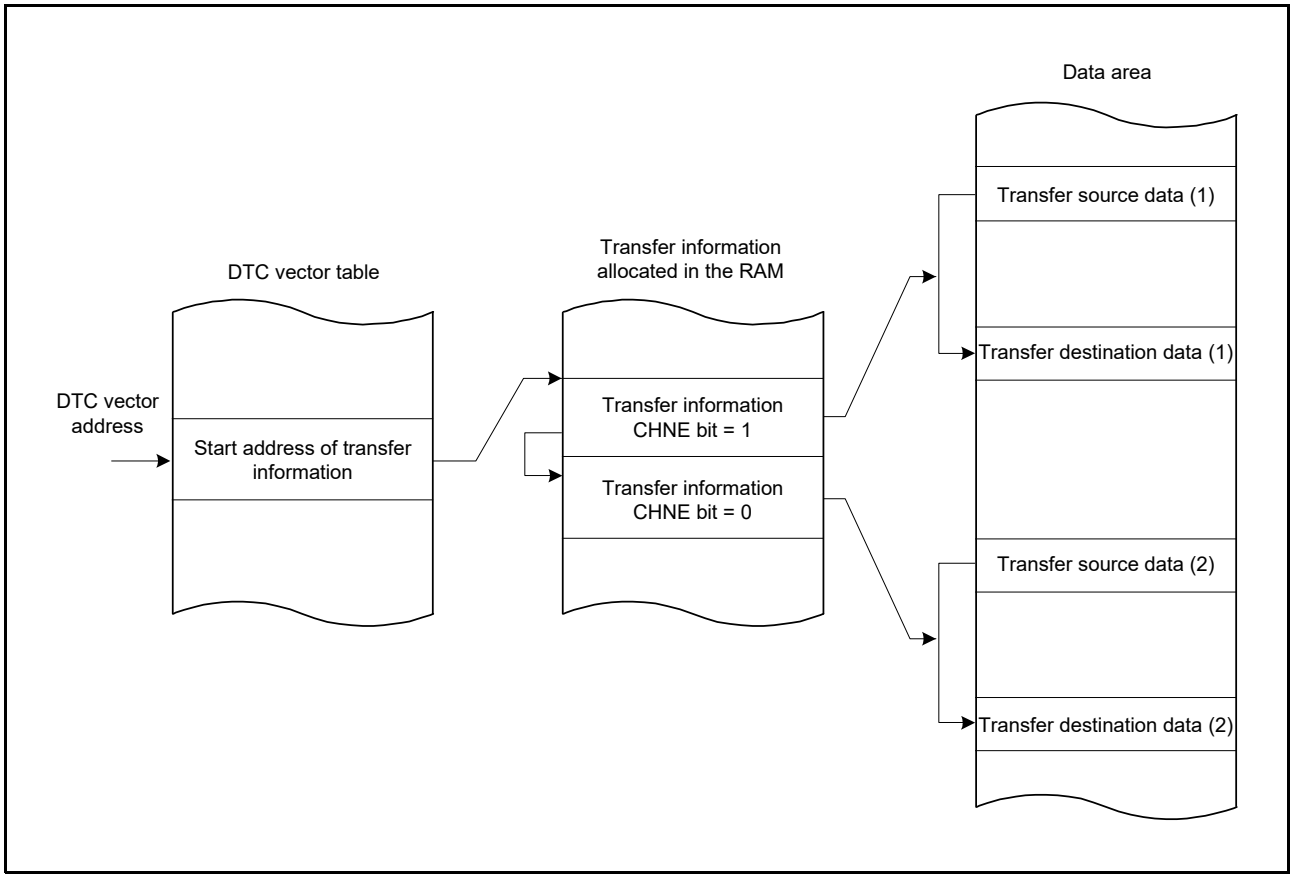


Figure 18.9 Chain Transfer Operation

If the MRB.CHNE bit is 1 and the CHNS bit is 1, chain transfer is performed only after completion of specified number of data transfers. In repeat transfer mode, chain transfer is performed after completion of specified number of data transfers.
For details on chain transfer conditions, refer to Table 18.4, Chain Transfer Conditions.

18.4.7 Operation Timing

Figure 18.10 to Figure 18.14 show examples of DTC operation timing.

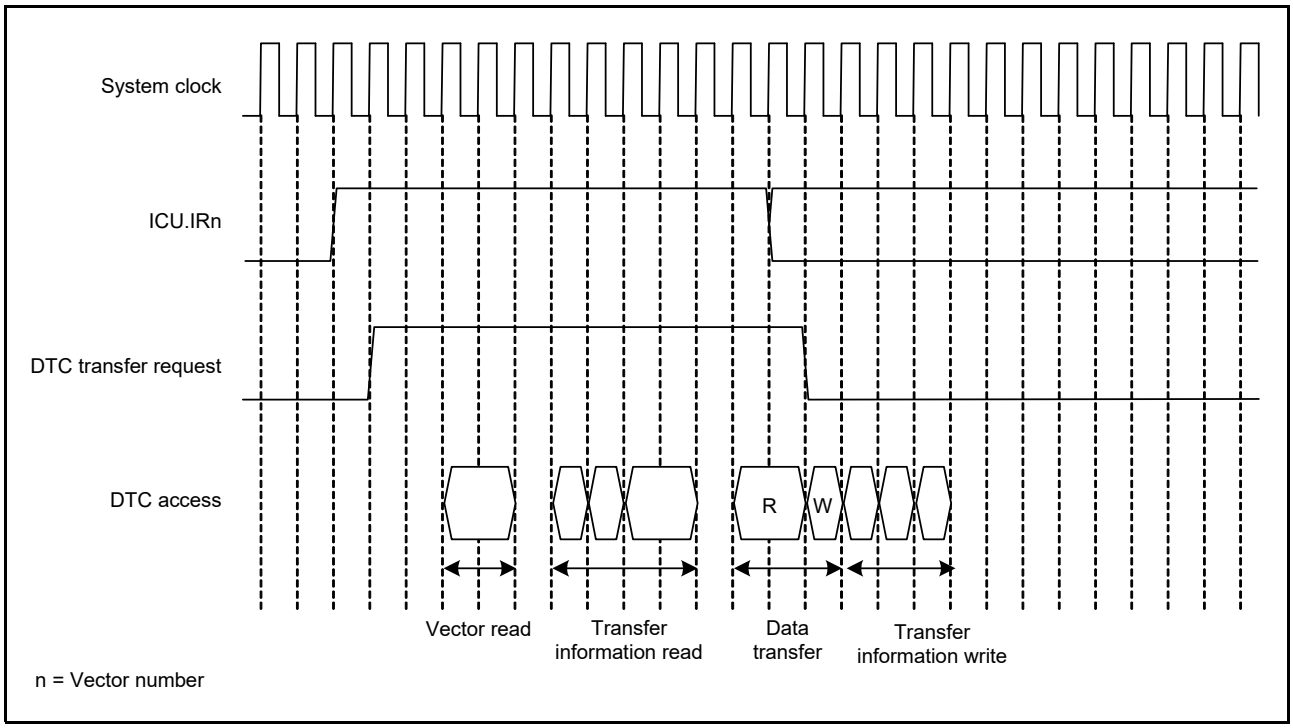


Figure 18.10 Example (1) of DTC Operation Timing (Short-Address Mode, Normal Transfer Mode, Repeat Transfer Mode)

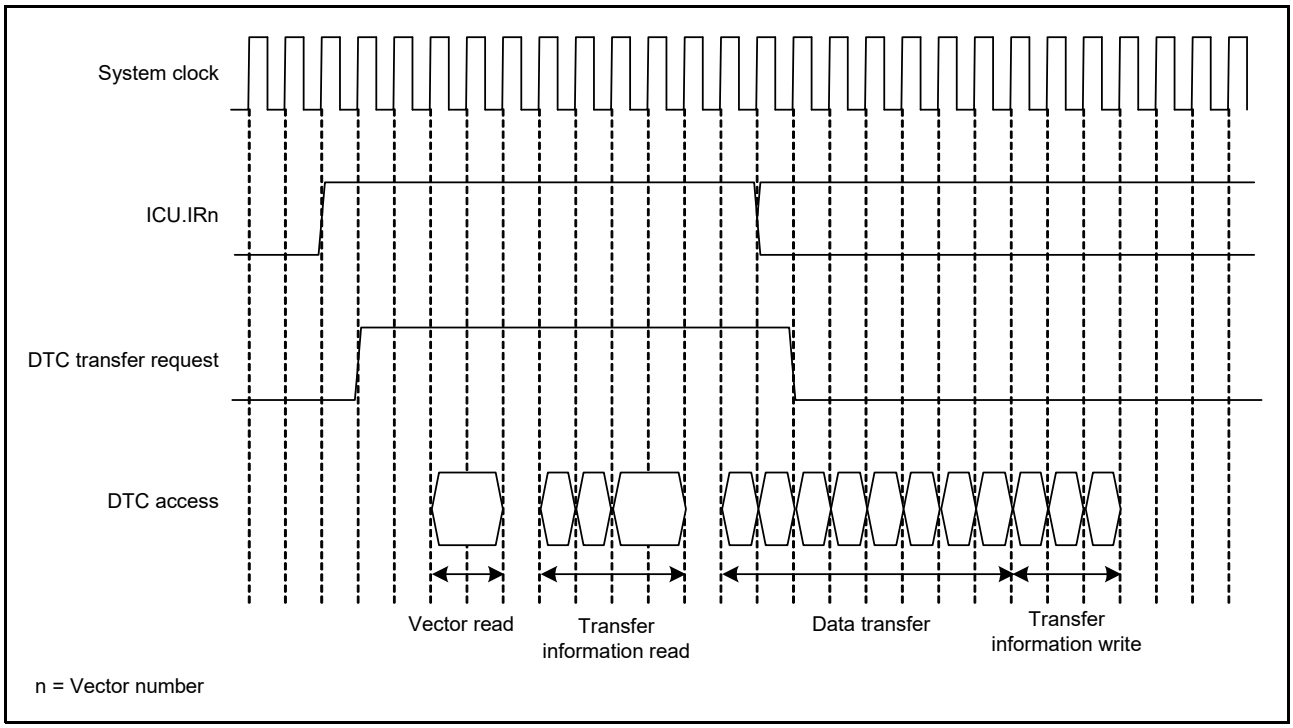


Figure 18.11 Example (2) of DTC Operation Timing (Short-Address Mode, Block Transfer Mode, Block Size = 4)

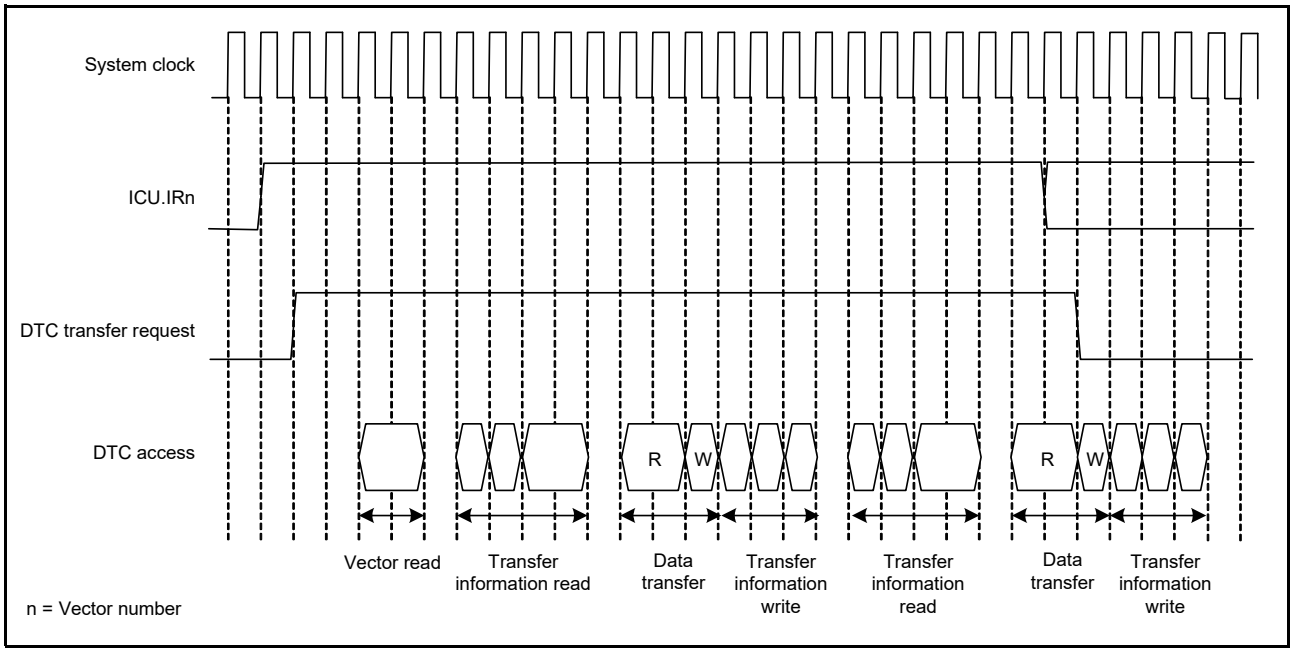


Figure 18.12 Example (3) of DTC Operation Timing (Short-Address Mode, Chain Transfer)

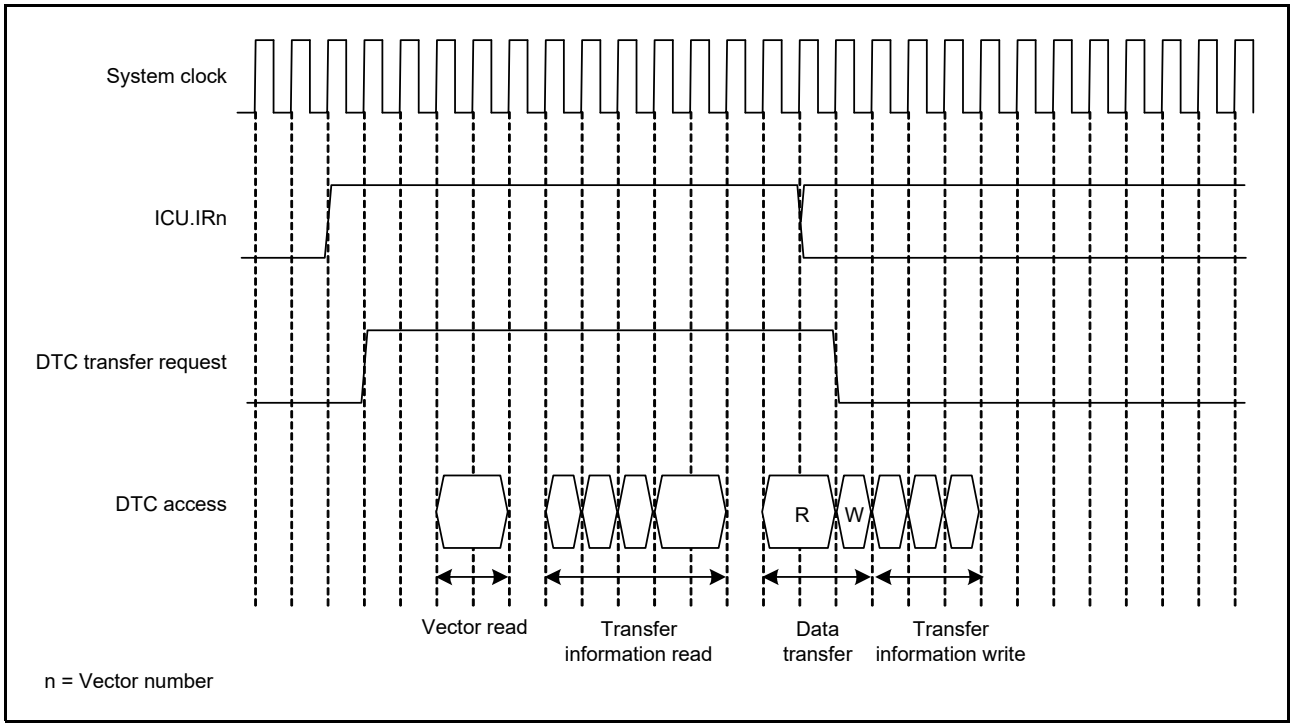
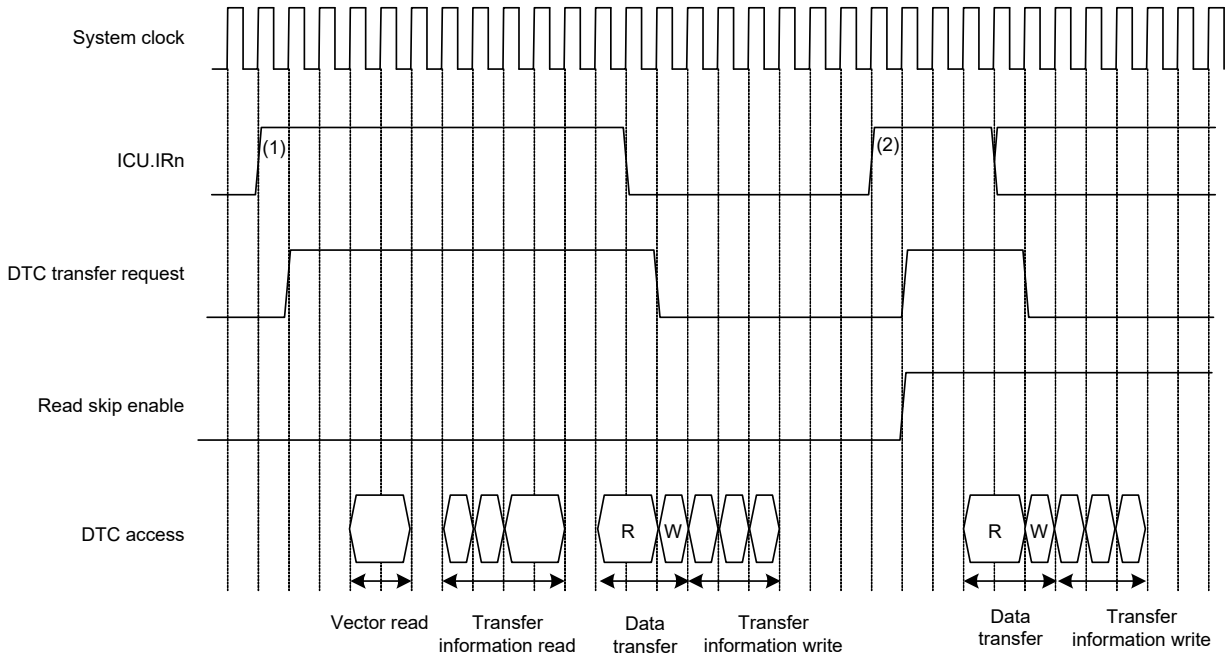


Figure 18.13 Example (4) of DTC Operation Timing (Full-Address Mode, Normal Transfer Mode, Repeat Transfer Mode)



n = Vector number

Note: When request sources (vector numbers) of (1) and (2) are the same and the DTCCR.RRS bit is 1, the transfer information read for request (2) is skipped.

Figure 18.14 Example of Operation When Transfer Information Read Skip is Executed (Vector, Transfer Information, and Transfer Destination Data on the RAM, and Transfer Source Data on the Peripheral Module)

18.4.8 Execution Cycles of the DTC

Table 18.9 lists the execution cycles of single data transfer of the DTC.
For the order of the execution states, refer to section 18.4.7, Operation Timing.

Table 18.9 Execution Cycles of the DTC

Transfer Mode	Vector Read		Transfer Information Read			Transfer Information Write			Data Transfer		Internal Operation	
									Read	Write		
Normal	Cv + 1	0*1	4 × Ci + 1*2	3 × Ci + 1*3	0*1	3 × Ci*4	2 × Ci*5	Ci*6	Cr + 1	Cw	2	0*1
Repeat									Cr + 1	Cw		
Block*7									P × Cr	P × Cw		

Note 1. When transfer information read is skipped
Note 2. In full-address mode
Note 3. In short-address mode
Note 4. When neither SAR nor DAR is set to address-fixed
Note 5. When SAR or DAR is set to address-fixed
Note 6. When SAR and DAR are set to address-fixed
Note 7. When the block size is 2 or more. If the block size is 1, the cycle number for normal transfer is applied.

P: Block size (initial settings of CRAH and CRAL)
Cv: Cycles for access to vector transfer information storage destination
Ci: Cycles for access to transfer information storage destination address
Cr: Cycles for access to data read destination
Cw: Cycles for access to data write destination
(The unit is system clocks (ICLK) for “+ 1” in the Vector Read, Transfer Information Read, and Data Transfer Read columns and “2” in the Internal Operation column.)
(Cv, Ci, Cr, and Cw vary depending on the corresponding access destination. For the number of cycles for respective access destinations, refer to section 47, RAM, section 48, Flash Memory (FLASH), and section 5, I/O Registers.)

18.4.9 DTC Bus Mastership Release Timing

The DTC does not release the bus mastership during transfer information read and transfer information write. While transfer information is not read or written, bus arbitration is made according to the priority determined by the bus master arbitrator.
For bus arbitration, refer to section 15, Buses.

18.4.10 Sequence Transfer

A sequence transfer can be executed on the request source that is specified in the DTCSQE register. A sequence transfer is started by setting the MRB.INDX bit to 1 and ended by setting the MRB.SQEND bit to 1. Setting the DTCOR.SQTFRL bit to 1, even during a sequence transfer, forcibly ends the transfer and the next DTC transfer request starts new sequence transfer with reference to the index table.
A sequence transfer includes the following processing.

(1) The first data transfer is executed by referring to the DTC vector table in response to a DTC transfer request from the source specified in the DTCSQE register.
(2) The DTC index table is referred based on the value of the lower 8 bits of the first data transferred data in (1) (sequence number).
(3) The transfer information is read from the address obtained from the DTC index table.
(4) A data transfer is executed in accordance with the transfer information. On completion of the data transfer, either one of the following operations is performed by using the values of bits MRB.CHNE and MRB.SQEND.
- A chain transfer is executed when the CHNE bit is 1. The next transfer information is read. Go to (4).
- The sequence transfer is suspended when the CHNE bit is 0 and the SQEND bit is 0. Go to (5).
- The sequence transfer ends when the CHNE bit is 0 and the SQEND bit is 1.

(5) When a DTC transfer request from the source specified in the DTCSQE register is accepted, the suspended sequence transfer is resumed and the next transfer information is read. Go to (4).

Note 1. When the ICU.DTCERn.DTCE bit becomes 0 based on the result of the data transfer, a DTC transfer request is not generated. Set the DTCE bit to 1 to resume sequence transfer. Refer to Figure 18.5 or section 14, Interrupt Controller (ICUG) for the conditions where the DTCE bit becomes 1.

Figure 18.15 and Figure 18.16 shows a basic sequence transfer operation.

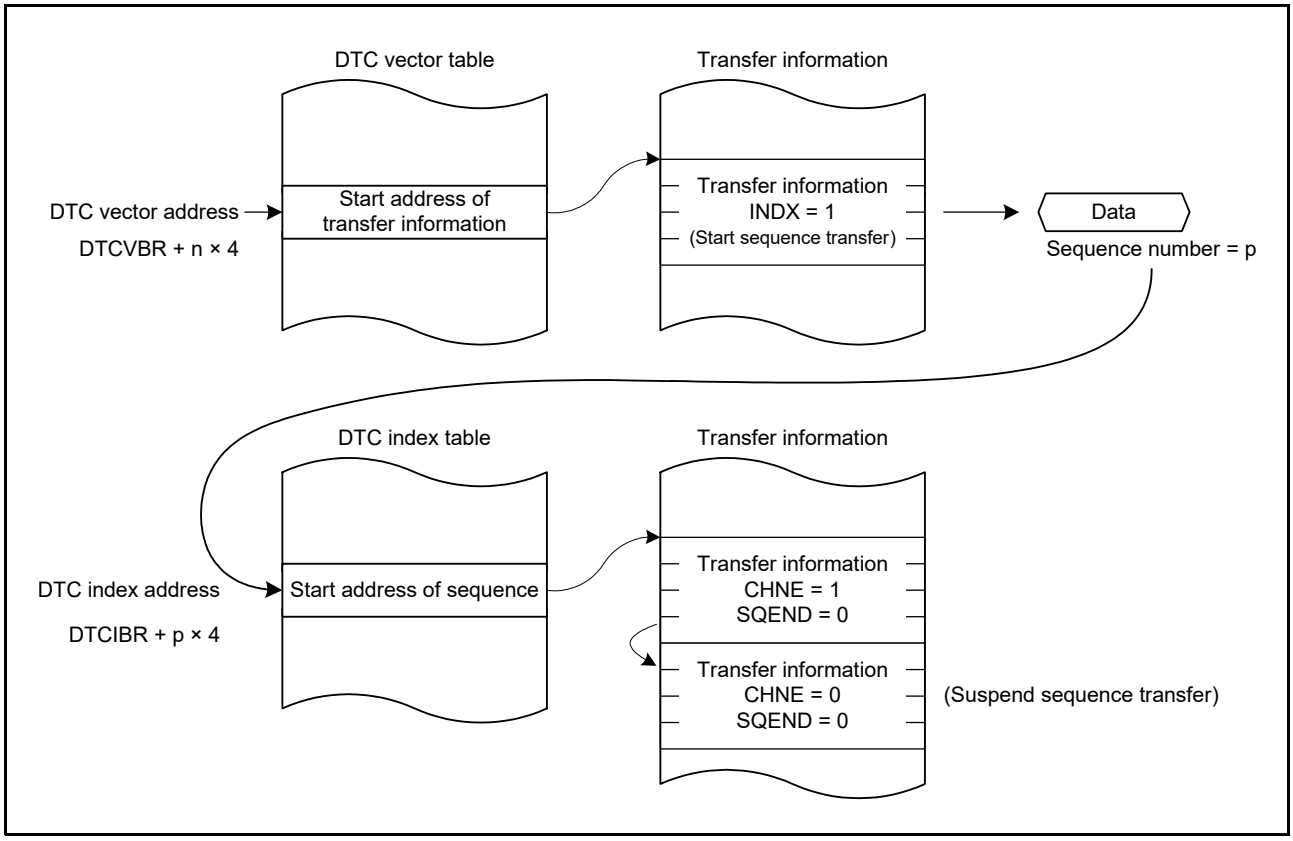


Figure 18.15 Start and Suspension of Sequence Transfer

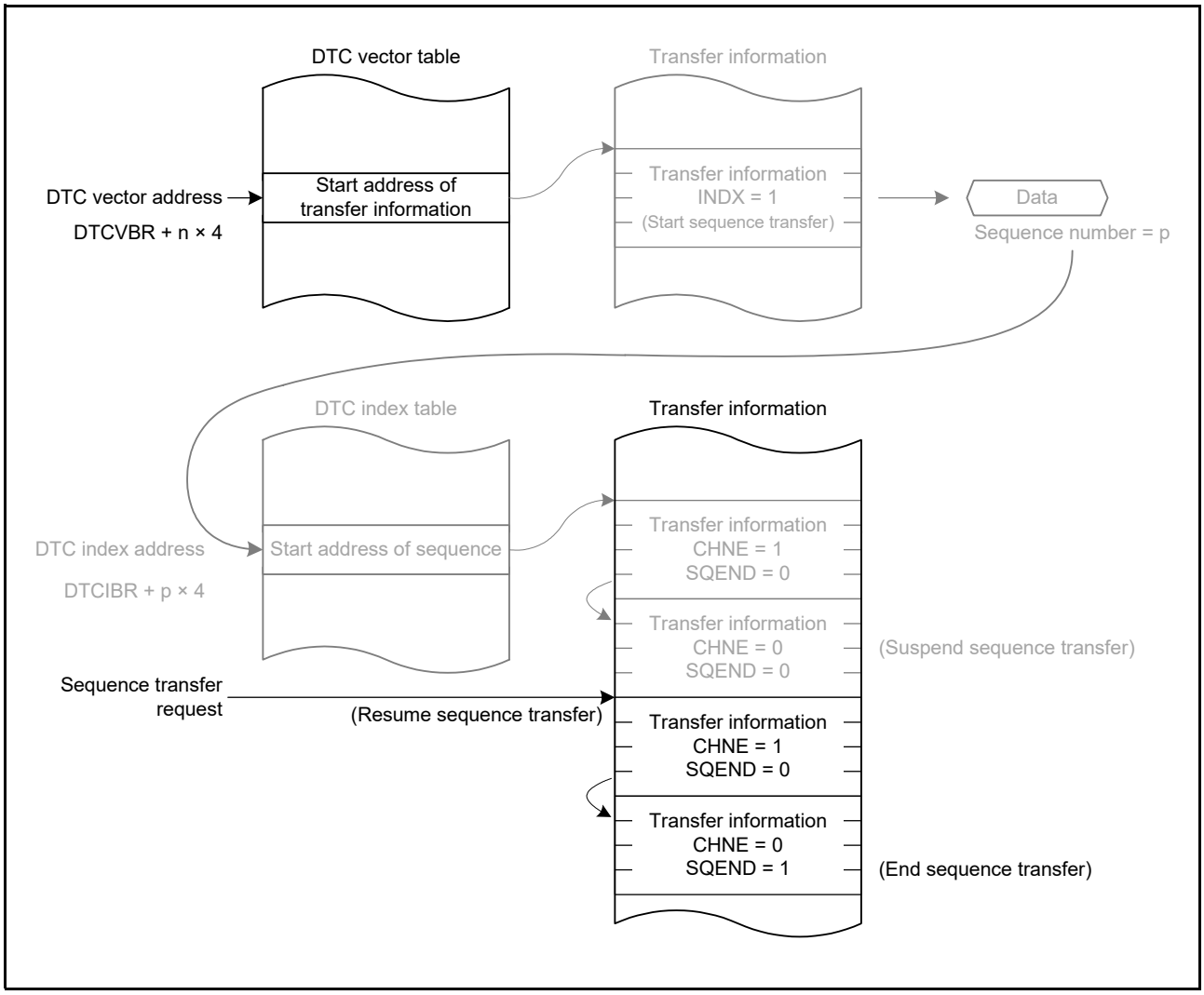


Figure 18.16 Resumption and End of Sequence Transfer

Table 18.10 lists the settings of bits CHNE, SQEND, and INDX during the sequence transfer.

Table 18.10 Sequence Transfer Process and Values of Bits CHNE, SQEND, and INDX

DTC Operations	CHNE Bit	SQEND Bit	INDX Bit
Start sequence transfer	0	0	1*1
Continue sequence transfer	1	0	0
Suspend sequence transfer*2	0	0	0
End sequence transfer	0	1	0
End current sequence transfer and Obtain new sequence number	0	1	1*1
Some other transfer (not sequence transfer)	—	0	0

Note: Do not set the values other than listed above.
Note 1. Set MRA.MD[1:0] bits to 00b (normal transfer mode) when setting the INDX bit to 1.
Note 2. When a sequence transfer is suspended, the ICU.DTCERn.DTCE bit may become 0. Set the DTCE bit to 1 to resume sequence transfer.

Even when a sequence transfer is suspended, a new sequence transfer cannot start until the suspended sequence transfer is eventually completed. When a sequence transfer request is received during suspension of the sequence transfer, the suspended sequence transfer is resumed.

18.4.11 DTC Index Table

The DTC index table is allocated to the area where its start address is configured in the DTCIBR register.
Store the start address of transfer information table p for sequence number p in the address of DTCIBR + p × 4.
The upper 30 bits of the start address is set to the upper 30 bits of the DTC index. Set the CPUSEL bit to select either of reading the transfer information and starting the sequence, or output an interrupt request to the CPU without starting the sequence. For a complicated sequence that the DTC cannot handle, set the CPUSEL bit to 1 to allow the CPU to handle such a sequence.

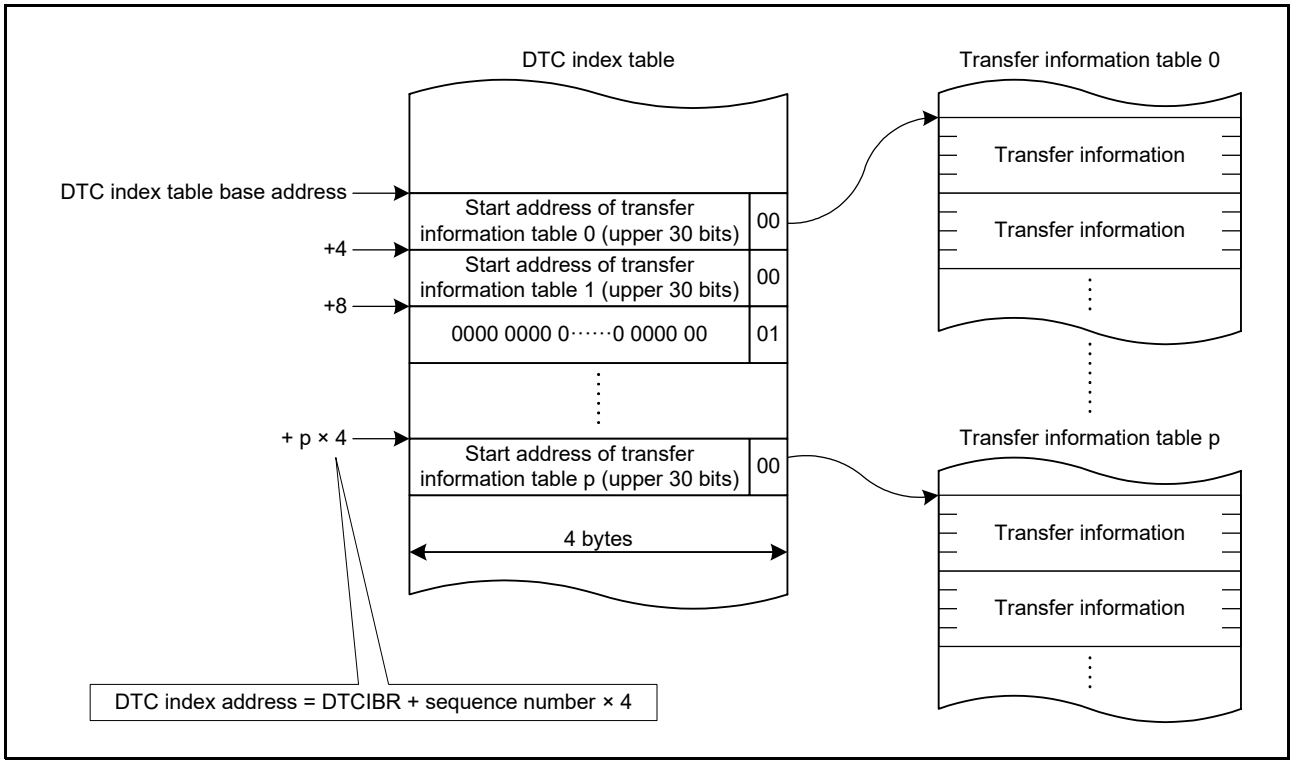
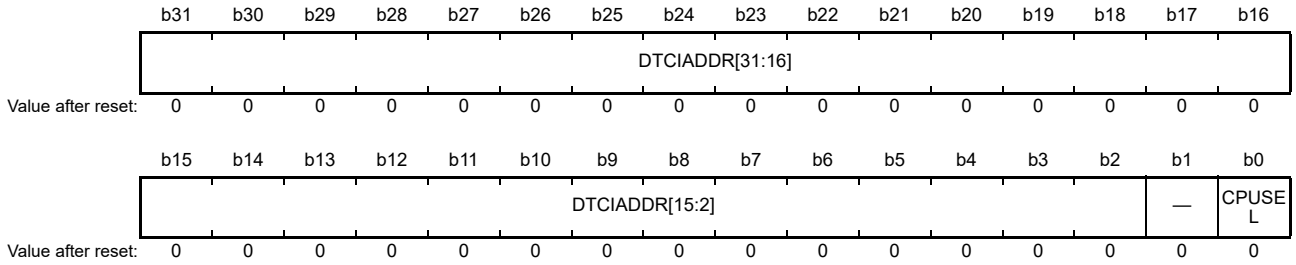


Figure 18.17 DTC Index Table

- DTC Index

Address(es): DTCIBR + p × 4



Bit	Symbol	Bit Name	Description	R/W
b0	CPUSEL	Sequence Transfer/CPU Interrupt Select	0: Continues the sequence transfer (starts the sequence) 1: Ends the sequence transfer and outputs an interrupt request to the CPU	—
b1	—	Reserved	Set this bit to 0.	—
b31 to b2	DTCIADDR[31:2]	Transfer Information Table Address	Set the upper 30 bits of the start address of the transfer information table to these bits. Writing to the upper 4 bits (b31 to b28) is ignored and the values in b31 to b28 become the same value as b27.	—

When the CPUSEL bit in the DTC index that the obtained sequence number indicates is 1, an interrupt request to the CPU is generated. At this time, the ICU.DTCERn.DTCE bit becomes 0. From this point, the interrupt request signal from the request source that is specified in the DTCSQE register is sent to the CPU, but not DTC. After completion of CPU interrupt processing, set the ICU.DTCERn.DTCE bit to 1 to enable DTC transfer request for starting the next sequence transfer.

18.4.12 Example of Sequence Transfer

Figure 18.18 shows a typical examples of a sequence transfer and Figure 18.19 to Figure 18.23 show configurations of the transfer information for the examples of the transfers in the figure.
In these examples, the interrupt source of vector number n is set as the source of the sequence transfer (DTCSQE.VECN[7:0] = n).
Once the DTC transfer request due to the interrupt source of vector number n (hereinafter referred to as “transfer request n”) is input, the DTC refers to the DTC vector table and reads the corresponding transfer information. The lower 8 bits that have been transferred based on the transfer information become a sequence number, selecting one sequence among possible 256 sequences.

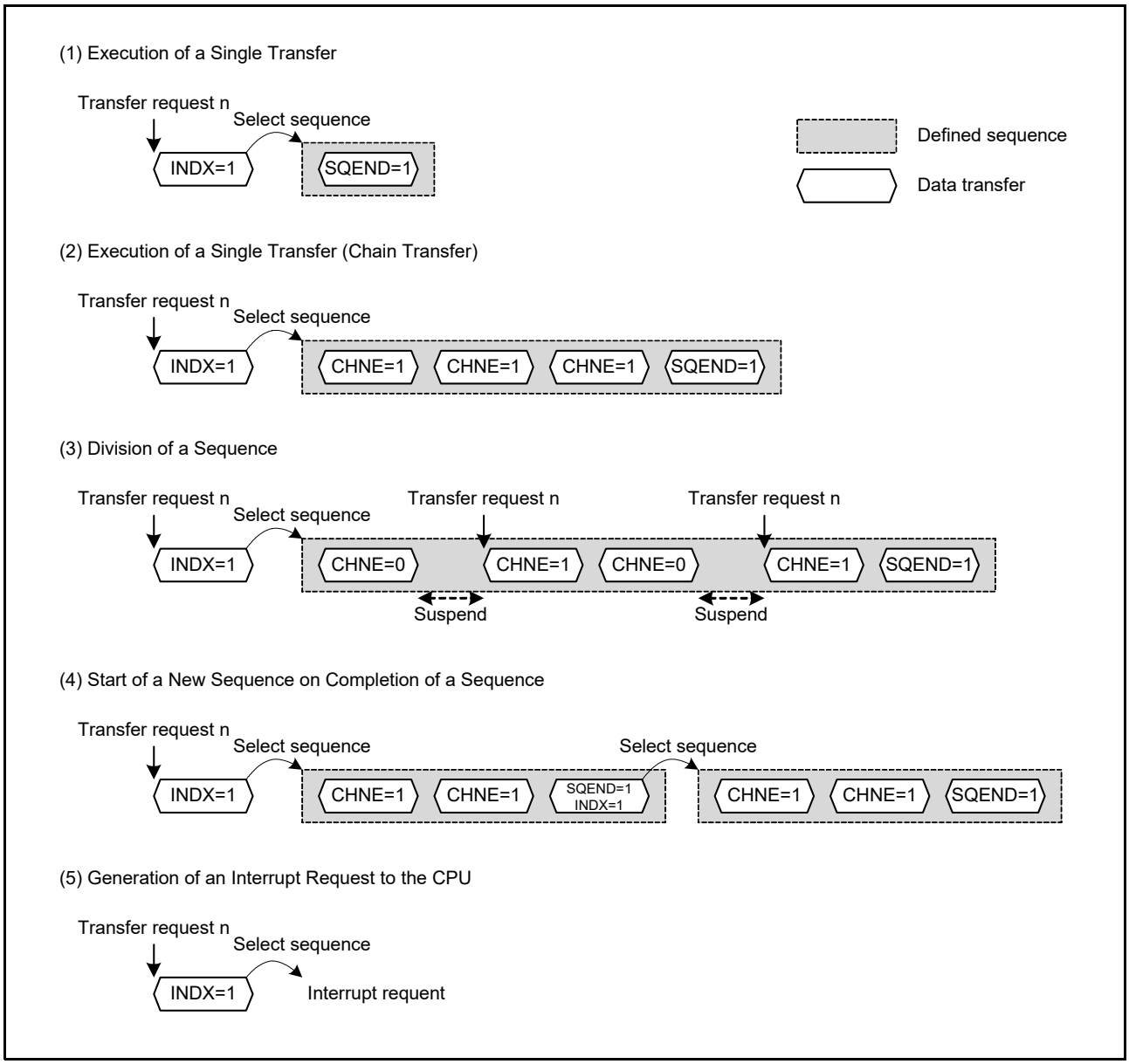


Figure 18.18 Examples of Sequence Transfers

(1) When Executing a Single Transfer

Figure 18.19 shows an example of a single transfer (normal, repeat, or block).
The DTC refers to the DTC index table, and reads the transfer information corresponding to the obtained sequence number p.
Since the values of the CHNE, INDX, and SQEND bits are 0, 0, and 1 respectively, the sequence ends after the specified transfer is executed.

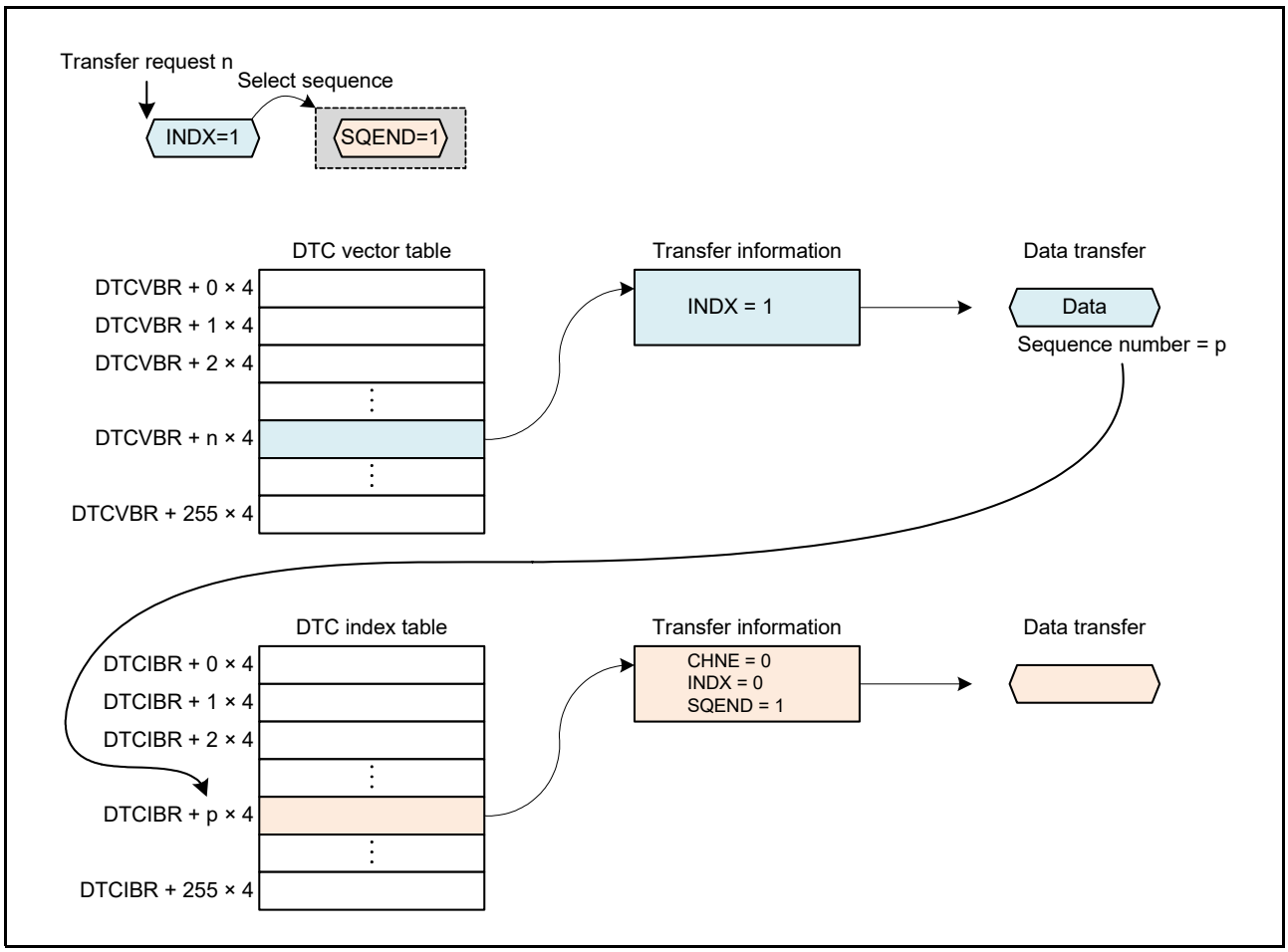


Figure 18.19 Example of a Sequence of Single Transfer

(2) When Executing a Single Chain Transfer

Figure 18.20 is an example of a sequence for a single chain transfer.
The DTC refers to the DTC index table, and reads the transfer information corresponding to the obtained sequence number q.
While the values of the CHNE, INDX, and SQEND bits are 1, 0, and 0 respectively, the specified chain transfer is executed. When the transfer information in which the CHNE, INDX, and SQEND bits are 0, 0, and 1 respectively is read, the sequence ends after the specified transfer is executed.

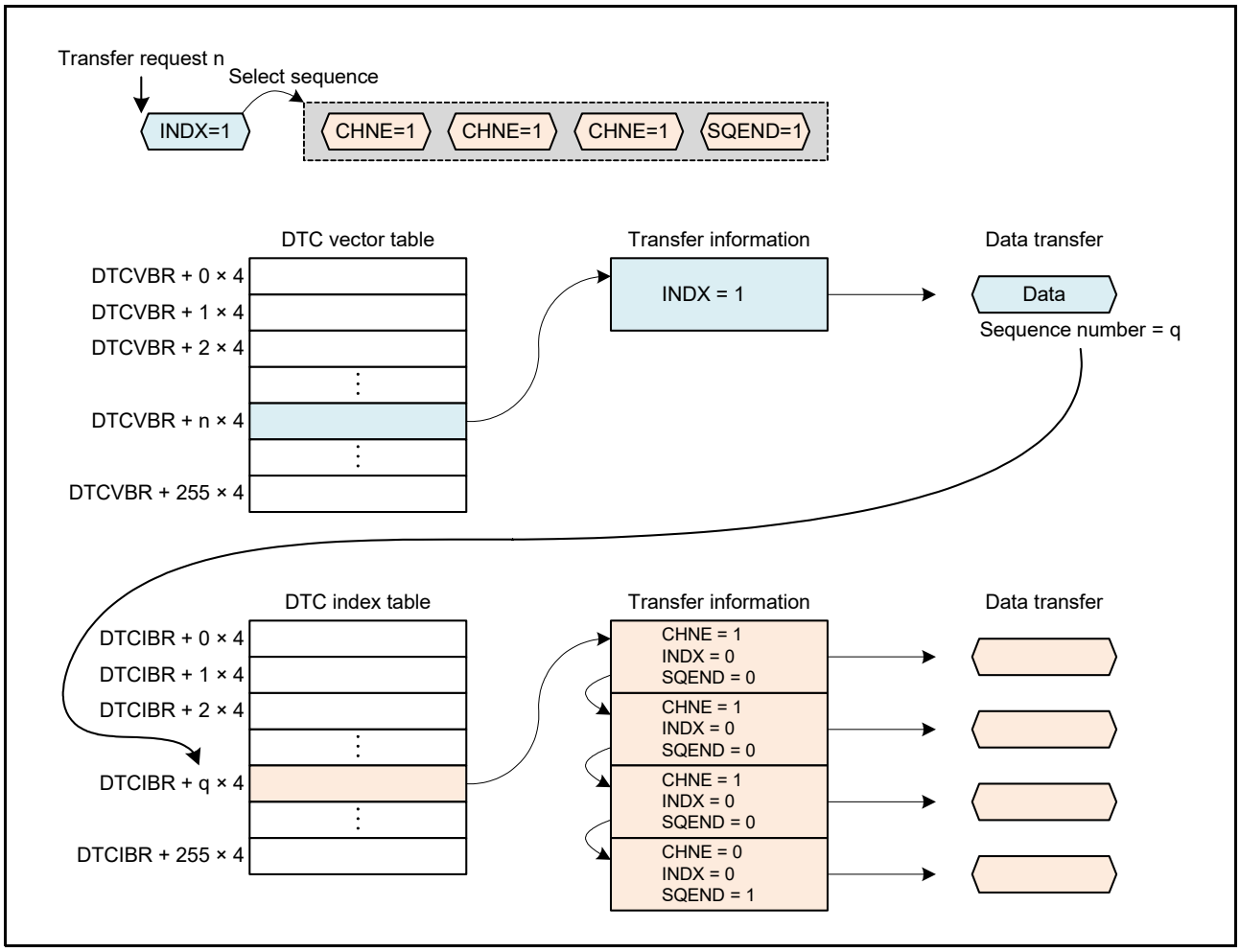


Figure 18.20 Example of Sequence of a Single Chain Transfer

(3) When Dividing a Sequence

Figure 18.21 is an example of the sequence that is divided into 3 parts.
The DTC refers to the DTC index table, and reads the transfer information corresponding to the obtained sequence number r.
Since the values of the CHNE, INDX, and SQEND bits in the transfer information are 0, 0, and 0 respectively, the sequence is suspended after the specified transfer is executed and the DTC waits for the next transfer request n.
When the transfer request n is input during a sequence transfer, the DTC vector table is not referred and the suspended sequence is resumed.
When the transfer information in which the CHNE, INDX, and SQEND bits are 0, 0, and 1 respectively is read, the sequence ends after the specified transfer is executed.

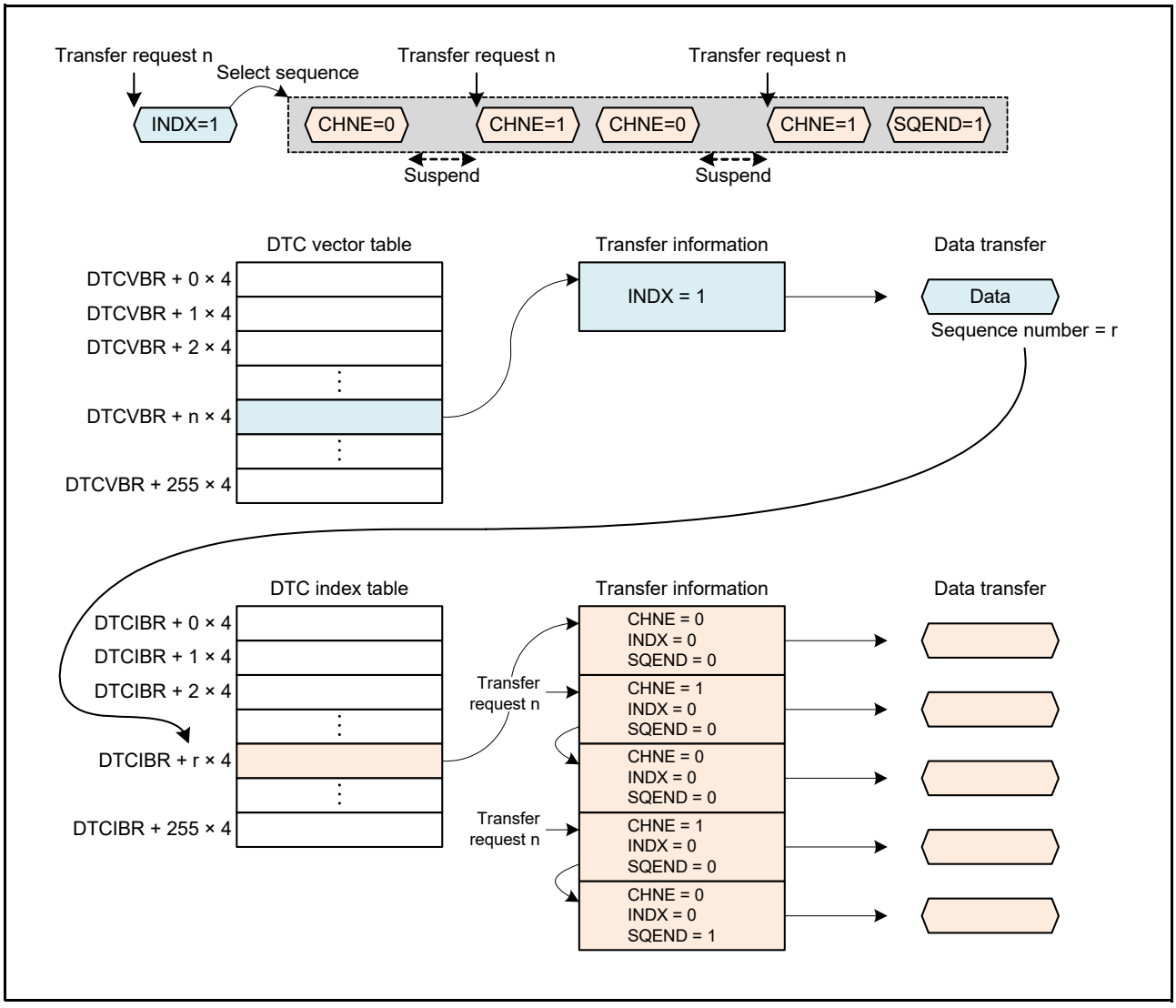


Figure 18.21 Example of Divided Sequence

(4) When Starting a New Sequence on completion of a Sequence

Figure 18.22 is an example for starting the next and new sequence on completion of the first sequence.
The DTC refers to the DTC index table, and reads the transfer information corresponding to the obtained sequence number s.
When the transfer information in which the CHNE, INDX, and SQEND bits are 0, 1, and 1 respectively is read, the specified transfer is executed, then a new sequence number is obtained from the lower 8 bits of the transferred data.
The DTC again refers to the DTC index table, and reads the transfer information corresponding to the obtained sequence number k and then starts a new sequence.
When the transfer information in which the CHNE, INDX, and SQEND bits are 0, 0, and 1 respectively is read, the sequence ends after the specified transfer is executed.

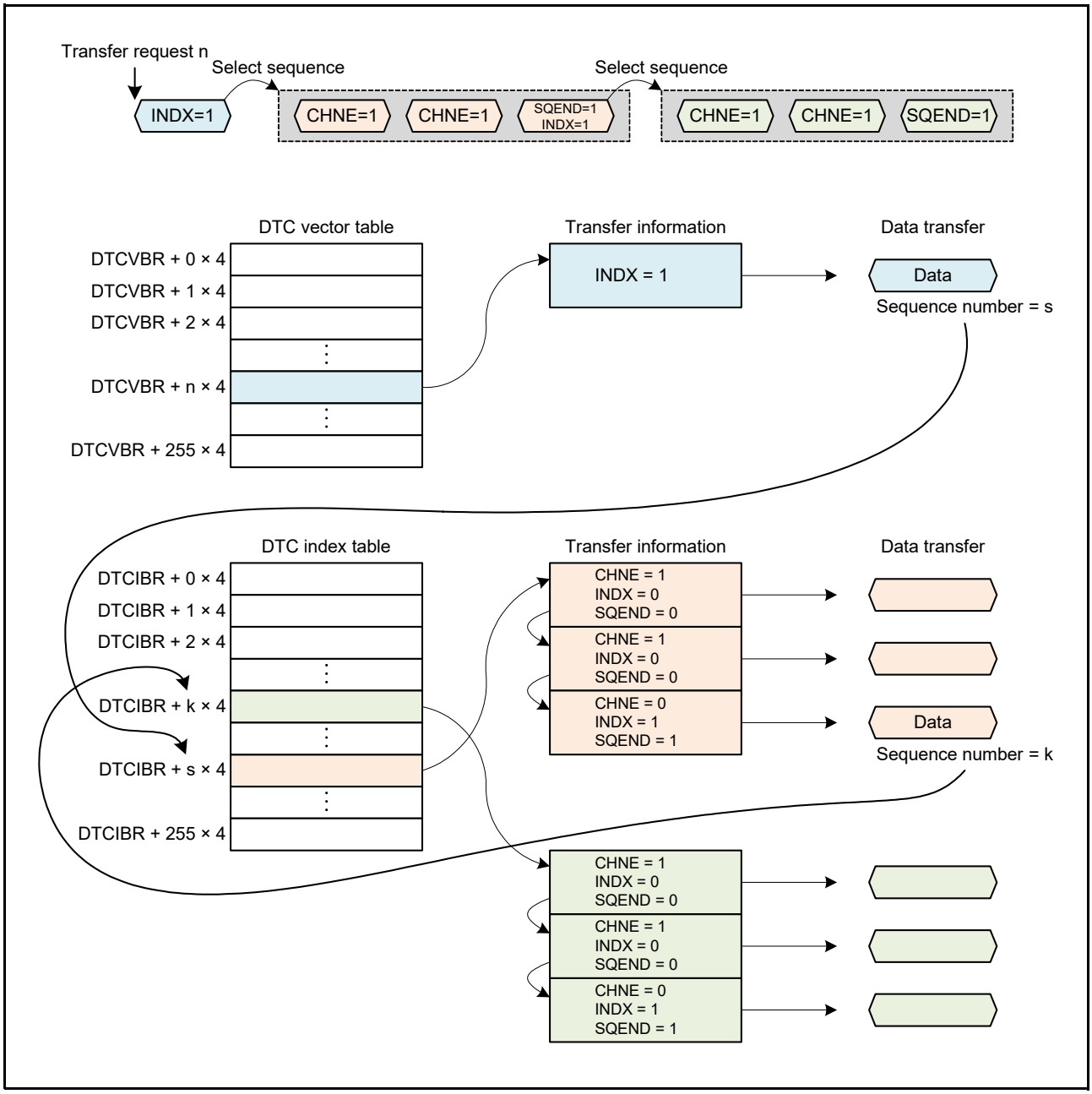


Figure 18.22 Example When Starting a New Sequence on Completion of a Sequence

(5) When Generating an Interrupt Request to the CPU

Figure 18.23 is an example of that an interrupt request is output to the CPU without starting of sequence.
The DTC obtains a DTC index that corresponds to the obtained sequence number t.
When the CPUSEL bit of the obtained DTC index is 1, the DTC ends the sequence transfer without starting the sequence, and then outputs an interrupt request to the CPU.

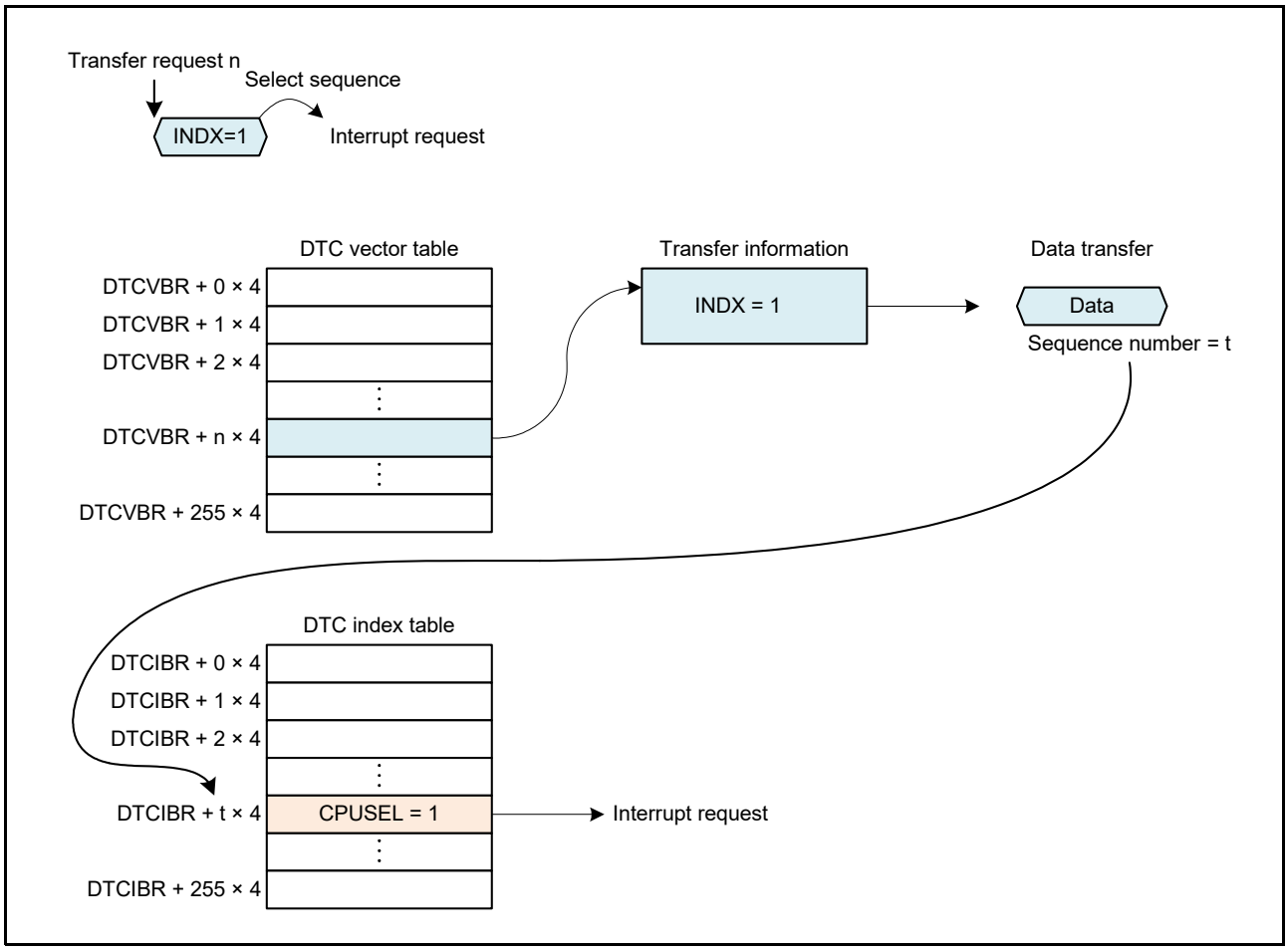


Figure 18.23 Example of Output of an Interrupt Request to the CPU

18.5 DTC Setting Procedure

Before using the DTC, set the DTC vector base register (DTCVBR). When using sequence transfer, also set the DTC index table base register (DTCIBR).

Figure 18.24 shows the procedure to set the DTC.

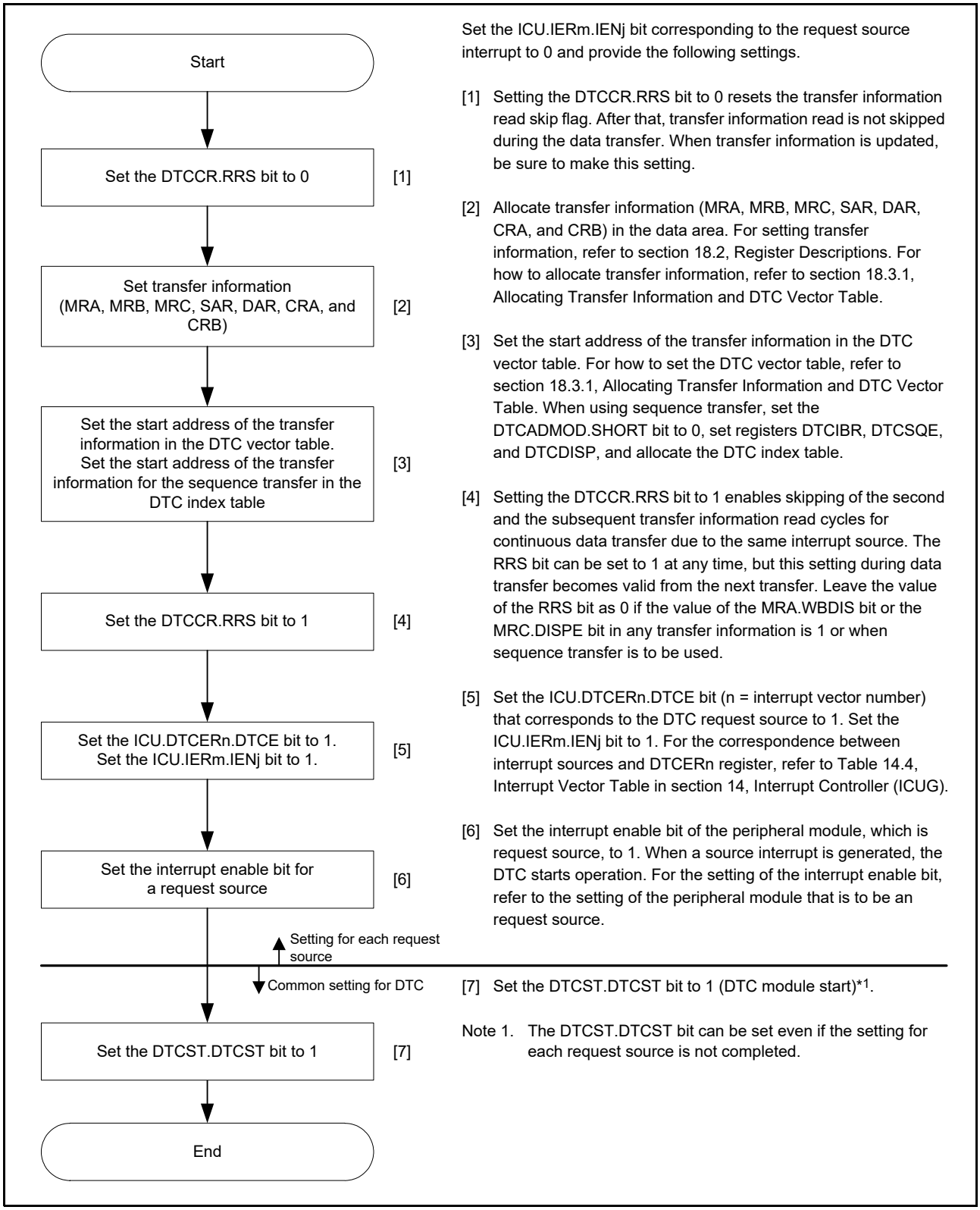


Figure 18.24 Procedure to Set the DTC

18.6 Examples of DTC Usage

18.6.1 Normal Transfer

As an example of DTC usage, its employment in the reception of 128 bytes of data by an SCI is described below.

(1) Transfer Information Setting

Set the MRA.MD[1:0] bits to 00b (normal transfer mode), the MRA.SZ[1:0] bits to 00b (byte transfer), and the MRA.SM[1:0] bits to 00b (source address is fixed). Set the MRB.CHNE bit to 0 (chain transfer is disabled), the MRB.DISEL bit to 0 (an interrupt request to the CPU is generated on completion of the specified number of data transfers), and the MRB.DM[1:0] bits to 10b (DAR is incremented after data transfer). The MRB.DTS bit can be set to any value. Set the RDR register address of the SCI in the SAR register, the start address of the RAM area for data storage in the DAR register, and 128 (0080h) in the CRA register. The CRB register can be set to any value.

(2) DTC Vector Table Setting

The start address of the transfer information for the RXI interrupt is set in the vector table for the DTC.

(3) ICU Setting and DTC Module Activation

Set the corresponding ICU.DTCERn.DTCE bit to 1 and the ICU.IERm.IENj bit to 1.
Set the DTCST.DTCST bit to 1.

(4) SCI Setting

Enable the RXI interrupt by setting the SCR.RIE bit in the SCI to 1. If a reception error occurs during the SCI receive operation, further reception is not performed. Accordingly, make settings so that the CPU can accept receive error interrupts.

(5) DTC Transfer

Every time the reception of 1 byte by the SCI is completed, an RXI interrupt is generated to start the data transfer. The DTC transfers the received byte from the RDR of the SCI to RAM, after which the DAR register is incremented and the CRA register is decremented.

(6) Interrupt Handling

After 128 times of data transfers have been completed and the value in the CRA register becomes 0, an RXI interrupt request is output to the CPU. Complete the process in the handling routine for this interrupt.

18.6.2 Chain Transfer When the Counter is 0

The second data transfer is performed only when the transfer counter is set to 0 in the first data transfer, and the first data transfer information is repeatedly changed in the second data transfer. Repeating this chain transfer enables transfers to be repeated more than 256 times.
The following shows an example of configuring a 128-Kbyte input buffer to addresses 20 0000h to 21 FFFFh (where the input buffer is set so that its lower address starts with 0000h). Figure 18.25 shows a chain transfer when the counter is 0.

(1) Set normal transfer mode for input data for the first data transfer. Set the following:
Transfer source address: Fixed, the CRA register is 0000h (65,536 times), the MRB.CHNE bit is 1 (chain transfer is enabled), the MRB.CHNS bit is 1 (chain transfer is performed only when the transfer counter becomes 0), and the MRB.DISEL bit is 0 (an interrupt request to the CPU is generated on completion of the specified number of data transfers).
(2) Prepare the upper 8 bits (in this case, 21h and 20h) of the start address at every 65,536 times of the transfer destination address for the first data transfer in another area (such as ROM).
(3) For the second data transfer, set repeat transfer mode (source is repeat area) for rewriting the transfer destination address of the first data transfer. The transfer destination is the address where the upper 8 bits of the DAR register in the first transfer information is allocated. In this case, set the MRB.CHNE bit to 0 (chain transfer is disabled) and the MRB.DISEL bit to 0 (an interrupt request to the CPU is generated on completion of the specified number of data transfers). In this case, set the transfer counter to 2.
(4) When a transfer request is accepted, the first data transfer is executed. When transfer is executed 65,536 times and the transfer counter of the first data transfer becomes 0, the second data transfer is started and the upper 8 bits of the transfer destination address of the first data transfer is set to 21h. At this time, the lower 16 bits of the transfer destination address and the transfer counter of the first data transfer have become 0000h.
(5) In succession, when another transfer request is accepted, the first data transfer is executed. When transfer is executed 65,536 times and the transfer counter of the first data transfer becomes 0, the second data transfer is started and the upper 8 bits of the transfer destination address of the first data transfer is set to 20h. At this time, the lower 16 bits of the transfer destination address and the transfer counter of the first data transfer have become 0000h.
(6) Steps (4) and (5) above are repeated infinitely. Because the second data transfer is in repeat transfer mode, no interrupt request to the CPU is generated.

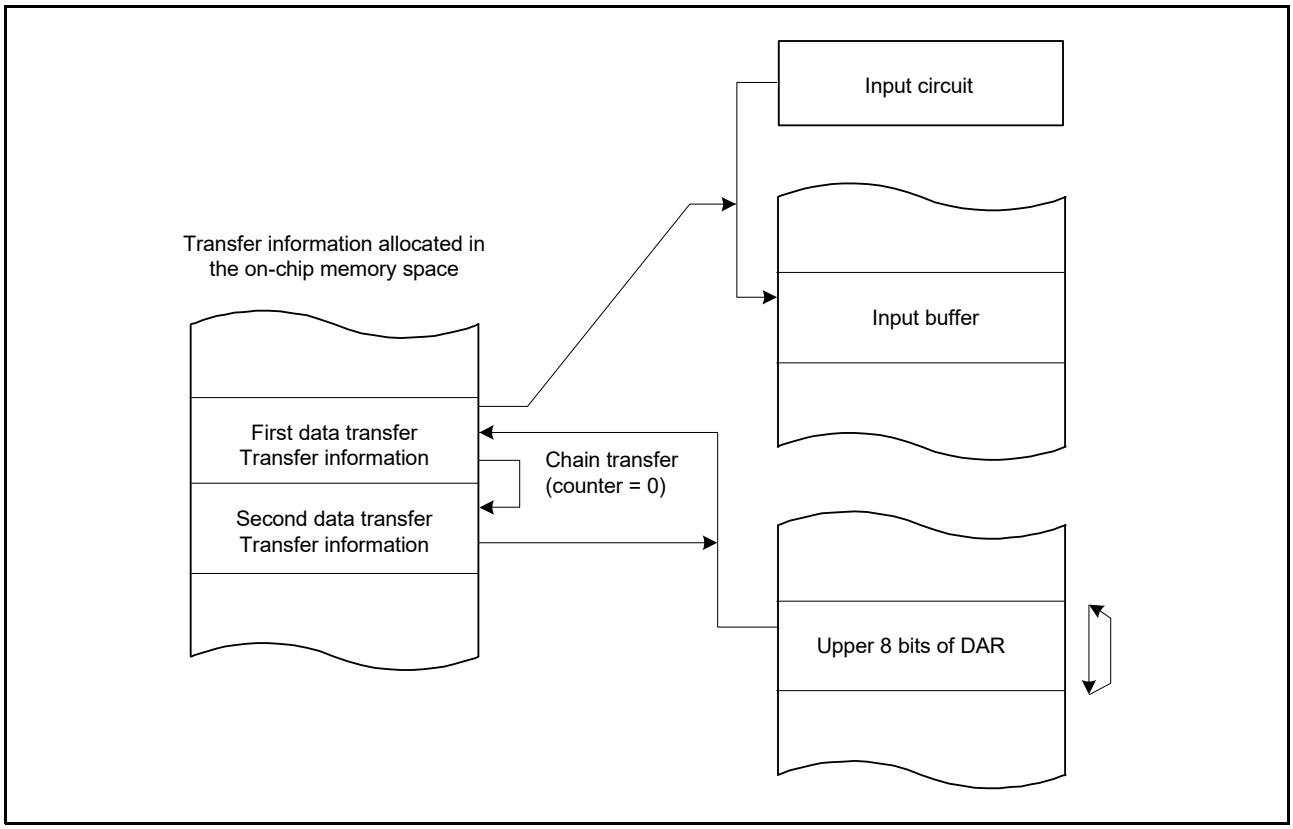


Figure 18.25 Chain Transfer When the Counter is 0

18.6.3 Sequence Transfer

The following is an example of using the SCI receive interrupt as a request source of sequence transfer.

(1) Transfer Information Settings

Set the MRA.MD[1:0] bits to 00b (normal transfer mode), the MRA.SZ[1:0] bits to 00b (byte transfer), and the MRA.SM[1:0] bits to 00b (source address is fixed). Set the MRB.CHNE bit to 0 (chain transfer is disabled), the MRB.DISEL bit to 0 (an interrupt request to the CPU is generated on completion of the specified number of data transfers), the MRB.DM[1:0] bits to 10b (DAR is incremented after data transfer), the MRB.INDX bit to 1 (start sequence transfer), and the MRB.SQEND bit to 0 (continue the sequence transfer). The MRB.DTS bit can be set to any value. Set the address of the SCIk.RDR register in the SAR register and set the start address of the RAM area which stores the data in the DAR register.
When the MRA.WBDIS bit is set to 1 (Does not write back the transfer information), the values of registers CRA and CRB are ignored.

(2) DTC Vector Table Setting

Set the start address of the transfer information for the corresponding receive data full interrupt (RXI) in the DTC vector table.

(3) DTC Index Table Setting

Set the start address of the transfer information for each sequence in the DTC index table.

(4) ICU Setting and DTC Module Activation

Set the corresponding ICU.DTCERn.DTCE bit to 1 and the ICU.IERm.IENj bit to 1. Set the DTCST.DTCST bit to 1.

(5) SCI Setting

Set the SCIk.SCR.RIE bit to 1 to enable the RXI interrupt. If a reception error occurs during the SCI receive operation, subsequent receptions are not performed. Accordingly, make settings so that the CPU can accept receive error interrupts.

(6) Start of the Sequence Transfer

On completion of reception of 1-byte data by the SCI, an RXI interrupt is generated to start the DTC. The DTC transfers the received data from the SCIk.RDR register to the RAM. The DTC looks up the DTC index table by using the value from the received data (sequence number) and continues to transfer data corresponding to the that number.
When the CPUSEL bit in the DTC index is 1, the DTC does not read the transfer information and sets the ICU.DTCERn.DTCE bit to 0. Then the DTC outputs an interrupt request to the CPU and ends the sequence transfer.

(7) During Suspension of the Sequence Transfer

Set the ICU.DTCERn.DTCE bit to 1 if the bit is 0. The DTC continues to transfer the data for every generation of the DTC transfer request in response to the corresponding RXI interrupt.

(8) End of the Sequence Transfer

Set the MRB.SQEND bit in the last transfer information of the sequence transfer to 1. After execution of this data transfer, the DTC ends the sequence transfer. The DTC starts to refer to the DTC vector table when a DTC transfer request is generated due to the next corresponding RXI interrupt.

18.7 Interrupt Source

When the DTC has finished data transfer of specified count or when data transfer with the MRB.DISEL bit set to 1 (an interrupt request to the CPU is generated each time the data transfer is performed) has been completed, an interrupt to the CPU is generated by the DTC trigger source. Such interrupts to the CPU are controlled according to the PSW.I bit (interrupt enable) of the CPU, the PSW.IPL[3:0] bits (processor interrupt priority level), and the priority level of the interrupt controller.

18.8 Event Link

The DTC outputs an event signal on completing data transfer in response to one request.

18.9 Low Power Consumption Function

Before making a transition to the module stop state, all-module clock stop mode, or software standby mode, set the DTCST.DTCST bit to 0 (DTC module stop), and then perform the following.

(1) Module Stop Function

Writing 1 (transition to the module-stop state is made) to the MSTPCRA.MSTPA28 bit enables the module stop function of the DTC. If data transfer is in progress at the time 1 is written to the MSTPCRA.MSTPA28 bit, the transition to the module stop state proceeds after data transfer has ended. While the MSTPCRA.MSTPA28 bit is 1, accessing the DTC registers is prohibited.
Writing 0 (release from the module-stop state) to the MSTPCRA.MSTPA28 bit releases the DTC from the module stop state.

(2) All-Module Clock Stop Mode

Make settings according to the procedure under section 11.5.2.1, Transition to All-Module Clock Stop Mode, in section 11, Low Power Consumption.
If any data transfer is in progress at the time the WAIT instruction is executed, the transition to all-module clock stop mode follows the completion of the data transfer.
The DTC is released from the module stop state by writing 0 to the MSTPCRA.MSTPA28 bit following recovery from all-module clock stop mode.

(3) Software Standby Mode

Make settings according to the procedure under section 11.5.3.1, Transition to Software Standby Mode, in section 11, Low Power Consumption.
If any data transfer is in progress at the time the WAIT instruction is executed, the transition to software standby mode follows the completion of the data transfer.

(4) Notes on Low Power Consumption Function

For the WAIT instruction and the register setting procedure, refer to section 11.6.5, Timing of WAIT Instruction in section 11, Low Power Consumption.
To perform data transfer after returning from a low power consumption mode, set the DTCST.DTCST bit to 1 again.
To use a request that is generated in all-module clock stop mode or software standby mode as an interrupt request to the CPU but not as a DTC transfer request, specify the CPU as the interrupt request destination according to the description in section 14.7.3.1, Interrupt Request Destination Setting Procedure in section 14, Interrupt Controller (ICUG), and then execute the WAIT instruction.

18.10 Usage Notes

18.10.1 Start Address of Transfer Information

Set multiples of 4 for the start addresses of the transfer information to be specified in the DTC vector table. If any value other than a multiple of 4 is specified, access still proceeds with the lower 2 bits of the address regarded as 00b.

18.10.2 Allocating Transfer Information

Allocate transfer information in the memory area according to the endian of the area as shown in Figure 18.26.
For example, when writing CRA and CRB settings in 16-bit units in big endian, write the CRA setting to the address plus 8h (Ch) and the CRB setting to the address plus Ah (Eh). In little endian, write the CRB setting to the address plus 8h (Ch) and the CRA setting to the address plus Ah (Eh). When writing CRA and CRB settings in 32-bit units, allocate the CRA setting at the MSB side of the 32 bits and the CRB setting at the LSB side, and write the settings to the address plus 8h (Ch), regardless of endian.

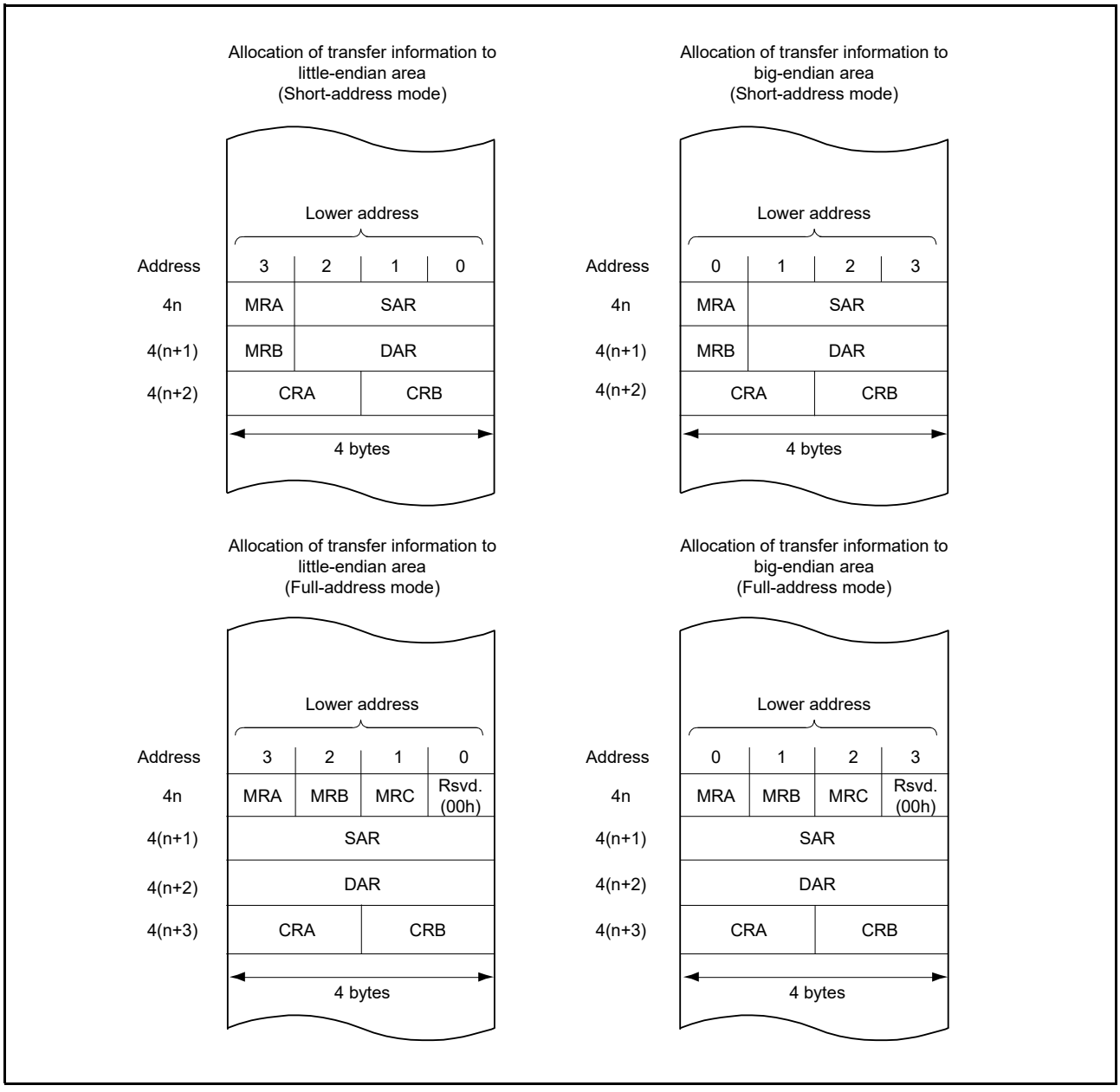


Figure 18.26 Allocation of Transfer Information

18.10.3 Setting the DTC Transfer Request Enable Register in the Interrupt Controller (ICU.DTCERn)

The DMA request should not be issued by setting the DMAC trigger select register (ICU.DMRSRm (m = DMAC channel number)) to the same vector number that has been specified by setting the ICU.DTCERn.DTCE bit to 1 (the corresponding interrupt source is selected as the DTC trigger). For details on the ICU.DTCERn and ICU.DMRSRm registers (m = DMAC channel number), refer to section 14, Interrupt Controller (ICUG).

18.10.4 Notes on Using the Sequence Transfer

When sequence transfer is to be used, make sure that the DTCADMOD.SHORT bit is 0 (full-address mode) and the DTCCR.RRS bit is also 0 (transfer information read is not skipped).
In addition, set the MRB.CHNE bit to 0 (chain transfer is disabled) when setting the MRB.INDX bit to 1 (start sequence transfer and refer the index table) or the MRB.SQEND bit to 1 (end the sequence transfer).

19. Event Link Controller (ELC)

19.1 Overview

The event link controller (ELC) uses the interrupt requests generated by various peripheral modules as event signals, and interconnects (links) peripheral modules. As a result, peripheral modules can directly perform interlinked operation among them without using software.
Event signals can be output regardless of the settings of the corresponding interrupt request enable bits.

Table 19.1 lists the specifications of the ELC, and Figure 19.1 shows a block diagram of the ELC.

Table 19.1 ELC Specifications

Item	Description
Event link function	• 182 types of event signals can be directly interconnected to modules. • Operation for timer modules when inputting an event signal can be selected. • Event linkage operation is possible for port B and port E. Single port*1: Event linkage operation can be set in a single specified port. Port group*1: Event linkage operation can be set by grouping multiple specified ports among total of eight ports.
Low power consumption function	Module stop state can be set.

Note 1. When an input signal to a corresponding pin changes, an event is generated in a single port or in a port group specified as the input.

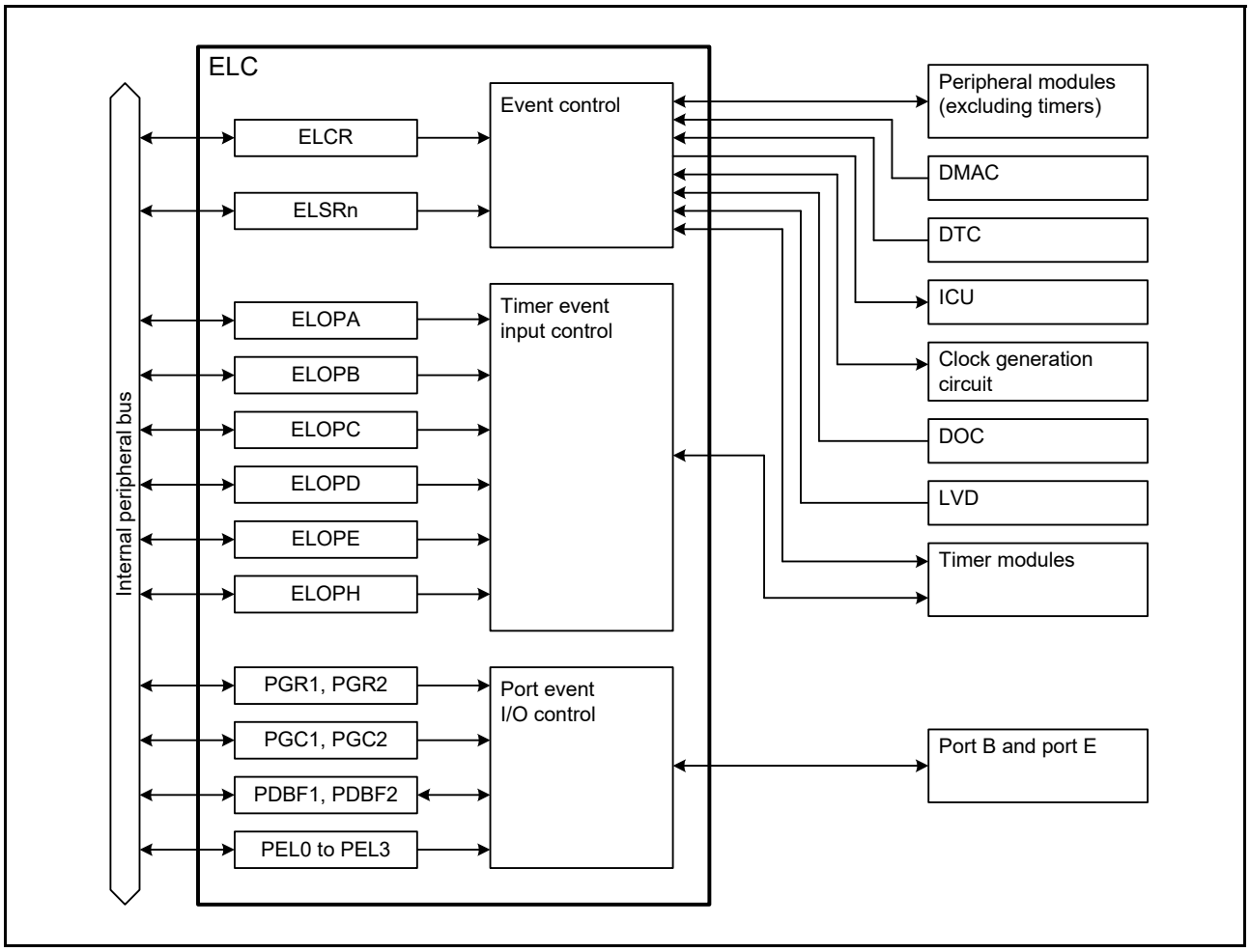


Figure 19.1 ELC Block Diagram (n = 0, 3, 4, 7, 10 to 13, 15, 16, 18 to 28, 30, 31, 33, 45 to 58)

19.2 Register Descriptions

19.2.1 Event Link Control Register (ELCR)

Address(es): ELC.ELCR 0008 B100h

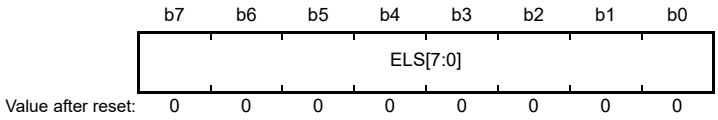
	b7	b6	b5	b4	b3	b2	b1	b0
	ELCON	—	—	—	—	—	—	—
Value after reset:	0	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b7	ELCON	All Event Link Enable	0: ELC function is disabled. 1: ELC function is enabled.	R/W

The ELCR register controls operation of the ELC.

19.2.2 Event Link Setting Register n (ELSRn) (n = 0, 3, 4, 7, 10 to 13, 15, 16, 18 to 28, 30, 31, 33, 45 to 58)

Address(es): ELC.ELSR0 0008 B101h, ELC.ELSR3 0008 B104h, ELC.ELSR4 0008 B105h, ELC.ELSR7 0008 B108h, ELC.ELSR10 0008 B10Bh, ELC.ELSR11 0008 B10Ch, ELC.ELSR12 0008 B10Dh, ELC.ELSR13 0008 B10Eh, ELC.ELSR15 0008 B110h, ELC.ELSR16 0008 B111h, ELC.ELSR18 0008 B113h, ELC.ELSR19 0008 B114h, ELC.ELSR20 0008 B115h, ELC.ELSR21 0008 B116h, ELC.ELSR22 0008 B117h, ELC.ELSR23 0008 B118h, ELC.ELSR24 0008 B119h, ELC.ELSR25 0008 B11Ah, ELC.ELSR26 0008 B11Bh, ELC.ELSR27 0008 B11Ch, ELC.ELSR28 0008 B11Dh, ELC.ELSR30 0008 B12Eh, ELC.ELSR31 0008 B12Fh, ELC.ELSR33 0008 B131h, ELC.ELSR45 0008 B13Dh, ELC.ELSR46 0008 B144h, ELC.ELSR47 0008 B145h, ELC.ELSR48 0008 B146h, ELC.ELSR49 0008 B147h, ELC.ELSR50 0008 B148h, ELC.ELSR51 0008 B149h, ELC.ELSR52 0008 B14Ah, ELC.ELSR53 0008 B14Bh, ELC.ELSR54 0008 B14Ch, ELC.ELSR55 0008 B14Dh, ELC.ELSR56 0008 B14Eh, ELC.ELSR57 0008 B14Fh, ELC.ELSR58 0008 B150h



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	ELS[7:0]	Event Link Select	00h: Event signal output to the corresponding peripheral module is disabled. 01h to FBh: Set the number for the event signal to be linked. Settings other than above are prohibited.	R/W

The ELSRn register specifies an event signal to be linked to for each peripheral module. Table 19.2 shows the correspondence between the ELSRn register and the peripheral modules. Table 19.3 shows the correspondence between values set in the ELSRn register and event signals.

Table 19.2 Correspondence between the ELSRn Register and the Peripheral Modules

Register Name	Peripheral Module
ELSR0	MTU0
ELSR3	MTU3
ELSR4	MTU4
ELSR7	CMT1
ELSR10	TMR0
ELSR11	TMR1
ELSR12	TMR2
ELSR13	TMR3
ELSR15	S12AD (ELCTRG00N)
ELSR16	DA0
ELSR18	ICU (Interrupt 1)*1
ELSR19	ICU (Interrupt 2)*1
ELSR20	Output port group 1
ELSR21	Output port group 2
ELSR22	Input port group 1
ELSR23	Input port group 2
ELSR24	Single port 0
ELSR25	Single port 1
ELSR26	Single port 2
ELSR27	Single port 3
ELSR28	Clock source switching to LOCO
ELSR30	MTU6
ELSR31	MTU7
ELSR33	CMTW0
ELSR45	S12AD1 (ELCTRG10N)
ELSR46	S12AD2 (ELCTRG20N)
ELSR47	MTU9
ELSR48	GPTW event source A (common to all channels)
ELSR49	GPTW event source B (common to all channels)
ELSR50	GPTW event source C (common to all channels)
ELSR51	GPTW event source D (common to all channels)
ELSR52	GPTW event source E (common to all channels)
ELSR53	GPTW event source F (common to all channels)
ELSR54	GPTW event source G (common to all channels)
ELSR55	GPTW event source H (common to all channels)
ELSR56	S12AD (ELCTRG01N)
ELSR57	S12AD1 (ELCTRG11N)
ELSR58	S12AD2 (ELCTRG21N)

Note 1. Specify an event number from among EAh to F1h. Do not set other values.

Table 19.3 Correspondence between Values Set in ELSRn.ELS[7:0] Bits and Event Signals (1/5)

ELS[7:0] Bit Value	Peripheral Modules	Event Signal Set in ELSRn
01h	Multifunction timer pulse unit 3	MTU0 compare match 0A
02h		MTU0 compare match 0B
03h		MTU0 compare match 0C
04h		MTU0 compare match 0D
05h		MTU0 compare match 0E
06h		MTU0 compare match 0F
07h		MTU0 overflow
10h		MTU3 compare match 3A
11h		MTU3 compare match 3B
12h		MTU3 compare match 3C
13h		MTU3 compare match 3D
14h		MTU3 overflow
15h		MTU4 compare match 4A
16h		MTU4 compare match 4B
17h		MTU4 compare match 4C
18h		MTU4 compare match 4D
19h		MTU4 overflow
1Ah		MTU4 underflow
1Eh		MTU6 compare match 6A
1Fh		MTU6 compare match 6B
20h		MTU6 compare match 6C
21h		MTU6 compare match 6D
22h		MTU6 overflow
23h		MTU7 compare match 7A
24h		MTU7 compare match 7B
25h		MTU7 compare match 7C
26h		MTU7 compare match 7D
27h		MTU7 overflow
28h		MTU7 underflow
2Fh		MTU9 compare match 9A
30h		MTU9 compare match 9B
31h		MTU9 compare match 9C
32h		MTU9 compare match 9D
33h		MTU9 compare match 9E
34h		MTU9 compare match 9F
35h		MTU9 overflow
37h	Compare match timer	CMT1 compare match 1
3Ah	Compare match timer W	CMTW0 compare match

Table 19.3 Correspondence between Values Set in ELSRn.ELS[7:0] Bits and Event Signals (2/5)

ELS[7:0] Bit Value	Peripheral Modules	Event Signal Set in ELSRn
3Ch	8-bit timers	TMR0 compare match A0
3Dh		TMR0 compare match B0
3Eh		TMR0 overflow
3Fh		TMR1 compare match A1
40h		TMR1 compare match B1
41h		TMR1 overflow
42h		TMR2 compare match A2
43h		TMR2 compare match B2
44h		TMR2 overflow
45h		TMR3 compare match A3
46h		TMR3 compare match B3
47h		TMR3 overflow
48h	General PWM timer	GPTW0 compare match A
49h		GPTW0 compare match B
4Ah		GPTW0 compare match C
4Bh		GPTW0 compare match D
4Ch		GPTW0 compare match E
4Dh		GPTW0 compare match F
4Eh		GPTW0 overflow
4Fh		GPTW0 underflow
50h		GPTW0 A/D converter start request A
51h		GPTW0 A/D converter start request B
52h		GPTW1 compare match A
53h		GPTW1 compare match B
54h		GPTW1 compare match C
55h		GPTW1 compare match D
56h		GPTW1 compare match E
57h		GPTW1 compare match F
58h		GPTW1 overflow
59h		GPTW1 underflow
5Ah		GPTW1 A/D converter start request A
5Bh		GPTW1 A/D converter start request B
5Ch		GPTW2 compare match A
5Dh		GPTW2 compare match B
5Eh		GPTW2 compare match C
5Fh		GPTW2 compare match D
60h		GPTW2 compare match E
61h		GPTW2 compare match F
62h		GPTW2 overflow
63h		GPTW2 underflow
64h		GPTW2 A/D converter start request A
65h		GPTW2 A/D converter start request B

Table 19.3 Correspondence between Values Set in ELSRn.ELS[7:0] Bits and Event Signals (3/5)

ELS[7:0] Bit Value	Peripheral Modules	Event Signal Set in ELSRn
66h	General PWM timer	GPTW3 compare match A
67h		GPTW3 compare match B
68h		GPTW3 compare match C
69h		GPTW3 compare match D
6Ah		GPTW3 compare match E
6Bh		GPTW3 compare match F
6Ch		GPTW3 overflow
6Dh		GPTW3 underflow
6Eh		GPTW3 A/D converter start request A
6Fh		GPTW3 A/D converter start request B
70h		GPTW4 compare match A
71h		GPTW4 compare match B
72h		GPTW4 compare match C
73h		GPTW4 compare match D
74h		GPTW4 compare match E
75h		GPTW4 compare match F
76h		GPTW4 overflow
77h		GPTW4 underflow
78h		GPTW4 A/D converter start request A
79h		GPTW4 A/D converter start request B
7Ah		GPTW5 compare match A
7Bh		GPTW5 compare match B
7Ch		GPTW5 compare match C
7Dh		GPTW5 compare match D
7Eh		GPTW5 compare match E
7Fh		GPTW5 compare match F
80h		GPTW5 overflow
81h		GPTW5 underflow
82h		GPTW5 A/D converter start request A
83h		GPTW5 A/D converter start request B
84h		GPTW6 compare match A
85h		GPTW6 compare match B
86h		GPTW6 compare match C
87h		GPTW6 compare match D
88h		GPTW6 compare match E
89h		GPTW6 compare match F
8Ah		GPTW6 overflow
8Bh		GPTW6 underflow
8Ch		GPTW6 A/D converter start request A
8Dh		GPTW6 A/D converter start request B

Table 19.3 Correspondence between Values Set in ELSRn.ELS[7:0] Bits and Event Signals (4/5)

ELS[7:0] Bit Value	Peripheral Modules	Event Signal Set in ELSRn
8Eh	General PWM timer	GPTW7 compare match A
8Fh		GPTW7 compare match B
90h		GPTW7 compare match C
91h		GPTW7 compare match D
92h		GPTW7 compare match E
93h		GPTW7 compare match F
94h		GPTW7 overflow
95h		GPTW7 underflow
96h		GPTW7 A/D converter start request A
97h		GPTW7 A/D converter start request B
AFh	Independent watchdog timer	IWDT underflow or refresh error
B8h	Serial communications interfaces	SCI5 error (receive error or error signal detection)
B9h		SCI5 receive data full
BAh		SCI5 transmit data empty
BBh		SCI5 transmit end
C8h	Serial communication interface	RSCI11 error
C9h		RSCI11 receive data full
CAh		RSCI11 transmit data empty
CBh		RSCI11 transmit end
CCh	I^2C-bus interface	RIIC0 communication error or event generation
CDh		RIIC0 receive data full
CEh		RIIC0 transmit data empty
CFh		RIIC0 transmit end
D0h	Serial peripheral interface	RSPI0 error (mode fault, overrun, underrun, or parity error)
D1h		RSPI0 idle
D2h		RSPI0 receive buffer full
D3h		RSPI0 transmit buffer empty
D4h		RSPI0 communication end
D6h	12-bit A/D converter	S12AD A/D conversion end
D8h		S12AD1 A/D conversion end
DAh		S12AD2 A/D conversion end
DCh	Comparator C	Comparison result change of comparator C0
DDh		Comparison result change of comparator C1
DEh		Comparison result change of comparator C2
DFh		Comparison result change of comparator C3
E0h		Comparison result change of comparator C4
E1h		Comparison result change of comparator C5
E2h	Voltage detection circuit	LVD1 voltage detection
E3h		LVD2 voltage detection
E4h	DMA controller	DMAC0 transfer end
E5h		DMAC1 transfer end
E6h		DMAC2 transfer end
E7h		DMAC3 transfer end
E8h	Data transfer controller	DTC transfer end
E9h	Clock generation circuit	Oscillation stop detection of clock generation circuit

Table 19.3 Correspondence between Values Set in ELSRn.ELS[7:0] Bits and Event Signals (5/5)

ELS[7:0] Bit Value	Peripheral Modules	Event Signal Set in ELSRn
EAh	I/O ports	Input edge detection of input port group 1
EBh		Input edge detection of input port group 2
ECh		Input edge detection of single input port 0
EDh		Input edge detection of single input port 1
EEh		Input edge detection of single input port 2
EFh		Input edge detection of single input port 3
F0h	Event link controller	Software event
F1h	Data operation circuit	DOC data operation condition met
F2h	I3C-bus interface	RI3C0 communication error or event generation
F3h		RI3C0 receive data full
F4h		RI3C0 transmit data empty
F6h	Serial peripheral interface	RSPIA0 error
F7h		RSPIA0 idle
F8h		RSPIA0 receive buffer full
F9h		RSPIA0 transmit buffer empty
FAh		RSPIA0 communication end
FBh	General PWM timer	GPTW (OPS) U-/V-/W-phase input edge detected
Settings other than above are prohibited.		

19.2.3 Event Link Option Setting Register A (ELOPA)

Address(es): ELC.ELOPA 0008 B11Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	MTU3MD[1:0]		—	—	—	—	MTU0MD[1:0]	
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	MTU0MD[1:0]	MTU0 Operation Select	b1 b0 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Input capture*1 1 1: Event output is disabled.	R/W
b5 to b2	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b7, b6	MTU3MD[1:0]	MTU3 Operation Select	b7 b6 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Input capture*2 1 1: Event output is disabled.	R/W

Note 1. The MTU0.TCNT value is captured into the MTU0.TGRA register.
Note 2. The MTU3.TCNT value is captured into the MTU3.TGRA register.

The ELOPA register specifies the operations of MTU0 and MTU3 when an event signal is input. Set 11b (event output is disabled) when the ELC function is not used.

19.2.4 Event Link Option Setting Register B (ELOPB)

Address(es): ELC.ELOPB 0008 B120h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	MTU4MD[1:0]	
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	MTU4MD[1:0]	MTU4 Operation Select	b1 b0 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Input capture*1 1 1: Event output is disabled.	R/W
b7 to b2	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

Note 1. The MTU4.TCNT value is captured into the MTU4.TGRA register.

The ELOPB register specifies the operation of MTU4 when an event signal is input. Set 11b (event output is disabled) when the ELC function is not used.

19.2.5 Event Link Option Setting Register C (ELOPC)

Address(es): ELC.ELOPC 0008 B121h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	CMT1MD[1:0]		—	—
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b3, b2	CMT1MD[1:0]	CMT1 Operation Select	b3 b2 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Event counter 1 1: Event output is disabled.	R/W
b7 to b4	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

The ELOPC register specifies the operation of CMT1 when an event signal is input. Set 11b (event output is disabled) when the ELC function is not used.

19.2.6 Event Link Option Setting Register D (ELOPD)

Address(es): ELC.ELOPD 0008 B122h

	b7	b6	b5	b4	b3	b2	b1	b0
	TMR3MD[1:0]		TMR2MD[1:0]		TMR1MD[1:0]		TMR0MD[1:0]	
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TMR0MD[1:0]	TMR0 Operation Select	b1 b0 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Event counter 1 1: Event output is disabled.	R/W
b3, b2	TMR1MD[1:0]	TMR1 Operation Select	b3 b2 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Event counter 1 1: Event output is disabled.	R/W
b5, b4	TMR2MD[1:0]	TMR2 Operation Select	b5 b4 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Event counter 1 1: Event output is disabled.	R/W
b7, b6	TMR3MD[1:0]	TMR3 Operation Select	b7 b6 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Event counter 1 1: Event output is disabled.	R/W

The ELOPD register specifies the operations of TMR0 to TMR3 when an event signal is input. Set 11b (event output is disabled) when the ELC function is not used.

19.2.7 Event Link Option Setting Register E (ELOPE)

Address(es): ELC.ELOPE 0008 B13Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	MTU9MD[1:0]		—	—	MTU7MD[1:0]		MTU6MD[1:0]	
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	MTU6MD[1:0]	MTU6 Operation Select	b1 b0 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Input capture*1 1 1: Event output is disabled.	R/W
b3, b2	MTU7MD[1:0]	MTU7 Operation Select	b3 b2 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Input capture*2 1 1: Event output is disabled.	R/W
b5, b4	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b7, b6	MTU9MD[1:0]	MTU9 Operation Select	b7 b6 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Input capture*3 1 1: Event output is disabled.	R/W

Note 1. The MTU6.TCNT value is captured into the MTU6.TGRA register.
Note 2. The MTU7.TCNT value is captured into the MTU7.TGRA register.
Note 3. The MTU9.TCNT value is captured into the MTU9.TGRA register.

The ELOPE register specifies the operations of MTU6, MTU7, and MTU9 when an event signal is input. Set 11b (event output is disabled) when the ELC function is not used.

19.2.8 Event Link Option Setting Register H (ELOPH)

Address(es): ELC.ELOPH 0008 B141h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	CMTW0MD[1:0]	
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CMTW0MD[1:0]	CMTW0 Operation Select	b1 b0 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Event counter 1 1: Event output is disabled.	R/W
b7 to b2	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

The ELOPH register specifies the operation of CMTW0 when an event signal is input. Set 11b (event output is disabled) when the ELC function is not used.

19.2.9 Port Group Setting Register n (PGRn) (n = 1, 2)

Address(es): ELC.PGR1 0008 B123h, ELC.PGR2 0008 B124h

	b7	b6	b5	b4	b3	b2	b1	b0
	PGR7	PGR6	PGR5	PGR4	PGR3	PGR2	PGR1	PGR0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PGR0	Port Group Setting 0	0: Does not specify the port as a member of the port group. 1: Specifies the port as a member of the port group.	R/W
b1	PGR1	Port Group Setting 1		R/W
b2	PGR2	Port Group Setting 2		R/W
b3	PGR3	Port Group Setting 3		R/W
b4	PGR4	Port Group Setting 4		R/W
b5	PGR5	Port Group Setting 5		R/W
b6	PGR6	Port Group Setting 6		R/W
b7	PGR7	Port Group Setting 7		R/W

The PGRn register specifies a group of I/O ports. Among the ports, ports corresponding to bits set to 1 in the register are selected for a port group.

For example, when the PGR6 and PGR3 bits in the PGR1 register are set to 1, the PB6 and PB3 pins are selected to a port group.

Table 19.4 shows the PGRn register and corresponding ports.

Table 19.4 Registers Related to Port Groups and Corresponding Port Numbers

Port Number	Port Group Setting Register (PGRn)	Port Group Control Register (PGCn)	Port Buffer Register (PDBFn)
Port B	PGR1 register	PGC1 register	PDBF1 register
Port E	PGR2 register	PGC2 register	PDBF2 register

19.2.10 Port Group Control Register n (PGCn) (n = 1, 2)

Address(es): ELC.PGC1 0008 B125h, ELC.PGC2 0008 B126h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	PGCO[2:0]			—	PGCOVE	PGCI[1:0]	
Value after reset:	1	0	0	0	1	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	PGCI[1:0]	Event Output Edge Select	b1 b0 0 0: Event signal is output upon detection of the rising edge of the input signal to the port. 0 1: Event signal is output upon detection of the falling edge of the input signal to the port. 1 x: Event signal is output upon detection of both the rising and falling edges of the input signal to the port.	R/W
b2	PGCOVE	PDBF Overwrite	0: Overwriting the PDBFn register is disabled. 1: Overwriting the PDBFn register is enabled.	R/W
b3	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b6 to b4	PGCO[2:0]	Port Group Operation Select	b6 b4 0 0 0: Low is output when an event signal is input. 0 0 1: High is output when an event signal is input. 0 1 0: The output is toggled (inverted) when an event signal is input. 0 1 1: The buffer value is output when an event signal is input. 1 x x: The output data is rotated (from MSB to LSB) in the port group when an event signal is input.	R/W
b7	—	Reserved	This bit is read as 1. The write value should be 1.	R/W

x: Don’t care

For the port group set as an output, the PGCn register specifies the form of outputting the signal from the port when an event signal is input. For the port group set as an input, the PGCn register enables/disables overwriting of the PDBFn register and specifies the conditions of event generation (edge of the input signal).
Specify the I/O direction of the port by the corresponding bit in the PDR register.
Refer to Table 19.4 for the PGCn register and corresponding ports.

19.2.11 Port Buffer Register n (PDBFn) (n = 1, 2)

Address(es): ELC.PDBF1 0008 B127h, ELC.PDBF2 0008 B128h

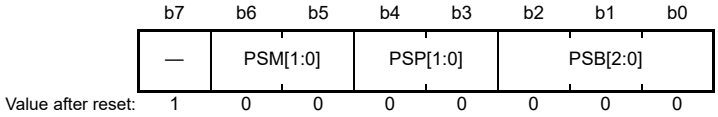
	b7	b6	b5	b4	b3	b2	b1	b0
	PDBF7	PDBF6	PDBF5	PDBF4	PDBF3	PDBF2	PDBF1	PDBF0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PDBF0	Port Buffer 0	Specify the data to be transferred to the PODR register when an event signal is input. The setting value is valid when the PGCn.PGCO[2:0] bits are 011b or 1xxb. Write access to the bit specified as a member of the input port group is disabled. For details, refer to section 19.3, Operation.	R/W
b1	PDBF1	Port Buffer 1		R/W
b2	PDBF2	Port Buffer 2		R/W
b3	PDBF3	Port Buffer 3		R/W
b4	PDBF4	Port Buffer 4		R/W
b5	PDBF5	Port Buffer 5		R/W
b6	PDBF6	Port Buffer 6		R/W
b7	PDBF7	Port Buffer 7		R/W

The PDBFn register is an 8-bit readable/writable register used in combination with the PGRn register. Refer to section 19.3.6, I/O Port Operation When Event Signal is Input and Event Generation for the PDBFn register operations. Refer to Table 19.4 for the PDBFn register and corresponding ports.

19.2.12 Event Link Port Setting Register m (PELm) (m = 0 to 3)

Address(es): ELC.PEL0 0008 B129h, ELC.PEL1 0008 B12Ah, ELC.PEL2 0008 B12Bh, ELC.PEL3 0008 B12Ch



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	PSB[2:0]	Bit Number Specification	Set a bit number for a port to be specified as a single port.	R/W
b4, b3	PSP[1:0]	Port Number Specification	b4 b3 0 0: Setting disabled 0 1: Port B (corresponding to PGR1) 1 0: Port E (corresponding to PGR2) 1 1: Setting prohibited	R/W
b6, b5	PSM[1:0]	Event Link Specification	• For the output port, specify the data to be output from the port. b6 b5 0 0: Low is output when an event signal is input. 0 1: High is output when an event signal is input. 1 x: The output is toggled (inverted) when an event signal is input. • For the input port, select the edge on which the event signal is to be output. b6 b5 0 0: Event signal is output upon detection of the rising edge. 0 1: Event signal is output upon detection of the falling edge. 1 x: Event signal is output upon detection of both the rising and falling edges.	R/W
b7	—	Reserved	This bit is read as 1. The write value should be 1.	R/W

x: Don’t care

The PELm register specifies the single port, the operation upon an event signal input, and the conditions of event generation. This MCU can specify a total of four bits in port B and port E to respective single ports.
Specify the I/O direction of the port by the corresponding bit in the PDR register.

19.2.13 Event Link Software Event Generation Register (ELSEGR)

Address(es): ELC.ELSEGR 0008 B12Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	WI	WE	—	—	—	—	—	SEG
Value after reset:	1	0	1	1	1	1	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	SEG	Software Event Generation	0: Normal operation 1: Software event is generated.	W
b5 to b1	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b6	WE	SEG Bit Write Enable	0: Write to SEG bit is disabled. 1: Write to SEG bit is enabled.	R/W
b7	WI	ELSEGR Register Write Disable	0: Write to ELSEGR register is enabled. 1: Write to ELSEGR register is disabled.	W

The MOV instruction must be used to write to this register.

SEG Bit (Software Event Generation)

When 1 is written to this bit while the WE bit is 1, a software event is generated.
This bit is read as 0. Even if 1 is written to this bit, this bit does not become 1.

WE Bit (SEG Bit Write Enable)

The SEG bit can be written to only when the WE bit is 1.
To set this bit to 1, write 0 to the WI bit and write 1 to this bit simultaneously.
To set this bit to 0, write 0 to the WI bit and write 0 to this bit simultaneously.

WI Bit (ELSEGR Register Write Disable)

The ELSEGR register can be written to only when the value to be written to the WI bit is 0.
This bit is read as 1.

19.3 Operation

19.3.1 Relation between Interrupt Handling and Event Linking

The peripheral modules incorporated in the MCU are provided with the interrupt request status flags and the interrupt enable bits to enable/disable these interrupt requests. When an interrupt request is generated in a peripheral module, the corresponding interrupt request status flag becomes 1. If the corresponding interrupt request is enabled then, the interrupt is requested to the CPU.

In contrast, the event link controller (ELC) uses the interrupt requests generated by various peripheral modules as event signals, interconnects (links) peripheral modules, and then, makes peripheral modules perform direct interlinked operation among them without using software. Event signals can be output regardless of the setting of the corresponding interrupt enable bit.

Figure 19.2 shows the relation between the interrupt handling and ELC.

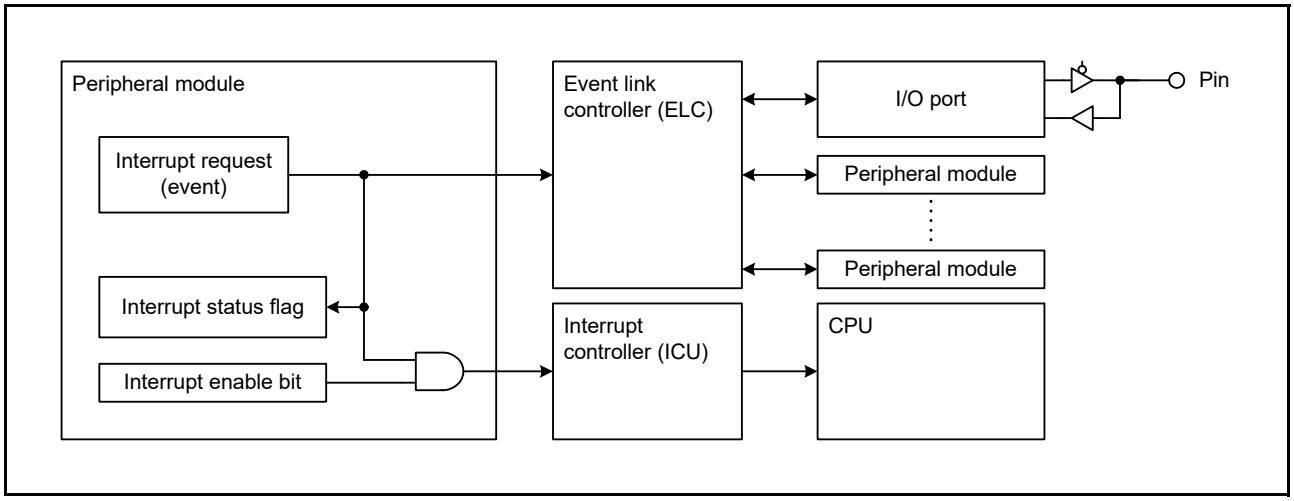


Figure 19.2 Relation between Interrupt Handling and ELC

19.3.2 Event Linkage

When events are specified in the ELSRn registers, the corresponding peripheral modules can be operated at generation of the specified events. A single peripheral module can link only with a single event. Set the ELSRn register after completing the initialization of the peripheral module to operate by an event. Table 19.5 lists the operations of peripheral modules when an event signal is input.

Table 19.5 Operations of Peripheral Modules When Event Signal is Input

Peripheral Module	Operations When Event Signal is Input		
MTU CMT CMTW TMR	The following operations can be selected by setting the ELOPA to ELOPE and ELOPH registers: • Starts counting when an event signal is input. • Restarts counting when an event signal is input. • Counts the input events (CMT, CMTW, TMR). • Performs input-capture operation when an event signal is input (MTU).		
GPTW	The following operations can be selected by setting registers in GPTW: • Starts counting when an event signal is input. • Stops counting when an event signal is input. • Clears counter when an event signal is input. • Increments counter when an event signal is input. • Decrements counter when an event signal is input. • Performs input-capture operation when an event signal is input.		
A/D converter	Starts A/D conversion when an event signal is input.		
D/A converter	Starts D/A conversion when an event signal is input.		
I/O ports (output)	The value of PODR register (port output data register) changes when an event signal is input (The level output from the corresponding pin changes).	Port group	• Changes the PODR register value to the specified value. • Transfers the PDBFn register value to the PODR register (n = 1, 2). • Rotates the PODR register.
		Single port	Changes the PODR register value to the specified value.
I/O ports (input)	When the signal level of the input pin changes	Port group	Generates an event.
		Single port	
	When an event signal is input	Port group	Transfers the signal level of the input pin to the PDBFn register.
		Single port	This combination cannot be used.
Clock generation circuit	Switches the clock source to the low-speed on-chip oscillator when an event signal is input.*1		
Interrupt controller	Request an interrupt to the CPU, starts DMA transfer, or starts DTC transfer when an event signal is input.		

Note 1. The SCKCR3.CKSEL[2:0] bits are modified to 000b (LOCO) regardless of the value of the protect register (PRCR.PRC0).

19.3.3 Operation of Peripheral Timer Modules When Event Signal is Input

For the timer modules except GPTW, set the ELOPA to ELOPE or ELOPH register to specify the operation for when an event signal is input.

(1) Count Start Operation

When an event signal is input, the timer starts counting and the count start bit*1 in each timer control register becomes 1. An event signal that is input while the count start bit is 1 is ignored.

(2) Count Restart Operation

When an event signal is input, the timer counter is cleared. Since the count start bit*1 in each timer control register is retained, counting is restarted when an event signal is input while the count start bit is 1.

(3) Event Counter Operation

Event signal is selected as the timer count source. When an event signal is input, the timer counter is incremented.

(4) Input Capture Operation

When an event signal is input, the timer performs input-capture operation.

Note 1. Refer to the register descriptions on starting the timer in the relevant peripheral timer module section.

19.3.4 Operation of GPTW When Event Signal is Input

Eight event signals specified by the ELSR48 to ELSR55 registers are connected to every channel of GPTW as GPTW event sources A to H. To set the operation of GPTW when inputting an event signal, enable any of the event sources A to H by a corresponding bit in registers listed in Table 19.6.

Table 19.6 Operations When Event Signal is Input and Corresponding Source Select Registers

Operations on Event	Register Symbol	Register Name
Count start	GTSSR	General PWM Timer Start Source Select Register
Count stop	GTPSR	General PWM Timer Stop Source Select Register
Counter clear	GTCSR	General PWM Timer Clear Source Select Register
Count-up	GTUPSR	General PWM Timer Count-Up Source Select Register
Count-down	GTDNSR	General PWM Timer Count-Down Source Select Register
Input capture A	GTICASR	General PWM Timer Input Capture Source Select Register A
Input capture B	GTICBSR	General PWM Timer Input Capture Source Select Register B

19.3.5 Operation of A/D and D/A Converters When Event Signal is Input

When an event signal is input, the ADCSR.ADST bit and the DACR.DAOE0 bit*1 are set to 1 and the A/D and D/A converter start A/D and D/A conversion, respectively.

Note 1. Refer to the bit descriptions in the A/D converter and D/A converter sections.

19.3.6 I/O Port Operation When Event Signal is Input and Event Generation

The I/O port operation at an event signal input and conditions for event generation are set by the registers in ELC. The I/O ports that are used to set an event linkage are port B and port E.

(1) Single Ports and Port Groups

There are two event link modes: event link to single ports and event link to port groups. In the former mode, events can be interconnected to any one of the I/O ports. In the latter mode, events can be interconnected to port groups consisting of any two or more bits in the same I/O ports.

A single port can be set by the PELm.PSP[1:0] and PSB[2:0] bits (m = 0 to 3). A port group can be specified by setting two or more bits in the PGRn register (n = 1, 2) to 1. Among the ports corresponding to the bits set to 1 in the PGRn register, a port set as output becomes an output port group member, and a port set as input becomes an input port group member.

If an I/O port is specified as both a single port and a member of a port group, both functions are enabled when the corresponding port is input, whereas only the port group function is enabled when the corresponding port is output.

Set the PDR register to select the direction of the I/O ports.

(2) Event Generation in Single Input Ports

A single port that is set as input generates an event signal when the input signal to the corresponding pin changes. The event generation condition is specified using the PELm.PSM[1:0] bits (m = 0 to 3). An example of operation is shown in Figure 19.3 (1).

(3) Single Output Ports Operation When Event Signal is Input

When an event signal is input to a single port set as output, the output level (the PODR register value) of the corresponding pin changes as specified by the PELm.PSM[1:0] bits. An example of operation is shown in Figure 19.3 (2).

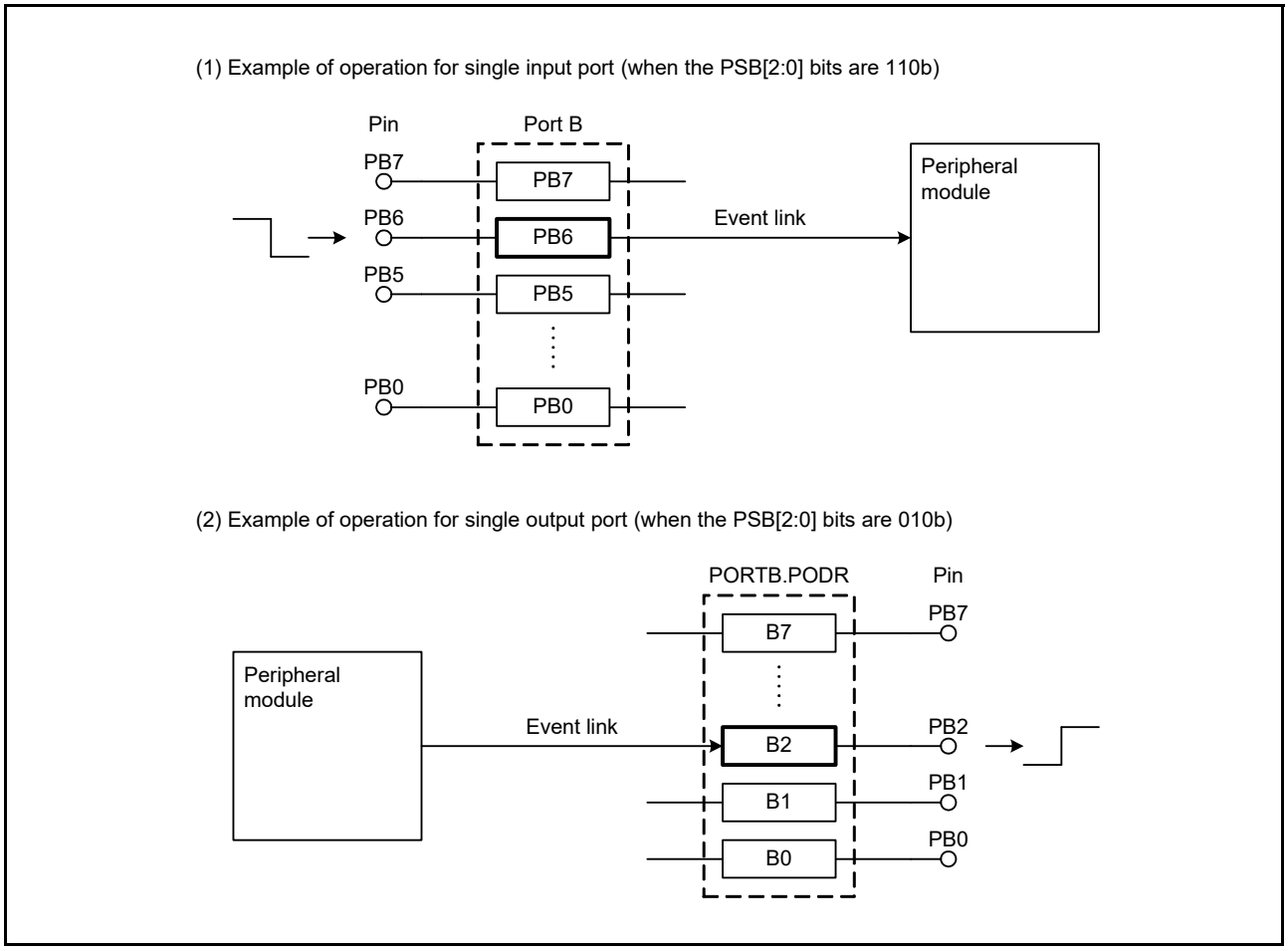


Figure 19.3 Event Linkage Related to Single Ports (Port B)

(4) Event Generation in Input Port Group

An input port group generates an event signal when any of input signals to the corresponding pins change. The event generation condition is specified using the PGCn.PGCI[1:0] bits (n = 1, 2).

(5) Input Port Group Operation When Event Signal is Input

When an event signal is input to an input port group, the level of the corresponding pins is transferred to the PDBFn register. Values of the bits corresponding to ports that are not specified as members of the input port group do not change. An example of operation is shown in Figure 19.4.

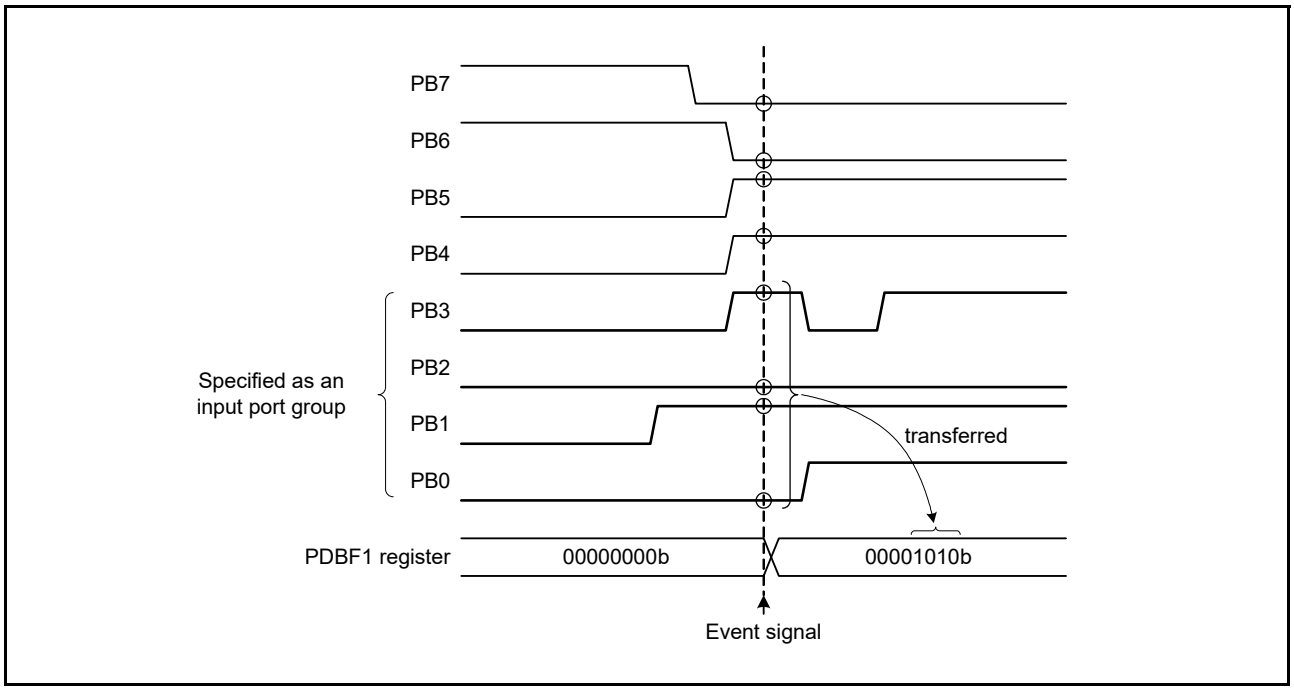


Figure 19.4 Event Linkage Related to Input Port Groups (Port B)

(6) Output Port Group Operation When Event Signal is Input

When an event signal is input to an output port group, the value of the corresponding PODR register changes according to a setting of the PGCn.PGCO[2:0] bits (n = 1, 2). An example of operation is shown in Figure 19.5.

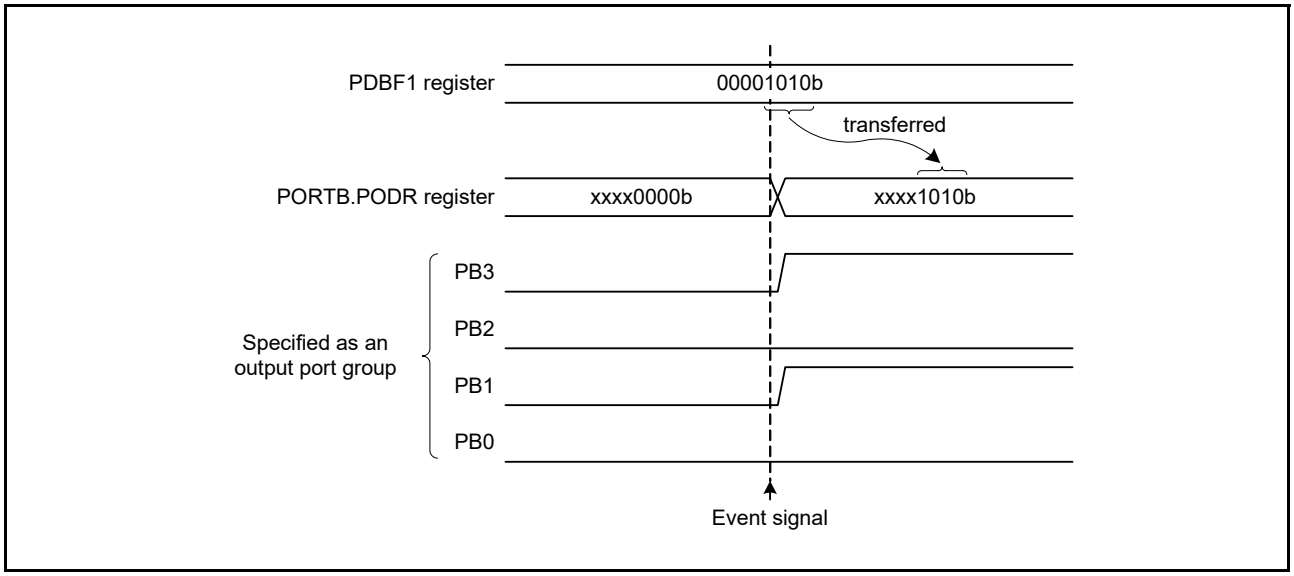


Figure 19.5 Event Linkage Related to Output Port Groups (Port B)

(7) Operation of the PDBFn Registers

(a) Input Port Groups

When an event signal is input to an input port group, the level of the corresponding pins is transferred to the PDBFn register (n = 1, 2). When another event signal is input to the input port group in this condition, different operations are performed depending on the PGCn.PGCOVE bit setting described as below.

- When the PGCn.PGCOVE bit is 0 (overwriting is disabled)
  When the value transferred to the PDBFn register after an input of the last event signal has already been read by the CPU or DTC, the level of the corresponding pins at the time is transferred to the PDBFn register. When the value has not been read, the level of the pins is not transferred to the PDBFn register, and the input event signal is ignored.
- When the PGCn.PGCOVE bit is 1 (overwriting is enabled)
  When another event signal is input to the input port group, the level of the corresponding pins is transferred to the PDBFn register.

(b) Output Port Groups

When an output port group is specified to output the PDBFn register value (PGCn.PGCO[2:0] bits = 011b), the PDBFn register value is transferred to the PODR register following an input of an event signal to the output port group. Data is not transferred to the bits corresponding to the ports that are not specified as members of the output port group.
When output data is specified to rotate in an output port group (PGCn.PGCO[2:0] bits = 1xxb), the data is transferred from the PDBFn register to the PODR register at first event signal, and the PODR register value is rotated from MSB to LSB within the relevant group at second and subsequent signals.
Examples of operation are shown in Figure 19.6.

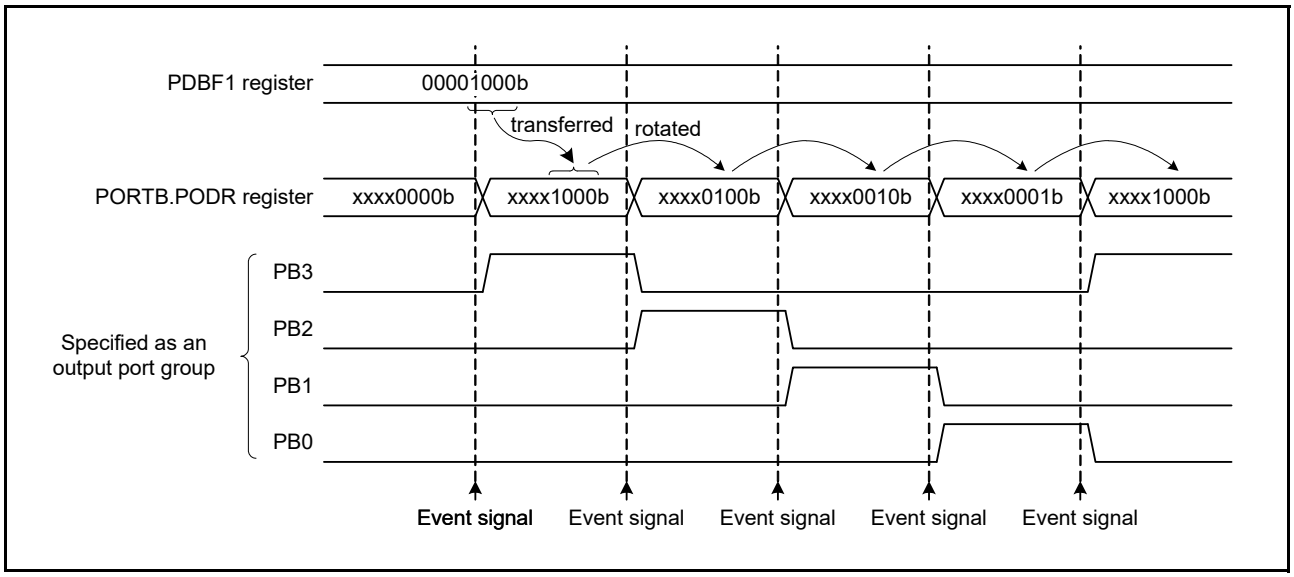


Figure 19.6 Bit-Rotating Operation of Output Port Groups (Port B)

(8) Restrictions on Writing to PODR and PDBF Registers

When the ELCR.ELCON bit is 1 (ELC function is enabled), write access to the PODR and PDBFn registers (n = 1, 2) becomes disabled at the following conditions.

- When a port is specified as a member of the input port group and when the event linkage is set, write access to the corresponding bit in the PDBFn register becomes disabled.
- When a port is specified as a member of the output port group, write access to the corresponding bit in the PODR register becomes disabled.
- When a port is specified as a single output port and when the event linkage for the port is set by the ELSRn register, write access to the corresponding bit in the PODR register becomes disabled.

19.3.7 Example of Procedure for Linking Events

The following describes the procedure for linking events.

(1) Initialize the peripheral module (destination) that operates based on an event signal.
(2) When event linkage is set to a port, set the following registers corresponding to the port.
PODR register: Set the initial values of the output ports.
PDR register: Set the I/O direction of the ports.
PGRn register: To operate ports for a port group, select ports to be specified as port group members (n = 1, 2).
PGCn register: Set the operation of the port group.
PELm register: When a port is operated as a single port, specify the port to be used, an operation of the port at an input of event signal, and the event generation condition (m = 0 to 3).
(3) Set the number of the event signal to the ELSRn register corresponding to the destination peripheral module.
(4) To link an event to a timer module, set any of the ELOPA to ELOPE and ELOPH registers corresponding to the timer as required.
(5) Set the ELCR.ELCON bit to 1, which enables linkage of all the events.
(6) Set the operation of the peripheral module (source) from which an event signal is output, and activate the module. The preset operation of the destination peripheral module is started by the event signal that is output from the source peripheral module.
(7) To stop event linkage of independent peripheral module, set 00h to the ELSRn register corresponding to the peripheral module. To stop linkage of all the events, set the ELCR.ELCON bit to 0.

Note: When using event signal output from the LVD, set the LVD and then the ELC. Set the corresponding ELSRn register to 00h and then disable the LVD.

19.4 Usage Notes

19.4.1 Setting ELSRn Register

(1) Setting ELSR18 and ELSR19 Registers

Specify an event number from among EAh to F1h. Do not set the value other than preceding numbers.

19.4.2 Setting Bit-Rotating Operation of Output Port Groups

When the values of the PDBFn register (n = 1, 2) are changed in the bit-rotating operation mode of the output port group, set the ELSRn register again. Set intervals for generating the event as at least one PCLKB cycle when using it for bit-rotating operation.

19.4.3 Linking DMA/DTC Transfer End Signal as Event

When linking the DMA/DTC transfer end signal as an event signal, do not set the same peripheral module as the DMA/DTC transfer destination and event link destination. If set, the peripheral module might be started before DMA/DTC transfer to the peripheral module is completed.

19.4.4 Clock Settings

To link events, make sure that the ELC and the related peripheral modules are in an operational condition. The peripheral modules cannot operate if they are in the module stop state or in mode which they stop (all-module clock stop mode or software standby mode).

19.4.5 Module Stop Function Setting

ELC operation can be disabled or enabled using module stop control register B (MSTPCRB). After reset is released, the ELC function is disabled. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

20. I/O Ports

20.1 Overview

The pins of an I/O port for this MCU function as general I/O port pins, I/O pins for peripheral modules, or interrupt input pins.

Each pin is also configurable as an I/O pin of a peripheral module or an input pin for an interrupt. All pins function as input pins immediately after a reset, and pin functions are switched by register settings. The setting of each pin is specified by the registers for the corresponding I/O port and peripheral modules.

This MCU has the following registers for each port pin.

- Port direction register (PDR) that selects input or output direction
- Port output data register (PODR) that holds data for output
- Port input register (PIDR) that indicates the pin states
- Open-drain control register y (ODRy, y = 0, 1) that selects the output type of each pin
- Pull-up resistor control register (PCR) that controls on/off of the input pull-up MOS
- Drive capacity control registers (DSCR and DSCR2) that select the drive capacity
- Port mode register (PMR) that specifies the pin function of each port
- General-purpose input/output pin select extension register (GPSEXT) that selects pin functions

For details on PMR, refer to section 21, Multi-Function Pin Controller (MPC).

This MCU also has the following registers for each port.

- Port output retention setting registers (POHSR1 and POHSR2) that control whether or not to retain the data output from the ports
- Port output retention control register (POHCR) that enables or disables retaining of the data output from the ports

By using these registers, simultaneous change of the data output from the specified ports is possible. For details on this functionality, see section 20.4, Retaining the Data Output from Ports.

The configuration of the I/O ports differs depending on the package. Table 20.1 shows the specifications of I/O ports, and Table 20.2 lists the port functions.

Table 20.1 Specifications of I/O Ports

	Package		Package		Package		Package	
Port	100 Pins	Number of Pin	80 Pins	Number of Pin	64 Pins	Number of Pin	48 Pins	Number of Pin
PORT0	P00, P01	2	P00, P01	2	P00, P01	2	P00	1
PORT1	P10, P11	2	P10, P11	2	P11	1	P10, P11	2
PORT2	P20 to P24, P27	6	P20 to P22, P27	4	P20 to P22	3	P20, P21	2
PORT3	P30 to P33, P36, P37	6	P30, P31, P36, P37	4	P36, P37	2	P36, P37	2
PORT4	P40 to P47	8	P40 to P47	8	P40 to P47	8	P40 to P44	5
PORT5	P50 to P55	6	P50 to P55	6	P52 to P54	3	P52, P53	2
PORT6	P60 to P65	6	P60, P64, P65	3	P64, P65	2	P62	1
PORT7	P70 to P76	7	P70 to P76	7	P70 to P76	7	P71 to P76	6
PORT8	P80 to P82	3	—	0	—	0	—	0
PORT9	P90 to P96	7	P90 to P96	7	P90 to P96	7	P91 to P95	5
PORTA	PA0 to PA5	6	PA3, PA5	2	—	0	—	0
PORTB	PB0 to PB7	8	PB0 to PB6	7	PB0 to PB6	7	PB0 to PB6	7
PORTD	PD0 to PD7	8	PD2 to PD7	6	PD3 to PD7	5	PD3, PD5, PD7	3
PORTE	PE0 to PE5	6	PE2 to PE4	3	PE2	1	PE2	1
PORTN	PN6, PN7	2	PN6, PN7	2	PN6, PN7	2	PN6	1
	Total of pins	83	Total of pins	63	Total of pins	50	Total of pins	38

Table 20.2 Port Functions

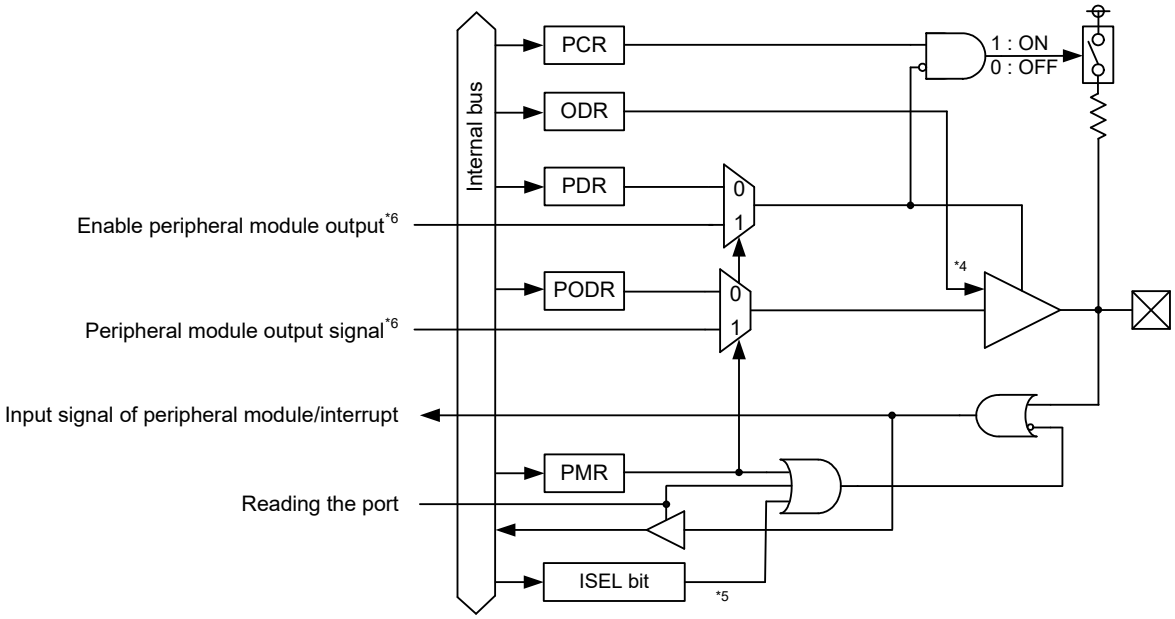
Port	Pin	Input Pull-up	Open-Drain Output	Driving Ability Switching	5-V Tolerant
PORT0	P00, P01	✓	✓	Normal drive/high drive	—
PORT1	P10, P11	✓	✓	Normal drive/high drive	—
PORT2	P20 to P24, P27	✓	✓	Normal drive/high drive	—
PORT3	P30 to P33	✓	✓	Normal drive/high drive	—
	P36, P37	✓	✓	Fixed to normal output	—
PORT4	P40 to P47	✓	✓	Fixed to normal output	—
PORT5	P50 to P55	✓	✓	Fixed to normal output	—
PORT6	P60 to P65	✓	✓	Fixed to normal output	—
PORT7	P70	✓	✓	Normal drive/high drive	—
	P71 to P76	✓	✓	Normal drive/high drive/ large current output	—
PORT8	P80, P82	✓	✓	Normal drive/high drive	—
	P81	✓	✓	Normal drive/high drive/ large current output	—
PORT9	P90 to P95	✓	✓	Normal drive/high drive/ large current output	—
	P96	✓	✓	Normal drive/high drive	—
PORTA	PA0 to PA5	✓	✓	Normal drive/high drive	—
PORTB	PB0, PB3, PB4, PB6, PB7	✓	✓	Normal drive/high drive	—
	PB1, PB2	✓	✓	Fixed to normal output	✓
	PB5	✓	✓	Normal drive/high drive/ large current output	—
PORTD	PD0 to PD2, PD4 to PD7	✓	✓	Normal drive/high drive	—
	PD3	✓	✓	Normal drive/high drive/ large current output	—
PORTE	PE0, PE1, PE3 to PE5	✓	✓	Normal drive/high drive	—
	PE2	—	—	—	—
PORTN	PN6*1	✓	✓	Normal drive/high drive	—
	PN7*2	✓	✓	Normal drive/high drive	—

Note 1. This pin is initially set to input and is pulled up.
Note 2. This pin is initially set to input and is pulled down.

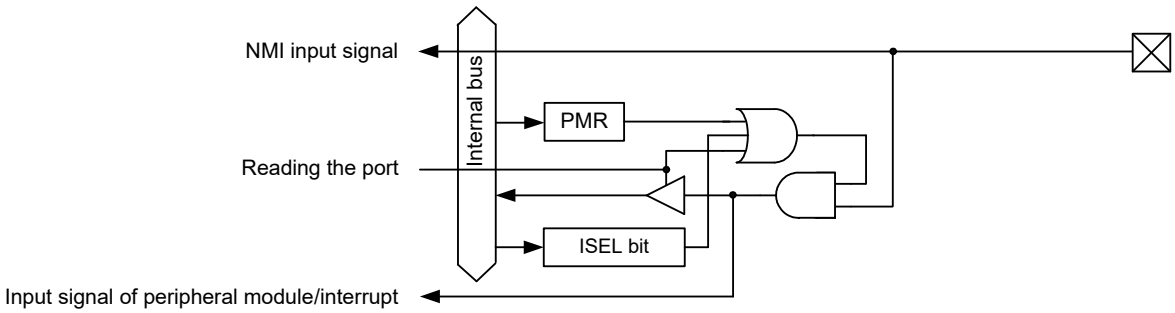
The settings for input pull-up, open-drain output, switching of driving ability, and 5-V tolerance are also applied to other signals multiplexed on the general I/O pins.

20.2 I/O Port Configuration

Port 0: P00, P01*1
Port 1: P10*2, P11
Port 2: P22*1, P23*3, P24*3, P27*1, *2
Port 3: P30*1, *2, P31*1, *2 P32*3, P33*3
Port 7: P70*1
Port 8: P80*3, P82*3
Port 9: P93, P95, P96*1
Port A: PA1*3, PA5*1, *2
Port B: PB0, PB1, PB3, PB4, PB6
Port D: PD4*1, PD5, PD6*1, PD7
Port E: PE0*3, PE1*3, PE3*1, *2, PE4*1, *2, PE5*3
Port N: PN7*1



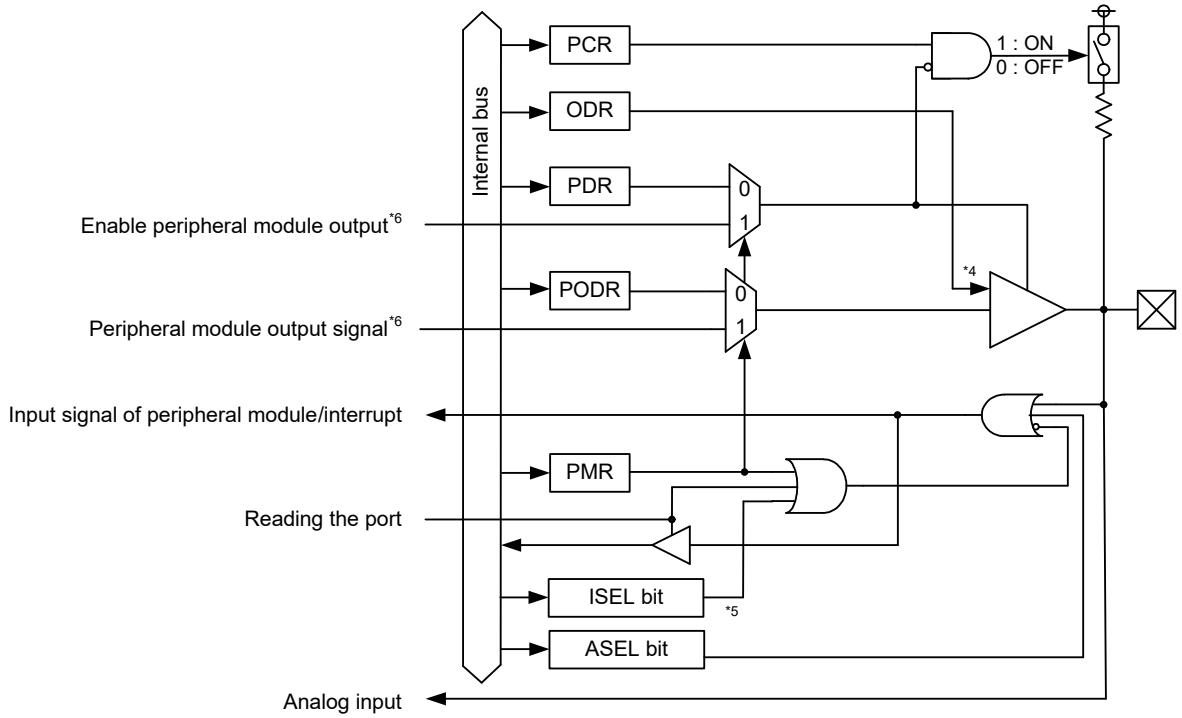
Port E: PE2



Note 1. Not provided on the 48-pin products.
Note 2. Not provided on the 64-pin products.
Note 3. Only 100-pin products have this pin.
Note 4. Control signal for N-channel open-drain output
Note 5. An external interrupt function is multiplexed on this pin.
Note 6. Output from peripheral modules is multiplexed on this pin.

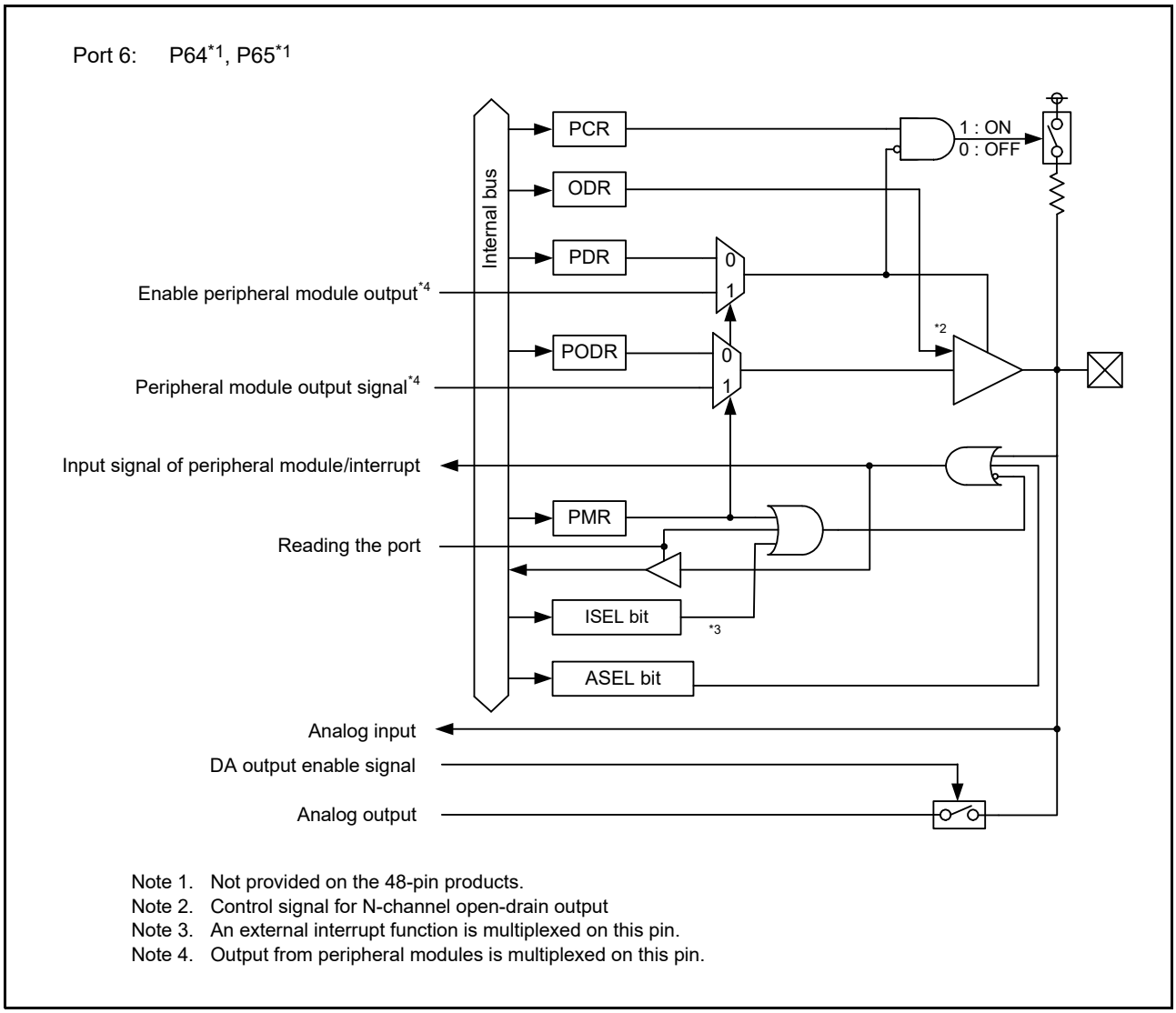
Figure 20.1 I/O Port Configuration (1)

Port 2: P20, P21
Port 5: P52, P53, P54*1, P55*1, *2
Port 6: P60*1, *2, P61*1, *2, *3, P62*2, *3, P63*1, *2, *3



Note 1. Not provided on the 48-pin products.
Note 2. Not provided on the 64-pin products.
Note 3. Not provided on the 80-pin products.
Note 4. Control signal for N-channel open-drain output
Note 5. An external interrupt function is multiplexed on this pin.
Note 6. Output from peripheral modules is multiplexed on this pin.

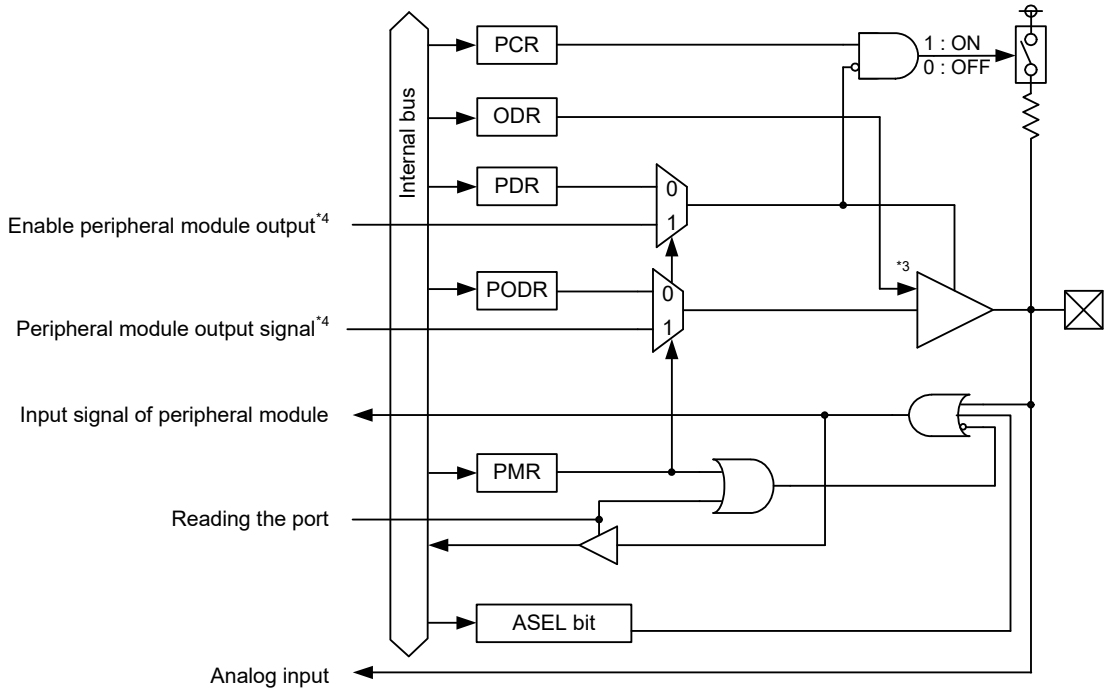
Figure 20.2 I/O Port Configuration (2)



Note 1. Not provided on the 48-pin products.
Note 2. Control signal for N-channel open-drain output
Note 3. An external interrupt function is multiplexed on this pin.
Note 4. Output from peripheral modules is multiplexed on this pin.

Figure 20.3 I/O Port Configuration (3)

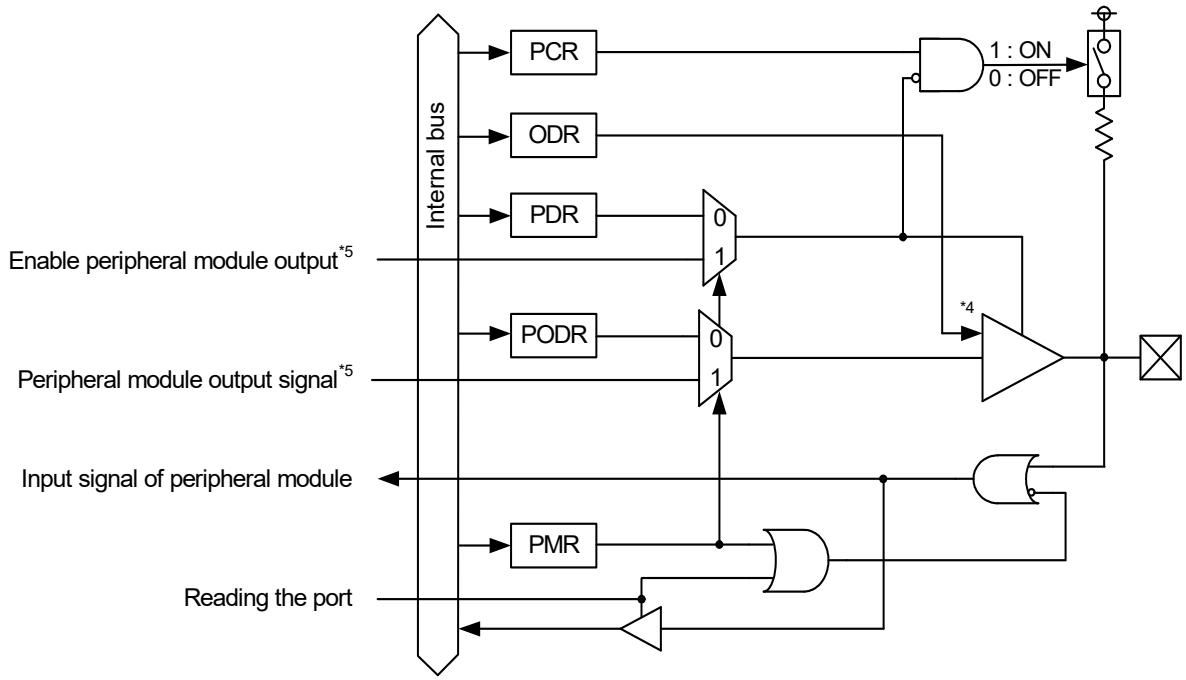
Port 4: P40 to P44, P45*1, P46*1, P47*1
Port 5: P50*1, *2, P51*1, *2



Note 1. Not provided on the 48-pin products.
Note 2. Not provided on the 64-pin products.
Note 3. Control signal for N-channel open-drain output
Note 4. Output from peripheral modules is multiplexed on this pin.

Figure 20.4 I/O Port Configuration (4)

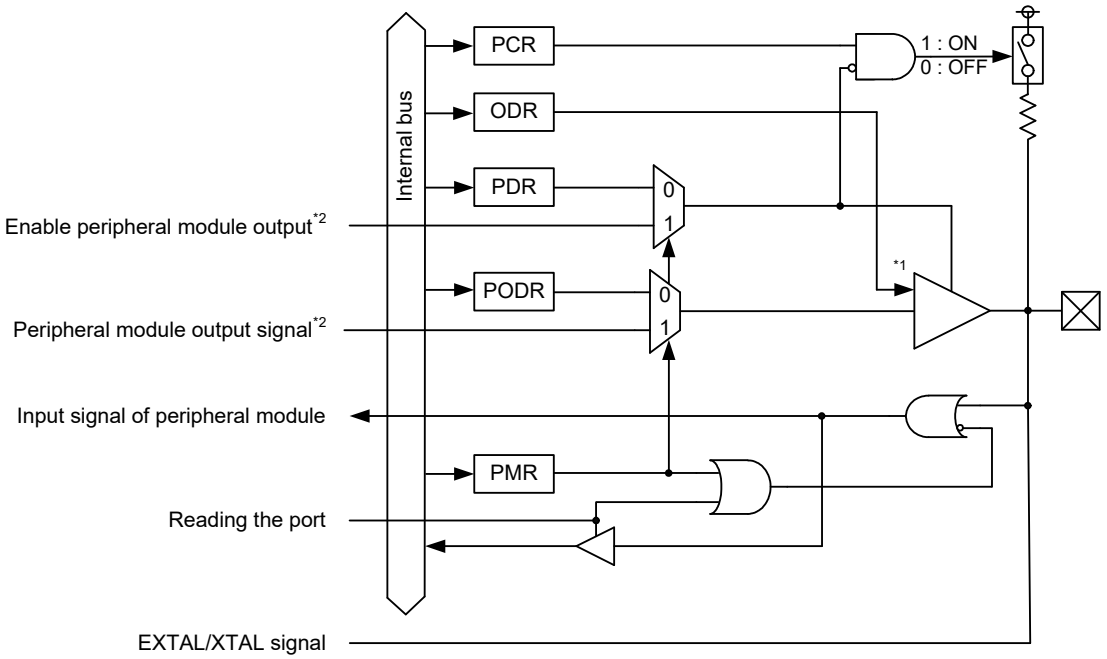
Port 7: P71 to P76
Port 8: P81*1
Port 9: P90*2, P91, P92, P94
Port A: PA0*1, PA2*1, PA3*2, *3, PA4*1
Port B: PB2, PB5, PB7*1
Port D: PD0*1, PD1*1, PD2*2, *3, PD3
Port N: PN6



Note 1. Only 100-pin products have this pin.
Note 2. Not provided on the 48-pin products.
Note 3. Not provided on the 64-pin products.
Note 4. Control signal for N-channel open-drain output
Note 5. Output from peripheral modules is multiplexed on this pin.

Figure 20.5 I/O Port Configuration (5)

Port 3: P36, P37



Note 1. Control signal for N-channel open-drain output
Note 2. Output from peripheral modules is multiplexed on this pin.

Figure 20.6 I/O Port Configuration (6)

20.3 Register Descriptions

20.3.1 Port Direction Register (PDR)

Address(es): PORT0.PDR 0008 C000h, PORT1.PDR 0008 C001h, PORT2.PDR 0008 C002h, PORT3.PDR 0008 C003h, PORT4.PDR 0008 C004h, PORT5.PDR 0008 C005h, PORT6.PDR 0008 C006h, PORT7.PDR 0008 C007h, PORT8.PDR 0008 C008h, PORT9.PDR 0008 C009h, PORTA.PDR 0008 C00Ah, PORTB.PDR 0008 C00Bh, PORTD.PDR 0008 C00Dh, PORTE.PDR 0008 C00Eh, PORTN.PDR 0008 C016h

	b7	b6	b5	b4	b3	b2	b1	b0
	B7	B6	B5	B4	B3	B2	B1	B0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm0 I/O Select	0: Input (Functions as an input pin.) 1: Output (Functions as an output pin.)	R/W
b1	B1	Pm1 I/O Select		R/W
b2	B2	Pm2 I/O Select		R/W
b3	B3	Pm3 I/O Select		R/W
b4	B4	Pm4 I/O Select		R/W
b5	B5	Pm5 I/O Select		R/W
b6	B6	Pm6 I/O Select		R/W
b7	B7	Pm7 I/O Select		R/W

m = 0 to 9, A, B, D, E, N

PDR is a register which is used to select the input or output direction for individual pins of the corresponding port when the pins are configured as the general I/O pins.
Each bit of PORTm.PDR corresponds to each pin of port m; I/O direction can be specified in 1-bit units.
Each bit of PDR corresponding to port m that does not exist is reserved.
Also, the PORTE.PDR.B2 bit corresponding to the PE2 pin is reserved, because the PE2 pin is input only.
Make settings of the reserved bit according to the description in section 20.5.1, Initialization of the Port Direction Register (PDR).

20.3.2 Port Output Data Register (PODR)

Address(es): PORT0.PODR 0008 C020h, PORT1.PODR 0008 C021h, PORT2.PODR 0008 C022h, PORT3.PODR 0008 C023h, PORT4.PODR 0008 C024h, PORT5.PODR 0008 C025h, PORT6.PODR 0008 C026h, PORT7.PODR 0008 C027h, PORT8.PODR 0008 C028h, PORT9.PODR 0008 C029h, PORTA.PODR 0008 C02Ah, PORTB.PODR 0008 C02Bh, PORTD.PODR 0008 C02Dh, PORTE.PODR 0008 C02Eh, PORTN.PODR 0008 C036h

	b7	b6	b5	b4	b3	b2	b1	b0
	B7	B6	B5	B4	B3	B2	B1	B0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm0 Output Data Store	0: Low output 1: High output	R/W
b1	B1	Pm1 Output Data Store		R/W
b2	B2	Pm2 Output Data Store		R/W
b3	B3	Pm3 Output Data Store		R/W
b4	B4	Pm4 Output Data Store		R/W
b5	B5	Pm5 Output Data Store		R/W
b6	B6	Pm6 Output Data Store		R/W
b7	B7	Pm7 Output Data Store		R/W

m = 0 to 9, A, B, D, E, N

PODR is a register which holds the data to be output from the pins used for general I/O.
Bits that correspond to port m that does not exist is reserved. Write 0 (low output) to these bits.
The PORTE.PODR.B2 bit corresponding to the PE2 pin is reserved, because the PE2 pin is input only. Writing to this bit has no effect.
The bit corresponding to a pin that does not exist is reserved. A reserved bit is always read as 0. The write value should always be 0.

20.3.3 Port Input Register (PIDR)

Address(es): PORT0.PIDR 0008 C040h, PORT1.PIDR 0008 C041h, PORT2.PIDR 0008 C042h, PORT3.PIDR 0008 C043h, PORT4.PIDR 0008 C044h, PORT5.PIDR 0008 C045h, PORT6.PIDR 0008 C046h, PORT7.PIDR 0008 C047h, PORT8.PIDR 0008 C048h, PORT9.PIDR 0008 C049h, PORTA.PIDR 0008 C04Ah, PORTB.PIDR 0008 C04Bh, PORTD.PIDR 0008 C04Dh, PORTE.PIDR 0008 C04Eh, PORTN.PIDR 0008 C056h

	b7	b6	b5	b4	b3	b2	b1	b0
	B7	B6	B5	B4	B3	B2	B1	B0
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm0	0: Low input 1: High input	R
b1	B1	Pm1		R
b2	B2	Pm2		R
b3	B3	Pm3		R
b4	B4	Pm4		R
b5	B5	Pm5		R
b6	B6	Pm6		R
b7	B7	Pm7		R

m = 0 to 9, A, B, D, E, N

PIDR is a register which reflects individual pin states of the port.

The pin states of port m can be read with the PORTm.PIDR, regardless of the values of PORTm.PDR and PORTm.PMR.

The states of pins cannot be read when the PmnPFS.ASEL bit is set to 1.

The NMI pin state is reflected in the PE2 bit.

The bit corresponding to a pin that does not exist is reserved. A reserved bit is read as undefined, and cannot be modified.

20.3.4 Port Mode Register (PMR)

Address(es): PORT0.PMR 0008 C060h, PORT1.PMR 0008 C061h, PORT2.PMR 0008 C062h, PORT3.PMR 0008 C063h, PORT4.PMR 0008 C064h, PORT5.PMR 0008 C065h, PORT6.PMR 0008 C066h, PORT7.PMR 0008 C067h, PORT8.PMR 0008 C068h, PORT9.PMR 0008 C069h, PORTA.PMR 0008 C06Ah, PORTB.PMR 0008 C06Bh, PORTD.PMR 0008 C06Dh, PORTE.PMR 0008 C06Eh, PORTN.PMR 0008 C076h

	b7	b6	b5	b4	b3	b2	b1	b0
	B7	B6	B5	B4	B3	B2	B1	B0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm0 Pin Mode Control	0: Uses the pin as a general I/O pin. 1: Uses the pin as a peripheral function.	R/W
b1	B1	Pm1 Pin Mode Control		R/W
b2	B2	Pm2 Pin Mode Control		R/W
b3	B3	Pm3 Pin Mode Control		R/W
b4	B4	Pm4 Pin Mode Control		R/W
b5	B5	Pm5 Pin Mode Control		R/W
b6	B6	Pm6 Pin Mode Control		R/W
b7	B7	Pm7 Pin Mode Control		R/W

m = 0 to 9, A, B, D, E, N

PMR is a register which specifies the function of the pins of the port.
Each bit of PORTm.PMR corresponds to each pin of port m; pin function can be specified in 1-bit units.
The bit corresponding to a pin that does not exist is reserved. A reserved bit is always read as 0. The write value should always be 0.

20.3.5 Open-Drain Control Register 0 (ODR0)

Address(es): PORT0.ODR0 0008 C080h, PORT1.ODR0 0008 C082h, PORT2.ODR0 0008 C084h, PORT3.ODR0 0008 C086h, PORT4.ODR0 0008 C088h, PORT5.ODR0 0008 C08Ah, PORT6.ODR0 0008 C08Ch, PORT7.ODR0 0008 C08Eh, PORT8.ODR0 0008 C090h, PORT9.ODR0 0008 C092h, PORTA.ODR0 0008 C094h, PORTB.ODR0 0008 C096h, PORTD.ODR0 0008 C09Ah, PORTE.ODR0 0008 C09Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	B6	—	B4	—	B2	—	B0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm0 Output Type Select	0: CMOS output 1: NMOS open-drain output	R/W
b1	—	Reserved		R/W
b2	B2	Pm1 Output Type Select		R/W
b3	—	Reserved		R/W
b4	B4	Pm2 Output Type Select		R/W
b5	—	Reserved		R/W
b6	B6	Pm3 Output Type Select		R/W
b7	—	Reserved		R/W

m = 0 to 9, A, B, D, E

ODR0 is a register which is used to select an output type for the pins of the port.
In the ODR0 register, the odd bits (b1, b3, b5, and b7) are reserved.
The bits corresponding to a pin that does not exist or that the open-drain output is not allocated to are reserved. A reserved bit is always read as 0. The write value should always be 0.

20.3.6 Open-Drain Control Register 1 (ODR1)

Address(es): PORT2.ODR1 0008 C085h, PORT3.ODR1 0008 C087h, PORT4.ODR1 0008 C089h, PORT5.ODR1 0008 C08Bh, PORT6.ODR1 0008 C08Dh, PORT7.ODR1 0008 C08Fh, PORT9.ODR1 0008 C093h, PORTA.ODR1 0008 C095h, PORTB.ODR1 0008 C097h, PORTD.ODR1 0008 C09Bh, PORTE.ODR1 0008 C09Dh, PORTN.ODR1 0008 C0ADh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	B6	—	B4	—	B2	—	B0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm4 Output Type Select	0: CMOS output 1: NMOS open-drain output	R/W
b1	—	Reserved		R/W
b2	B2	Pm5 Output Type Select		R/W
b3	—	Reserved		R/W
b4	B4	Pm6 Output Type Select		R/W
b5	—	Reserved		R/W
b6	B6	Pm7 Output Type Select		R/W
b7	—	Reserved		R/W

m = 2 to 7, 9, A, B, D, E, N

ODR1 is used to select an output type for each pin of the port.
The odd bits (b1, b3, b5, and b7) in the ODR1 register are reserved.
The bits corresponding to a pin that does not exist or that the open-drain output is not allocated to are reserved. A reserved bit is always read as 0. The write value should always be 0.

20.3.7 Pull-Up Resistor Control Register (PCR)

Address(es): PORT0.PCR 0008 C0C0h, PORT1.PCR 0008 C0C1h, PORT2.PCR 0008 C0C2h, PORT3.PCR 0008 C0C3h, PORT4.PCR 0008 C0C4h, PORT5.PCR 0008 C0C5h, PORT6.PCR 0008 C0C6h, PORT7.PCR 0008 C0C7h, PORT8.PCR 0008 C0C8h, PORT9.PCR 0008 C0C9h, PORTA.PCR 0008 C0CAh, PORTB.PCR 0008 C0CBh, PORTD.PCR 0008 C0CDh, PORTE.PCR 0008 C0CEh, PORTN.PCR 0008 C0D6h

	b7	b6	b5	b4	b3	b2	b1	b0
	B7	B6	B5	B4	B3	B2	B1	B0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm0 Input Pull-Up Resistor Control	0: Disables an input pull-up resistor. 1: Enables an input pull-up resistor.	R/W
b1	B1	Pm1 Input Pull-Up Resistor Control		R/W
b2	B2	Pm2 Input Pull-Up Resistor Control		R/W
b3	B3	Pm3 Input Pull-Up Resistor Control		R/W
b4	B4	Pm4 Input Pull-Up Resistor Control		R/W
b5	B5	Pm5 Input Pull-Up Resistor Control		R/W
b6	B6	Pm6 Input Pull-Up Resistor Control		R/W
b7	B7	Pm7 Input Pull-Up Resistor Control		R/W

m = 0 to 9, A, B, D, E, N

PCR is a register which enables or disables an input pull-up resistor for individual pins of the port.
While a pin is in the input state with the corresponding bit in PORTm.PCR set to 1, the pull-up resistor connected to the pin is enabled.
When a pin is in use for the general-purpose output port or peripheral function output, set the corresponding bit to 0.
The pull-up resistor is disabled in the reset state.
The bits corresponding to a pin that does not exist or that the input pull-up is not allocated to are reserved. A reserved bit is always read as 0. The write value should always be 0.

20.3.8 Drive Capacity Control Register (DSCR)

Address(es): PORT0.DSCR 0008 C0E0h, PORT1.DSCR 0008 C0E1h, PORT2.DSCR 0008 C0E2h, PORT3.DSCR 0008 C0E3h, PORT7.DSCR 0008 C0E7h, PORT8.DSCR 0008 C0E8h, PORT9.DSCR 0008 C0E9h, PORTA.DSCR 0008 C0EAh, PORTB.DSCR 0008 C0EBh, PORTD.DSCR 0008 C0EDh, PORTE.DSCR 0008 C0EEh, PORTN.DSCR 0008 C0F6h

	b7	b6	b5	b4	b3	b2	b1	b0
	B7	B6	B5	B4	B3	B2	B1	B0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm0 Drive Capacity Control	0: Normal drive output 1: High-drive output	R/W
b1	B1	Pm1 Drive Capacity Control		R/W
b2	B2	Pm2 Drive Capacity Control		R/W
b3	B3	Pm3 Drive Capacity Control		R/W
b4	B4	Pm4 Drive Capacity Control		R/W
b5	B5	Pm5 Drive Capacity Control		R/W
b6	B6	Pm6 Drive Capacity Control		R/W
b7	B7	Pm7 Drive Capacity Control		R/W

m = 0 to 3, 7 to 9, A, B, D, E, and N

DSCR is a register which is used to switch the drive capacity of the port.
When a pin is set for large current output in the DSCR2 register, the drive capacity of that pin cannot be changed. For setting of the drive capacity by the DSCR and DSCR2 registers, refer to Table 20.3, Drive Capacity Setting by DSCR and DSCR2 Registers.
The bits corresponding to a pin that does not exist or whose drive capacity is fixed are reserved. A reserved bit is always read as 0. The write value should always be 0.

20.3.9 Drive Capacity Control Register 2 (DSCR2)

Address(es): PORT7.DSCR2 0008 C12Fh, PORT8.DSCR2 0008 C130h, PORT9.DSCR2 0008 C131h, PORTB.DSCR2 0008 C133h, PORTD.DSCR2 0008 C135h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	B6	B5	B4	B3	B2	B1	B0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm0 Drive Capacity Control 2	0: Normal drive/high-drive output*1 1: Large current output	R/W
b1	B1	Pm1 Drive Capacity Control 2		R/W
b2	B2	Pm2 Drive Capacity Control 2		R/W
b3	B3	Pm3 Drive Capacity Control 2		R/W
b4	B4	Pm4 Drive Capacity Control 2		R/W
b5	B5	Pm5 Drive Capacity Control 2		R/W
b6	B6	Pm6 Drive Capacity Control 2		R/W
b7	—	Reserved		R/W

m = 7 to 9, B, and D

Note 1. Pins that support switching drive capacity by the DSCR register depend on the setting of the DSCR register.

DSCR2 is a register which is used to switch the drive capacity of the port.

Table 20.3 shows the setting of drive capacity by the DSCR and DSCR2 registers.

The bits corresponding to a pin that does not exist or that cannot be switched to large current output are reserved. A reserved bit is always read as 0. The write value should always be 0.

Table 20.3 Drive Capacity Setting by DSCR and DSCR2 Registers

PORTm.DSCR2.Bx	PORTm.DSCR.Bx	Drive Capacity*1
0	0	Normal drive output
0	1	High-drive output
1	Don't care	Large current output

Note 1. When drive capacity is fixed, or drive capacity of a pin whose drive capacity cannot be switched cannot be changed.

20.3.10 Port Output Retention Setting Register 1 (POHSR1)

Address(es): PORT.POHSR1 0008 C110h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	PEHLD	PDHLD	—	PBHLD	PAHLD	P9HLD	P8HLD	P7HLD	P6HLD	P5HLD	P4HLD	P3HLD	P2HLD	P1HLD	P0HLD
Value after reset:	0	0	0	0	0	0	0	0								

Bit	Symbol	Bit Name	Description	R/W
b0	P0HLD	Port 0 Output Retention Control	0: Not retained 1: Retained	R/W
b1	P1HLD	Port 1 Output Retention Control		R/W
b2	P2HLD	Port 2 Output Retention Control		R/W
b3	P3HLD	Port 3 Output Retention Control		R/W
b4	P4HLD	Port 4 Output Retention Control		R/W
b5	P5HLD	Port 5 Output Retention Control		R/W
b6	P6HLD	Port 6 Output Retention Control		R/W
b7	P7HLD	Port 7 Output Retention Control		R/W
b8	P8HLD	Port 8 Output Retention Control		R/W
b9	P9HLD	Port 9 Output Retention Control		R/W
b10	PAHLD	Port A Output Retention Control		R/W
b11	PBHLD	Port B Output Retention Control		R/W
b12	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b13	PDHLD	Port D Output Retention Control	0: Not retained 1: Retained	R/W
b14	PEHLD	Port E Output Retention Control		R/W
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

PmHLD Bit (Port m Output Retention Control) (m = 0 to 9, A, B, D, E)

Each of the bits in this register is used to control whether or not to retain the data output from the following ports: ports 0 to 9, port A, port B, port D, and port E. Setting the POHCR.POHE bit to 1 enables retention of the data set in the PODR register for the ports for which retention is specified (PmHLD = 1) in the POHSR1 and POHSR2 registers. Once retained, the ports output the retained data even if the data in the PODR register is changed. If the PmHLD bit is changed from 1 to 0, the ports output the data in the PODR register regardless of the setting of the POHCR.POHE bit. Note that the setting of this register is only valid for output port pins, but not valid for input port pins.

20.3.11 Port Output Retention Setting Register 2 (POHSR2)

Address(es): PORT.POHSR2 0008 C112h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	PNHLD	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	PNHLD	Port N Output Retention Control	0: Not retained 1: Retained	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

PNHLD Bit (Port N Output Retention Control)

This bit is used to control whether or not to retain the data output from port N. Setting the POHCR.POHE bit to 1 enables retention of the data output from port N, which has been set in the PODR register. Once retained, port N outputs the retained data even if the data in the PODR register is changed. If the PmHLD bit is changed from 1 to 0, port N outputs the data in the PODR register regardless of the setting of the POHCR.POHE bit. Note that the setting of this register is only valid for output port pins, but not valid for input port pins.

20.3.12 Port Output Retention Control Register (POHCR)

Address(es): PORT.POHCR 0008 C114h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	POHE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	POHE	Port Output Retention Enable	0: Retaining of the data output from the ports is disabled. 1: Retaining of the data output from the ports is enabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

POHE Bit (Port Output Retention Enable)

This bit enables or disables retaining of the data output from the ports for which retention is specified in the POHSR1 and POHSR2 registers. Setting this bit to 1 enables retention of the data set in the PODR register. Only 1 can be written to this bit, and it is automatically cleared after that.

For details on this functionality, see section 20.4, Retaining the Data Output from Ports.

20.3.13 General-purpose Input/Output Pin Select Extension Register (GPSEXT)

Address(es): PORT.GPSEXT 0008 C122h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	GPSMD	GPSEMLE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	GPSEMLE	EMLE Pin Function Select	0: The EMLE function is selected. 1: A function other than EMLE is selected.	R/W
b1	GPSMD	MD Pin Function Select	0: The MD/FINED function is selected. 1: The general-purpose input/output pin function is selected.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

GPSEMLE Bit (EMLE Pin Function Select)

This bit selects the use of the EMLE function or another function on the EMLE pin. When the on-chip emulator is to be used, leave this bit set to 0. Do not set this bit to 0 when its setting is 1. Do not set this bit to 1 on the 48-pin products. For points to note when resetting this MCU through the RES# pin, see section 20.5.3, Points to Note in Handling of the EMLE Pin.

The pull-down resistor connected to the EMLE pin is disabled when a function other than EMLE is selected.

GPSMD Bit (MD Pin Function Select)

This bit selects use of the MD/FINED function or the general-purpose input/output pin function.

The pull-up resistor connected to the MD pin is disabled when the general-purpose input/output pin function is selected.

20.4 Retaining the Data Output from Ports

Setting the POHCR.POHE bit to 1 enables retention of the data set in the PODR register for the ports for which retention is specified (PmHLD = 1) in the POHSR1 and POHSR2 registers. Using this functionality allows simultaneous change of data output from the specified ports. Figure 20.7 is an example of the flow of settings for the simultaneous change of output data.

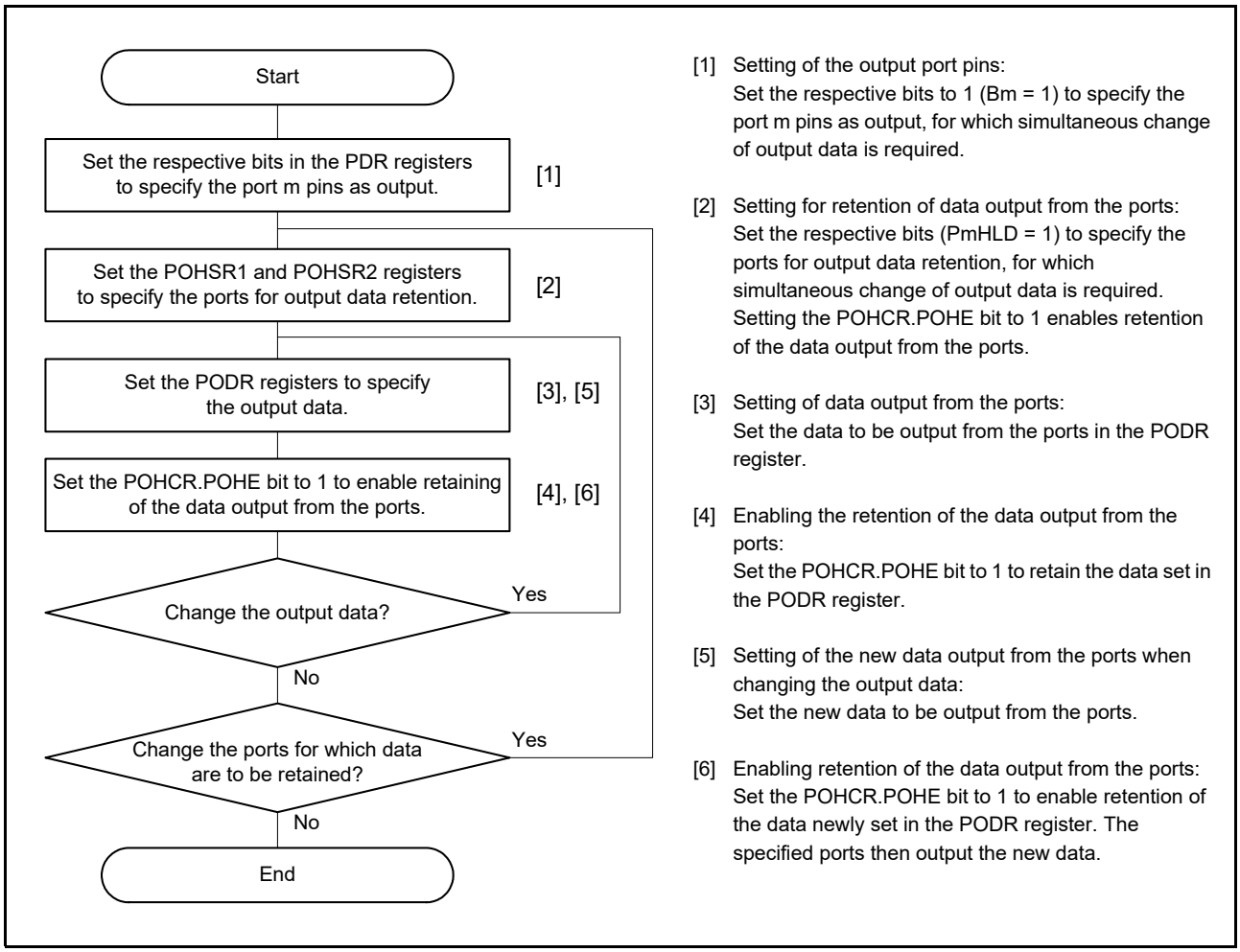


Figure 20.7 Example of the Flow of Settings for the Simultaneous Change of Output Data

20.5 Usage Notes

20.5.1 Initialization of the Port Direction Register (PDR)

Initialize reserved bits in the PDR register according to Table 20.4 to Table 20.7.

- The blank columns in Table 20.4 to Table 20.7 indicate the bits corresponding to the pins listed in Table 20.1, Specifications of I/O Ports.
  The corresponding bits should be set to 1 (output) or 0 (input) depending on the user system.
  However, each bit of PDR corresponding to PE2 pins which are input only is reserved.
  This bit should be set to 0 (input).
- The columns other than the blank columns in Table 20.4 to Table 20.7 indicate reserved bits (input-dedicated pins are excluded).
  A reserved bit should be set to 0 (input) or 1 (output) according to Table 20.4 to Table 20.7.
  When setting a value to a reserved bit, access in byte units.

Table 20.4 PDR Register Settings in 100-Pin Packages

	PDR Register							
Port Symbol	**b7**	**b6**	**b5**	**b4**	**b3**	**b2**	**b1**	**b0**
PORT0	0	0	0	0	0	0		
PORT1	0	0	0	0	0	0		
PORT2		0	0					
PORT3			0	0				
PORT4								
PORT5	0	0						
PORT6	0	0						
PORT7	0							
PORT8	0	0	0	0	0			
PORT9	0							
PORTA	0	0						
PORTB								
PORTD								
PORTE	0	0				0		
PORTN			0	0	0	0	0	0

Table 20.5 PDR Register Settings in 80-Pin

	PDR Register							
Port Symbol	**b7**	**b6**	**b5**	**b4**	**b3**	**b2**	**b1**	**b0**
PORT0	0	0	0	0	0	0		
PORT1	0	0	0	0	0	0		
PORT2		0	0	1	1			
PORT3			0	0	1	1		
PORT4								
PORT5	0	0						
PORT6	0	0			1	1	1	
PORT7	0							
PORT8	0	0	0	0	0	1	1	1
PORT9	0							
PORTA	0	0		1		1	1	1
PORTB	1							
PORTD							1	1
PORTE	0	0	1			0	1	1
PORTN			0	0	0	0	0	0

Table 20.6 PDR Register Settings in 64-Pin Packages

	PDR Register							
Port Symbol	**b7**	**b6**	**b5**	**b4**	**b3**	**b2**	**b1**	**b0**
PORT0	0	0	0	0	0	0		
PORT1	0	0	0	0	0	0		1
PORT2	1	0	0	1	1			
PORT3			0	0	1	1	1	1
PORT4								
PORT5	0	0	1				1	1
PORT6	0	0			1	1	1	1
PORT7	0							
PORT8	0	0	0	0	0	1	1	1
PORT9	0							
PORTA	0	0	1	1	1	1	1	1
PORTB	1							
PORTD						1	1	1
PORTE	0	0	1	1	1	0	1	1
PORTN			0	0	0	0	0	0

Table 20.7 PDR Register Settings in 48-Pin Packages

	PDR Register							
Port Symbol	**b7**	**b6**	**b5**	**b4**	**b3**	**b2**	**b1**	**b0**
PORT0	0	0	0	0	0	0	1	
PORT1	0	0	0	0	0	0		
PORT2	1	0	0	1	1	1		
PORT3			0	0	1	1	1	1
PORT4	1	1	1					
PORT5	0	0	1	1			1	1
PORT6	0	0	1	1	1		1	1
PORT7	0							1
PORT8	0	0	0	0	0	1	1	1
PORT9	0	1						1
PORTA	0	0	1	1	1	1	1	1
PORTB	1							
PORTD		1		1		1	1	1
PORTE	0	0	1	1	1	0	1	1
PORTN			0	0	0	0	0	0

20.5.2 Handling of Unused Pins

Details on the handling of unused pins are given in Table 20.8.

Table 20.8 Handling of Unused Pins

Pin Name	Handling
PN7/EMLE	Connect this pin to VSS via a resistor (pulling down), or leave this pin open-circuit.
PN6/MD/FINED	Use this as a mode pin.
RES#	Connect this pin to VCC via a resistor (pulling up).
PE2/NMI	Connect this pin to VCC via a resistor (pulling up).
P36/EXTAL	Set the MOSCCR.MOSTP bit to 1 (general port P36) when not using the main clock When this pin is not used as port P36, handle as port 0 to 3, 7 to 9, A, B, D, E.
P37/XTAL	Set the MOSCCR.MOSTP bit to 1 (general port P37) when not using the main clock When this pin is not used as port P37, handle as port 0 to 3, 7 to 9, A, B, D, E. When an external clock is to be input to the EXTAL pin, leave this pin open-circuit.
Port 0 to Port 3, Port 7 to Port 9, Port A, Port B, Port D, Port E	• If the direction setting is for input (PORTn.PDR = 0), the corresponding pin is connected to VCC (pulled up) via a resistor or to VSS (pulled down) via a resistor.*1 • If the direction setting is for output (PORTn.PDR = 1), leave these pins open-circuit.*1, *2
Port 4 to Port 6	• The corresponding pin is connected to analog power supply (AVCC1/AVCC2) (pulled up) via a resistor or to analog ground (AVSS1/AVSS2) (pulled down) via a resistor.*3 • If the direction setting is for output (PORTn.PDR = 1), leave these pins open-circuit.*1, *2, *4

Note 1. Clear the PORTn.PMR bit, the PmnPFS.ISEL bit and the PmnPFS.ASEL bit to 0.
Note 2. In the case of release when the setting is for output, the port is an input over the period from release from the reset state to the pin becoming an output. Since the voltage on the pin is undefined while it is an input, this may lead to an increase in the current drawn.
Note 3. Connect the P4 pin to AVCC1/AVSS1 as the power source via resistor, and connect the P5 and P6 pins to AVCC2/AVSS2 as the power source via resistor.
Note 4. Do not set PORTn.PDR to 1 for input-dedicated pins.

20.5.3 Points to Note in Handling of the EMLE Pin

When resetting the MCU through the RES# pin and the pin on which the EMLE function is multiplexed is to be used as a general-purpose input/output port (PN7), ensure that a low-level signal is input to the pin or the pin is left open-circuit during a reset. If the MCU has been reset through the RES# pin while a high-level signal was being input to the EMLE pin (the default function of the pin following a reset), the PD3 pin might output a signal if both of the following conditions are satisfied, depending on the level of the signal to be input to the PD6 pin.

- The high level is being applied to the PD7 pin.
- At least four cycles of a clock or clock-like signal waveform have been input to the PD4 pin.

21. Multi-Function Pin Controller (MPC)

21.1 Overview

The multi-function pin controller (MPC) selects and assigns input/output of peripheral functions and interrupt input signals from multiple ports.

Table 21.1 lists the functions assigned to each multiplexed pin. The symbols ✓ and × in the table indicate whether the pin is available or unavailable for the package. Selecting a single function for multiple pins is prohibited.

Table 21.1 Functions Assigned to Each Multiplexed Pin (1/15)

Module/Function	Channel	Pin Functions	Allocation Port	Package					
				With 64 Kbytes of RAM				With 48 Kbytes of RAM	
				100-pin	80-pin	64-pin	48-pin	64-pin	48-pin
Interrupt		NMI (input)	PE2	✓	✓	✓	✓	✓	✓
Interrupt	IRQ0	IRQ0 (input)	P10	✓	✓	×	✓	×	✓
			P52	✓	✓	✓	✓	✓	✓
			PE2	✓	✓	✓	✓	✓	✓
			PE5	✓	×	×	×	×	×
	IRQ1	IRQ1 (input)	P11	✓	✓	✓	✓	✓	✓
			P53	✓	✓	✓	✓	✓	✓
			P95	✓	✓	✓	✓	✓	✓
			PA5	✓	✓	×	×	×	×
			PE4	✓	✓	×	×	×	×
	IRQ2	IRQ2 (input)	P00	✓	✓	✓	✓	✓	✓
			P54	✓	✓	✓	×	✓	×
			PB6	✓	✓	✓	✓	✓	✓
			PD4	✓	✓	✓	×	✓	×
			PE3	✓	✓	×	×	×	×
	IRQ3	IRQ3 (input)	P55	✓	✓	×	×	×	×
			P82	✓	×	×	×	×	×
			PB4	✓	✓	✓	✓	✓	✓
	IRQ4	IRQ4 (input)	P01	✓	✓	✓	×	✓	×
			P24	✓	×	×	×	×	×
			P60	✓	✓	×	×	×	×
			P96	✓	✓	✓	×	✓	×
			PB1	✓	✓	✓	✓	✓	✓
	IRQ5	IRQ5 (input)	P61	✓	×	×	×	×	×
			P70	✓	✓	✓	×	✓	×
			P80	✓	×	×	×	×	×
			PD6	✓	✓	✓	×	✓	×
			PN7	✓	✓	✓	×	✓	×
	IRQ6	IRQ6 (input)	P21	✓	✓	✓	✓	✓	✓
			P31	✓	✓	×	×	×	×
			P62	✓	×	×	✓	×	✓
			PD5	✓	✓	✓	✓	✓	✓
	IRQ7	IRQ7 (input)	P20	✓	✓	✓	✓	✓	✓
			P30	✓	✓	×	×	×	×
			P63	✓	×	×	×	×	×
			PE0	✓	×	×	×	×	×

Table 21.1 Functions Assigned to Each Multiplexed Pin (2/15)

Module/Function	Channel	Pin Functions	Allocation Port	Package					
				With 64 Kbytes of RAM				With 48 Kbytes of RAM	
				100-pin	80-pin	64-pin	48-pin	64-pin	48-pin
Interrupt	IRQ8	IRQ8 (input)	P64	✓	✓	✓	×	✓	×
			PB0	✓	✓	✓	✓	✓	✓
			PD7	✓	✓	✓	✓	✓	✓
	IRQ9	IRQ9 (input)	P65	✓	✓	✓	×	✓	×
			PB3	✓	✓	✓	✓	✓	✓
	IRQ10	IRQ10 (input)	P22	✓	✓	✓	×	✓	×
	IRQ11	IRQ11 (input)	P23	✓	×	×	×	×	×
	IRQ12	IRQ12 (input)	P32	✓	×	×	×	×	×
	IRQ13	IRQ13 (input)	P33	✓	×	×	×	×	×
	IRQ14	IRQ14 (input)	P93	✓	✓	✓	✓	✓	✓
			PA1	✓	×	×	×	×	×
	IRQ15	IRQ15 (input)	P27	✓	✓	×	×	×	×
			PE1	✓	×	×	×	×	×
Multi-function timer unit 3	MTU0	MTIOC0A (input/output) /MTIOC0A# (input/output)	P31	✓	✓	×	×	×	×
			P70	✓	✓	✓	×	✓	×
			PB3	✓	✓	✓	✓	✓	✓
		MTIOC0B (input/output) /MTIOC0B# (input/output)	P30	✓	✓	×	×	×	×
			PB2	✓	✓	✓	✓	✓	✓
		MTIOC0C (input/output) /MTIOC0C# (input/output)	P27	✓	✓	×	×	×	×
			PB1	✓	✓	✓	✓	✓	✓
		MTIOC0D (input/output) /MTIOC0D# (input/output)	PB0	✓	✓	✓	✓	✓	✓
	MTU1	MTIOC1A (input/output) /MTIOC1A# (input/output)	P27	✓	✓	×	×	×	×
			P95	✓	✓	✓	✓	✓	✓
			PA5	✓	✓	×	×	×	×
		MTIOC1B (input/output) /MTIOC1B# (input/output)	PA4	✓	×	×	×	×	×
	MTU2	MTIOC2A (input/output) /MTIOC2A# (input/output)	P94	✓	✓	✓	✓	✓	✓
			PA3	✓	✓	×	×	×	×
		MTIOC2B (input/output) /MTIOC2B# (input/output)	PA2	✓	×	×	×	×	×
	MTU3	MTIOC3A (input/output) /MTIOC3A# (input/output)	P11	✓	✓	✓	✓	✓	✓
			P33	✓	×	×	×	×	×
		MTIOC3B (input/output) /MTIOC3B# (input/output)	P71	✓	✓	✓	✓	✓	✓
		MTIOC3C (input/output) /MTIOC3C# (input/output)	P32	✓	×	×	×	×	×
		MTIOC3D (input/output) /MTIOC3D# (input/output)	P74	✓	✓	✓	✓	✓	✓
	MTU4	MTIOC4A (input/output) /MTIOC4A# (input/output)	P72	✓	✓	✓	✓	✓	✓
		MTIOC4B (input/output) /MTIOC4B# (input/output)	P73	✓	✓	✓	✓	✓	✓
		MTIOC4C (input/output) /MTIOC4C# (input/output)	P75	✓	✓	✓	✓	✓	✓
		MTIOC4D (input/output) /MTIOC4D# (input/output)	P76	✓	✓	✓	✓	✓	✓

Table 21.1 Functions Assigned to Each Multiplexed Pin (3/15)

Module/Function	Channel	Pin Functions	Allocation Port	Package					
				With 64 Kbytes of RAM				With 48 Kbytes of RAM	
				100-pin	80-pin	64-pin	48-pin	64-pin	48-pin
Multi-function timer unit 3	MTU5	MTIC5U (input) /MTIC5U# (input)	P24	✓	×	×	×	×	×
			P82	✓	×	×	×	×	×
		MTIC5V (input) /MTIC5V# (input)	P23	✓	×	×	×	×	×
			P81	✓	×	×	×	×	×
		MTIC5W (input) /MTIC5W# (input)	P22	✓	✓	✓	×	✓	×
			P80	✓	×	×	×	×	×
	MTU6	MTIOC6A (input/output) /MTIOC6A# (input/output)	P93	✓	✓	✓	✓	✓	✓
			PA1	✓	×	×	×	×	×
		MTIOC6B (input/output) /MTIOC6B# (input/output)	P95	✓	✓	✓	✓	✓	✓
		MTIOC6C (input/output) /MTIOC6C# (input/output)	P92	✓	✓	✓	✓	✓	✓
			PA0	✓	×	×	×	×	×
		MTIOC6D (input/output) /MTIOC6D# (input/output)	P92	✓	✓	✓	✓	✓	✓
	MTU7	MTIOC7A (input/output) /MTIOC7A# (input/output)	P94	✓	✓	✓	✓	✓	✓
		MTIOC7B (input/output) /MTIOC7B# (input/output)	P93	✓	✓	✓	✓	✓	✓
		MTIOC7C (input/output) /MTIOC7C# (input/output)	P91	✓	✓	✓	✓	✓	✓
		MTIOC7D (input/output) /MTIOC7D# (input/output)	P90	✓	✓	✓	×	✓	×
	MTU9	MTIOC9A (input/output) /MTIOC9A# (input/output)	P00	✓	✓	✓	✓	✓	✓
			P21	✓	✓	✓	✓	✓	✓
			PD7	✓	✓	✓	✓	✓	✓
		MTIOC9B (input/output)	P22	✓	✓	✓	×	✓	×
		MTIOC9B (input/output) /MTIOC9B# (input/output)	P10	✓	✓	×	✓	×	✓
			PE0	✓	×	×	×	×	×
		MTIOC9C (input/output) /MTIOC9C# (input/output)	P01	✓	✓	✓	×	✓	×
			P20	✓	✓	✓	✓	✓	✓
			PD6	✓	✓	✓	×	✓	×
		MTIOC9D (input/output)	P11	✓	✓	✓	✓	✓	✓
		MTIOC9D (input/output) /MTIOC9D# (input/output)	PE1	✓	×	×	×	×	×
			PE5	✓	×	×	×	×	×
			PN7	✓	✓	✓	×	✓	×
	MTU	MTCLKA (input) /MTCLKA# (input)	P21	✓	✓	✓	✓	✓	✓
			P33	✓	×	×	×	×	×
		MTCLKB (input) /MTCLKB# (input)	P20	✓	✓	✓	✓	✓	✓
			P32	✓	×	×	×	×	×
		MTCLKC (input) /MTCLKC# (input)	P11	✓	✓	✓	✓	✓	✓
			P31	✓	✓	×	×	×	×
			P70	✓	✓	✓	×	✓	×
			PE4	✓	✓	×	×	×	×
		MTCLKD (input) /MTCLKD# (input)	P10	✓	✓	×	✓	×	✓
			P22	✓	✓	✓	×	✓	×
			P30	✓	✓	×	×	×	×
			PE3	✓	✓	×	×	×	×

Table 21.1 Functions Assigned to Each Multiplexed Pin (4/15)

Module/Function	Channel	Pin Functions	Allocation Port	Package					
				With 64 Kbytes of RAM				With 48 Kbytes of RAM	
				100-pin	80-pin	64-pin	48-pin	64-pin	48-pin
Multi-function timer unit 3	MTU	ADSM0 (output)	PB2	✓	✓	✓	✓	✓	✓
		ADSM1 (output)	PB1	✓	✓	✓	✓	✓	✓
General PWM timer	GPTW0	GTIOC0A (input/output) /GTIOC0A# (input/output)	P71	✓	✓	✓	✓	✓	✓
			PD2	✓	✓	×	×	×	×
			PD7	✓	✓	✓	✓	✓	✓
		GTIOC0B (input/output) /GTIOC0B# (input/output)	P74	✓	✓	✓	✓	✓	✓
			PD1	✓	×	×	×	×	×
			PD6	✓	✓	✓	×	✓	×
	GPTW1	GTIOC1A (input/output) /GTIOC1A# (input/output)	P72	✓	✓	✓	✓	✓	✓
			PD0	✓	×	×	×	×	×
			PD5	✓	✓	✓	✓	✓	✓
		GTIOC1B (input/output) /GTIOC1B# (input/output)	P75	✓	✓	✓	✓	✓	✓
			PB7	✓	×	×	×	×	×
			PD4	✓	✓	✓	×	✓	×
	GPTW2	GTIOC2A (input/output) /GTIOC2A# (input/output)	P73	✓	✓	✓	✓	✓	✓
			PB6	✓	✓	✓	✓	✓	✓
			PD3	✓	✓	✓	✓	✓	✓
		GTIOC2B (input/output) /GTIOC2B# (input/output)	P76	✓	✓	✓	✓	✓	✓
			PB5	✓	✓	✓	✓	✓	✓
			PD2	✓	✓	×	×	×	×
	GPTW3	GTIOC3A (input/output) /GTIOC3A# (input/output)	P10	✓	✓	×	✓	×	✓
			P32	✓	×	×	×	×	×
			PB6	✓	✓	✓	✓	✓	✓
			PD1	✓	×	×	×	×	×
			PD7	✓	✓	✓	✓	✓	✓
			PE5	✓	×	×	×	×	×
		GTIOC3B (input/output) /GTIOC3B# (input/output)	P11	✓	✓	✓	✓	✓	✓
			P33	✓	×	×	×	×	×
			PB5	✓	✓	✓	✓	✓	✓
			PD0	✓	×	×	×	×	×
			PD6	✓	✓	✓	×	✓	×
	GPTW4	GTIOC4A (input/output) /GTIOC4A# (input/output)	P71	✓	✓	✓	✓	✓	✓
			P95	✓	✓	✓	✓	✓	✓
		GTIOC4B (input/output) /GTIOC4B# (input/output)	P74	✓	✓	✓	✓	✓	✓
			P92	✓	✓	✓	✓	✓	✓
	GPTW5	GTIOC5A (input/output) /GTIOC5A# (input/output)	P72	✓	✓	✓	✓	✓	✓
			P94	✓	✓	✓	✓	✓	✓
		GTIOC5B (input/output) /GTIOC5B# (input/output)	P75	✓	✓	✓	✓	✓	✓
			P91	✓	✓	✓	✓	✓	✓
	GPTW6	GTIOC6A (input/output) /GTIOC6A# (input/output)	P73	✓	✓	✓	✓	✓	✓
			P93	✓	✓	✓	✓	✓	✓
		GTIOC6B (input/output) /GTIOC6B# (input/output)	P76	✓	✓	✓	✓	✓	✓
			P90	✓	✓	✓	×	✓	×

Table 21.1 Functions Assigned to Each Multiplexed Pin (5/15)

Module/Function	Channel	Pin Functions	Allocation Port	Package					
				With 64 Kbytes of RAM				With 48 Kbytes of RAM	
				100-pin	80-pin	64-pin	48-pin	64-pin	48-pin
General PWM timer	GPTW7	GTIOC7A (input/output) /GTIOC7A# (input/output)	P32	✓	×	×	×	×	×
			P95	✓	✓	✓	✓	✓	✓
			PB2	✓	✓	✓	✓	✓	✓
		GTIOC7A (input/output)	PD5	✓	✓	✓	✓	✓	✓
		GTIOC7B (input/output) /GTIOC7B# (input/output)	P33	✓	×	×	×	×	×
			P92	✓	✓	✓	✓	✓	✓
			PB1	✓	✓	✓	✓	✓	✓
		GTIOC7B (input/output)	PD3	✓	✓	✓	✓	✓	✓
	GPTW	GTETRGA (input)	P01	✓	✓	✓	×	✓	×
			P11	✓	✓	✓	✓	✓	✓
			P70	✓	✓	✓	×	✓	×
			P96	✓	✓	✓	×	✓	×
			PB4	✓	✓	✓	✓	✓	✓
			PD5	✓	✓	✓	✓	✓	✓
			PE3	✓	✓	×	×	×	×
			PE4	✓	✓	×	×	×	×
		GTETRGB (input)	P01	✓	✓	✓	×	✓	×
			P10	✓	✓	×	✓	×	✓
			P70	✓	✓	✓	×	✓	×
			P96	✓	✓	✓	×	✓	×
			PB4	✓	✓	✓	✓	✓	✓
			PD4	✓	✓	✓	×	✓	×
			PE3	✓	✓	×	×	×	×
			PE4	✓	✓	×	×	×	×
			PE5	✓	×	×	×	×	×
		GTETRGC (input)	P01	✓	✓	✓	×	✓	×
			P11	✓	✓	✓	✓	✓	✓
			P70	✓	✓	✓	×	✓	×
			P96	✓	✓	✓	×	✓	×
			PB4	✓	✓	✓	✓	✓	✓
			PD3	✓	✓	✓	✓	✓	✓
			PE3	✓	✓	×	×	×	×
			PE4	✓	✓	×	×	×	×
		GTETRGD (input)	P01	✓	✓	✓	×	✓	×
			P10	✓	✓	×	✓	×	✓
			P70	✓	✓	✓	×	✓	×
			P96	✓	✓	✓	×	✓	×
			PB4	✓	✓	✓	✓	✓	✓
			PE3	✓	✓	×	×	×	×
			PE4	✓	✓	×	×	×	×
			PE5	✓	×	×	×	×	×
		GTADSM0 (output)	P94	✓	✓	✓	✓	✓	✓
			PA3	✓	✓	×	×	×	×
			PB2	✓	✓	✓	✓	✓	✓
		GTADSM1 (output)	PA2	✓	×	×	×	×	×
			PB1	✓	✓	✓	✓	✓	✓

Table 21.1 Functions Assigned to Each Multiplexed Pin (6/15)

Module/Function	Channel	Pin Functions	Allocation Port	Package					
				With 64 Kbytes of RAM				With 48 Kbytes of RAM	
				100-pin	80-pin	64-pin	48-pin	64-pin	48-pin
General PWM timer	GPTW	GTCPPO0 (output)	P11	✓	✓	✓	✓	✓	✓
			P33	✓	×	×	×	×	×
			P70	✓	✓	✓	×	✓	×
			PB4	✓	✓	✓	✓	✓	✓
		GTCPPO4 (output)	P96	✓	✓	✓	×	✓	×
			PA1	✓	×	×	×	×	×
		GTIU (input)	P00	✓	✓	✓	✓	✓	✓
			P21	✓	✓	✓	✓	✓	✓
			P31	✓	✓	×	×	×	×
			PB3	✓	✓	✓	✓	✓	✓
			PD7	✓	✓	✓	✓	✓	✓
		GTIV (input)	P10	✓	✓	×	✓	×	✓
			P22	✓	✓	✓	×	✓	×
			P30	✓	✓	×	×	×	×
			PB2	✓	✓	✓	✓	✓	✓
			PE0	✓	×	×	×	×	×
		GTIW (input)	P01	✓	✓	✓	×	✓	×
			P20	✓	✓	✓	✓	✓	✓
			PB1	✓	✓	✓	✓	✓	✓
			PD6	✓	✓	✓	×	✓	×
		GTOULO (output)	P74	✓	✓	✓	✓	✓	✓
			P92	✓	✓	✓	✓	✓	✓
		GTOUUP (output)	P71	✓	✓	✓	✓	✓	✓
			P95	✓	✓	✓	✓	✓	✓
		GTOVLO (output)	P75	✓	✓	✓	✓	✓	✓
			P91	✓	✓	✓	✓	✓	✓
		GTOVUP (output)	P72	✓	✓	✓	✓	✓	✓
			P94	✓	✓	✓	✓	✓	✓
		GTOWLO (output)	P76	✓	✓	✓	✓	✓	✓
			P90	✓	✓	✓	×	✓	×
		GTOWUP (output)	P73	✓	✓	✓	✓	✓	✓
			P93	✓	✓	✓	✓	✓	✓
Port output enable 3	POE0	POE0# (input)	P70	✓	✓	✓	×	✓	×
	POE4	POE4# (input)	P96	✓	✓	✓	×	✓	×
	POE8	POE8# (input)	PB4	✓	✓	✓	✓	✓	✓
	POE9	POE9# (input)	P11	✓	✓	✓	✓	✓	✓
			P27	✓	✓	×	×	×	×
	POE10	POE10# (input)	PE2	✓	✓	✓	✓	✓	✓
			PE4	✓	✓	×	×	×	×
	POE11	POE11# (input)	PE3	✓	✓	×	×	×	×
	POE12	POE12# (input)	P01	✓	✓	✓	×	✓	×
			P10	✓	✓	×	✓	×	✓

Table 21.1 Functions Assigned to Each Multiplexed Pin (7/15)

Module/Function	Channel	Pin Functions	Allocation Port	Package					
				With 64 Kbytes of RAM				With 48 Kbytes of RAM	
				100-pin	80-pin	64-pin	48-pin	64-pin	48-pin
8-bit timer	TMR0	TMO0 (output)	P33	✓	×	×	×	×	×
			PB0	✓	✓	✓	✓	✓	✓
			PD3	✓	✓	✓	✓	✓	✓
		TMCI0 (input)	PB1	✓	✓	✓	✓	✓	✓
			PD4	✓	✓	✓	×	✓	×
		TMRI0 (input)	PB2	✓	✓	✓	✓	✓	✓
			PD5	✓	✓	✓	✓	✓	✓
	TMR1	TMO1 (output)	PD6	✓	✓	✓	×	✓	×
		TMCI1 (input)	PD2	✓	✓	×	×	×	×
			PE0	✓	×	×	×	×	×
		TMRI1 (input)	PD7	✓	✓	✓	✓	✓	✓
	TMR2	TMO2 (output)	P20	✓	✓	✓	✓	✓	✓
			P23	✓	×	×	×	×	×
			P27	✓	✓	×	×	×	×
			P92	✓	✓	✓	✓	✓	✓
			PA0	✓	×	×	×	×	×
			PD1	✓	×	×	×	×	×
		TMCI2 (input)	P24	✓	×	×	×	×	×
		TMRI2 (input)	P22	✓	✓	✓	×	✓	×
	TMR3	TMO3 (output)	P11	✓	✓	✓	✓	✓	✓
		TMCI3 (input)	P95	✓	✓	✓	✓	✓	✓
			PA5	✓	✓	×	×	×	×
		TMRI3 (input)	P10	✓	✓	×	✓	×	✓
	TMR4	TMO4 (output)	P22	✓	✓	✓	×	✓	×
			P82	✓	×	×	×	×	×
			P93	✓	✓	✓	✓	✓	✓
			PA1	✓	×	×	×	×	×
			PD2	✓	✓	×	×	×	×
		TMCI4 (input)	P21	✓	✓	✓	✓	✓	✓
			P81	✓	×	×	×	×	×
		TMRI4 (input)	P20	✓	✓	✓	✓	✓	✓
			P80	✓	×	×	×	×	×
	TMR5	TMO5 (output)	PE1	✓	×	×	×	×	×
		TMCI5 (input)	PE0	✓	×	×	×	×	×
		TMRI5 (input)	PD7	✓	✓	✓	✓	✓	✓
	TMR6	TMO6 (output)	P21	✓	✓	✓	✓	✓	✓
			P24	✓	×	×	×	×	×
			P27	✓	✓	×	×	×	×
			P32	✓	×	×	×	×	×
			PD0	✓	×	×	×	×	×
		TMCI6 (input)	P30	✓	✓	×	×	×	×
			PD4	✓	✓	✓	×	✓	×
		TMRI6 (input)	P31	✓	✓	×	×	×	×
			P70	✓	✓	✓	×	✓	×
			PD5	✓	✓	✓	✓	✓	✓

Table 21.1 Functions Assigned to Each Multiplexed Pin (8/15)

Module/Function	Channel	Pin Functions	Allocation Port	Package					
				With 64 Kbytes of RAM				With 48 Kbytes of RAM	
				100-pin	80-pin	64-pin	48-pin	64-pin	48-pin
8-bit timer	TMR7	TMO7 (output)	PA2	✓	×	×	×	×	×
		TMCI7 (input)	PA4	✓	×	×	×	×	×
		TMRI7 (input)	P94	✓	✓	✓	✓	✓	✓
			PA3	✓	✓	×	×	×	×
Compare match timer W	CMTW0	TOC0 (output)	PB6	✓	✓	✓	✓	✓	✓
		TIC0 (input)	PB5	✓	✓	✓	✓	✓	✓
		TOC1 (output)	PB3	✓	✓	✓	✓	✓	✓
		TIC1 (input)	PB2	✓	✓	✓	✓	✓	✓
	CMTW1	TOC2 (output)	PB1	✓	✓	✓	✓	✓	✓
		TIC2 (input)	PB0	✓	✓	✓	✓	✓	✓
		TOC3 (output)	P11	✓	✓	✓	✓	✓	✓
		TIC3 (input)	P00	✓	✓	✓	✓	✓	✓
			P10	✓	✓	×	✓	×	✓
Serial communications interface	SCI1	RXD1 (input) /SMISO1 (input/output) /SSCL1 (input/output)	PD5	✓	✓	✓	✓	✓	✓
		TXD1 (output) /SMOSI1 (input/output) /SSDA1 (input/output)	PD3	✓	✓	✓	✓	✓	✓
		SCK1 (input/output)	PD4	✓	✓	✓	×	✓	×
		CTS1# (input) /RTS1# (output) /SS1# (input)	PD6	✓	✓	✓	×	✓	×
	SCI5	RXD5 (input) /SMISO5 (input/output) /SSCL5 (input/output)	P37	✓	✓	✓	✓	✓	✓
			P91	✓	✓	✓	✓	✓	✓
			PB6	✓	✓	✓	✓	✓	✓
			PE0	✓	×	×	×	×	×
		TXD5 (output) /SMOSI5 (input/output) /SSDA5 (input/output)	P36	✓	✓	✓	✓	✓	✓
			P90	✓	✓	✓	×	✓	×
			PB5	✓	✓	✓	✓	✓	✓
			PD7	✓	✓	✓	✓	✓	✓
		SCK5 (input/output)	P70	✓	✓	✓	×	✓	×
			PB7	✓	×	×	×	×	×
			PD2	✓	✓	×	×	×	×
		CTS5# (input) /RTS5# (output) /SS5# (input)	PB4	✓	✓	✓	✓	✓	✓
			PE1	✓	×	×	×	×	×
	SCI6	RXD6 (input) /SMISO6 (input/output) /SSCL6 (input/output)	P80	✓	×	×	×	×	×
			P95	✓	✓	✓	✓	✓	✓
			PA5	✓	✓	×	×	×	×
			PB1	✓	✓	✓	✓	✓	✓
		TXD6 (output) /SMOSI6 (input/output) /SSDA6 (input/output)	P81	✓	×	×	×	×	×
			PB0	✓	✓	✓	✓	✓	✓
			PB2	✓	✓	✓	✓	✓	✓
		SCK6 (input/output)	P82	✓	×	×	×	×	×
			PA4	✓	×	×	×	×	×
			PB3	✓	✓	✓	✓	✓	✓

Table 21.1 Functions Assigned to Each Multiplexed Pin (9/15)

Module/Function	Channel	Pin Functions	Allocation Port	Package					
				With 64 Kbytes of RAM				With 48 Kbytes of RAM	
				100-pin	80-pin	64-pin	48-pin	64-pin	48-pin
Serial communications interface	SCI6	CTS6# (input) /RTS6# (output) /SS6# (input)	P10	✓	✓	×	✓	×	✓
			PA2	✓	×	×	×	×	×
	SCI12	RXD12 (input) /SMISO12 (input/output) /SSCL12 (input/output) /RXDX12 (input)	P00	✓	✓	✓	✓	✓	✓
			P22	✓	✓	✓	×	✓	×
			P80	✓	×	×	×	×	×
			PB4	✓	✓	✓	✓	✓	✓
			PB6	✓	✓	✓	✓	✓	✓
			PD6	✓	✓	✓	×	✓	×
		TXD12 (output) /SMOSI12 (input/output) /SSDA12 (input/output) /TXDX12 (output) /SIOX12 (input/output)	P01	✓	✓	✓	×	✓	×
			P21	✓	✓	✓	✓	✓	✓
			P23	✓	×	×	×	×	×
			P81	✓	×	×	×	×	×
			PB3	✓	✓	✓	✓	✓	✓
			PB5	✓	✓	✓	✓	✓	✓
			PD4	✓	✓	✓	×	✓	×
		SCK12 (input/output)	P82	✓	×	×	×	×	×
			PB7	✓	×	×	×	×	×
		CTS12# (input) /RTS12# (output) /SS12# (input)	PE1	✓	×	×	×	×	×
Serial communications interface	RSCI8	RXD008 (input) /SMISO008 (input/output) /SSCL008 (input/output)	P20	✓	✓	✓	✓	×	×
			P22	✓	✓	✓	×	×	×
			P95	✓	✓	✓	✓	×	×
			PA5	✓	✓	×	×	×	×
			PD1	✓	×	×	×	×	×
		TXD008 (output) /TXDA008 (output) /SMOSI008 (input/output) /SSDA008 (input/output)	P21	✓	✓	✓	✓	×	×
			P23	✓	×	×	×	×	×
			PA4	✓	×	×	×	×	×
			PB0	✓	✓	✓	✓	×	×
			PD0	✓	×	×	×	×	×
			PD7	✓	✓	✓	✓	×	×
		SCK008 (input/output)	P11	✓	✓	✓	✓	×	×
			P22	✓	✓	✓	×	×	×
			P24	✓	×	×	×	×	×
			P30	✓	✓	×	×	×	×
			P94	✓	✓	✓	✓	×	×
			PA3	✓	✓	×	×	×	×
			PD2	✓	✓	×	×	×	×
		TXDB008 (output)	P22	✓	✓	✓	×	×	×
			P94	✓	✓	✓	✓	×	×
			PA3	✓	✓	×	×	×	×
			PD2	✓	✓	×	×	×	×

Table 21.1 Functions Assigned to Each Multiplexed Pin (10/15)

Module/Function	Channel	Pin Functions	Allocation Port	Package					
				With 64 Kbytes of RAM				With 48 Kbytes of RAM	
				100-pin	80-pin	64-pin	48-pin	64-pin	48-pin
Serial communications interface	RSCI8	CTS008# (input) /RTS008# (output) /SS008# (input)	P20	✓	✓	✓	✓	×	×
			P24	✓	×	×	×	×	×
			P30	✓	✓	×	×	×	×
			P96	✓	✓	✓	×	×	×
		DE008 (output)	P20	✓	✓	✓	✓	×	×
			P24	✓	×	×	×	×	×
			P30	✓	✓	×	×	×	×
			P96	✓	✓	✓	×	×	×
	RSCI9	RXD009 (input) /SMISO009 (input/output) /SSCL009 (input/output)	P00	✓	✓	✓	✓	×	×
			PA2	✓	×	×	×	×	×
		TXD009 (output) /TXDA009 (output) /SMOSI009 (input/output) /SSDA009 (input/output)	P01	✓	✓	✓	×	×	×
			P10	✓	✓	×	✓	×	×
			P93	✓	✓	✓	✓	×	×
			P94	✓	✓	✓	✓	×	×
			PA1	✓	×	×	×	×	×
			PA3	✓	✓	×	×	×	×
		SCK009 (input/output)	P11	✓	✓	✓	✓	×	×
			P92	✓	✓	✓	✓	×	×
			PA0	✓	×	×	×	×	×
			PD7	✓	✓	✓	✓	×	×
			PE4	✓	✓	×	×	×	×
			PE5	✓	×	×	×	×	×
		TXDB009 (output)	P11	✓	✓	✓	✓	×	×
			P92	✓	✓	✓	✓	×	×
			PA0	✓	×	×	×	×	×
			PD7	✓	✓	✓	✓	×	×
			PE4	✓	✓	×	×	×	×
			PE5	✓	×	×	×	×	×
		CTS009# (input) /RTS009# (output) /SS009# (input)	P70	✓	✓	✓	×	×	×
			PB3	✓	✓	✓	✓	×	×
			PE3	✓	✓	×	×	×	×
			PE5	✓	×	×	×	×	×
		DE009 (output)	P70	✓	✓	✓	×	×	×
			PB3	✓	✓	✓	✓	×	×
			PE3	✓	✓	×	×	×	×
	RSCI11	RXD011 (input) /SMISO011 (input/output) /SSCL011 (input/output)	P93	✓	✓	✓	✓	×	×
			PA1	✓	×	×	×	×	×
			PB6	✓	✓	✓	✓	×	×
			PD5	✓	✓	✓	✓	×	×
		TXD011 (output) /TXDA011 (output) /SMOSI011 (input/output) /SSDA011 (input/output)	P92	✓	✓	✓	✓	×	×
			PA0	✓	×	×	×	×	×
			PB5	✓	✓	✓	✓	×	×
			PD3	✓	✓	✓	✓	×	×
		SCK011 (input/output)	PB4	✓	✓	✓	✓	×	×
			PB7	✓	×	×	×	×	×
			PD4	✓	✓	✓	×	×	×

Table 21.1 Functions Assigned to Each Multiplexed Pin (11/15)

Module/Function	Channel	Pin Functions	Allocation Port	Package					
				With 64 Kbytes of RAM				With 48 Kbytes of RAM	
				100-pin	80-pin	64-pin	48-pin	64-pin	48-pin
Serial communications interface	RSCI11	TXDB011 (output)	PB4	✓	✓	✓	✓	×	×
			PB7	✓	×	×	×	×	×
			PD4	✓	✓	✓	×	×	×
		CTS011# (input) /RTS011# (output) /SS011# (input)	PB0	✓	✓	✓	✓	×	×
			PB4	✓	✓	✓	✓	×	×
			PD6	✓	✓	✓	×	×	×
		DE011 (output)	PB0	✓	✓	✓	✓	×	×
			PD6	✓	✓	✓	×	×	×
I2C bus interface		SCL0 (input/output)	PB1	✓	✓	✓	✓	✓	✓
		SDA0 (input/output)	PB2	✓	✓	✓	✓	✓	✓
I3C bus interface		SCL00 (input/output)	PB1	✓	✓	✓	✓	×	×
		SDA00 (input/output)	PB2	✓	✓	✓	✓	×	×
CAN FD module	CANFD0	CRX0 (input)	P22	✓	✓	✓	×	✓	×
			P93	✓	✓	✓	✓	✓	✓
			PA1	✓	×	×	×	×	×
			PB4	✓	✓	✓	✓	✓	✓
			PB6	✓	✓	✓	✓	✓	✓
			PE0	✓	×	×	×	×	×
		CTX0 (output)	P23	✓	×	×	×	×	×
			P92	✓	✓	✓	✓	✓	✓
			PA0	✓	×	×	×	×	×
			PB3	✓	✓	✓	✓	✓	✓
			PB5	✓	✓	✓	✓	✓	✓
			PD7	✓	✓	✓	✓	✓	✓
Serial peripheral interface	RSPI0	RSPCKA (input/output)	P20	✓	✓	✓	✓	✓	✓
			P24	✓	×	×	×	×	×
			P27	✓	✓	×	×	×	×
			PA4	✓	×	×	×	×	×
			PB3	✓	✓	✓	✓	✓	✓
			PD0	✓	×	×	×	×	×
		MOSIA (input/output)	P21	✓	✓	✓	✓	✓	✓
			P23	✓	×	×	×	×	×
			PB0	✓	✓	✓	✓	✓	✓
			PD2	✓	✓	×	×	×	×
		MISOA (input/output)	P22	✓	✓	✓	×	✓	×
			P95	✓	✓	✓	✓	✓	✓
			PA5	✓	✓	×	×	×	×
			PB4	✓	✓	✓	✓	✓	✓
			PD1	✓	×	×	×	×	×
		SSLA0 (input/output)	P30	✓	✓	×	×	×	×
			P70	✓	✓	✓	×	✓	×
			P94	✓	✓	✓	✓	✓	✓
			PA3	✓	✓	×	×	×	×
			PD6	✓	✓	✓	×	✓	×
		SSLA1 (output)	P31	✓	✓	×	×	×	×
			PA2	✓	×	×	×	×	×
			PD7	✓	✓	✓	✓	✓	✓

Table 21.1 Functions Assigned to Each Multiplexed Pin (12/15)

Module/Function	Channel	Pin Functions	Allocation Port	Package					
				With 64 Kbytes of RAM				With 48 Kbytes of RAM	
				100-pin	80-pin	64-pin	48-pin	64-pin	48-pin
Serial peripheral interface	RSPI0	SSLA2 (output)	P32	✓	×	×	×	×	×
			P93	✓	✓	✓	✓	✓	✓
			PA1	✓	×	×	×	×	×
			PE0	✓	×	×	×	×	×
		SSLA3 (output)	P33	✓	×	×	×	×	×
			P92	✓	✓	✓	✓	✓	✓
			PA0	✓	×	×	×	×	×
			PE1	✓	×	×	×	×	×
	RSPIA0	RSPCK0 (input/output)	P20	✓	✓	✓	✓	×	×
			P24	✓	×	×	×	×	×
			P27	✓	✓	×	×	×	×
			P70	✓	✓	✓	×	×	×
			P91	✓	✓	✓	✓	×	×
			P96	✓	✓	✓	×	×	×
			PA4	✓	×	×	×	×	×
			PB5	✓	✓	✓	✓	×	×
			PD0	✓	×	×	×	×	×
		MOSI0 (input/output)	P21	✓	✓	✓	✓	×	×
			P23	✓	×	×	×	×	×
			P72	✓	✓	✓	✓	×	×
			P93	✓	✓	✓	✓	×	×
			PB0	✓	✓	✓	✓	×	×
			PD2	✓	✓	×	×	×	×
			PD3	✓	✓	✓	✓	×	×
		MISO0 (input/output)	P22	✓	✓	✓	×	×	×
			P71	✓	✓	✓	✓	×	×
			P92	✓	✓	✓	✓	×	×
			P95	✓	✓	✓	✓	×	×
			PA5	✓	✓	×	×	×	×
			PB6	✓	✓	✓	✓	×	×
			PD1	✓	×	×	×	×	×
		SSL00 (input/output)	P30	✓	✓	×	×	×	×
			P73	✓	✓	✓	✓	×	×
			P94	✓	✓	✓	✓	×	×
			PA3	✓	✓	×	×	×	×
			PD5	✓	✓	✓	✓	×	×
			PD6	✓	✓	✓	×	×	×
		SSL01 (output)	P31	✓	✓	×	×	×	×
			P74	✓	✓	✓	✓	×	×
			P90	✓	✓	✓	×	×	×
			PA2	✓	×	×	×	×	×
			PB4	✓	✓	✓	✓	×	×
			PD7	✓	✓	✓	✓	×	×

Table 21.1 Functions Assigned to Each Multiplexed Pin (13/15)

Module/Function	Channel	Pin Functions	Allocation Port	Package					
				With 64 Kbytes of RAM				With 48 Kbytes of RAM	
				100-pin	80-pin	64-pin	48-pin	64-pin	48-pin
Serial peripheral interface	RSPIA0	SSL02 (output)	P32	✓	×	×	×	×	×
			P75	✓	✓	✓	✓	×	×
			P93	✓	✓	✓	✓	×	×
			P95	✓	✓	✓	✓	×	×
			PA1	✓	×	×	×	×	×
			PD4	✓	✓	✓	×	×	×
			PE0	✓	×	×	×	×	×
		SSL03 (output)	P33	✓	×	×	×	×	×
			P76	✓	✓	✓	✓	×	×
			P92	✓	✓	✓	✓	×	×
			P96	✓	✓	✓	×	×	×
			PA0	✓	×	×	×	×	×
			PB7	✓	×	×	×	×	×
			PE1	✓	×	×	×	×	×
12-bit A/D converter		AN000 (input)*1	P40	✓	✓	✓	✓	✓	✓
		AN001 (input)*1	P41	✓	✓	✓	✓	✓	✓
		AN002 (input)*1	P42	✓	✓	✓	✓	✓	✓
		AN003 (input)*1	P43	✓	✓	✓	✓	✓	✓
		AN004 (input)*1, *2	P44	×	×	×	×	✓	✓
		AN005 (input)*1, *2	P45	×	×	×	×	✓	×
		AN006 (input)*1, *2	P46	×	×	×	×	✓	×
		ADTRG0# (input)	P20	✓	✓	✓	✓	✓	✓
			P93	✓	✓	✓	✓	✓	✓
			PA1	✓	×	×	×	×	×
			PA4	✓	×	×	×	×	×
		ADST0 (output)	PD6	✓	✓	✓	×	✓	×
			PE5	✓	×	×	×	×	×
			PN7	✓	✓	✓	×	✓	×
		AN100 (input)*1	P44	✓	✓	✓	✓	×	×
		AN101 (input)*1	P45	✓	✓	✓	×	×	×
		AN102 (input)*1	P46	✓	✓	✓	×	×	×
		AN103 (input)*1	P47	✓	✓	✓	×	×	×
		ADTRG1# (input)	P21	✓	✓	✓	✓	×	×
			P95	✓	✓	✓	✓	×	×
			PA5	✓	✓	×	×	×	×
		ADST1 (output)	P00	✓	✓	✓	✓	×	×
		AN200 (input)*1	P52	✓	✓	✓	✓	✓	✓
		AN201 (input)*1	P53	✓	✓	✓	✓	✓	✓
		AN202 (input)*1	P54	✓	✓	✓	×	✓	×
		AN203 (input)*1	P55	✓	✓	×	×	×	×
		AN204 (input)*1	P50	✓	✓	×	×	×	×
		AN205 (input)*1	P51	✓	✓	×	×	×	×
		AN206 (input)*1	P47	×	×	×	×	✓	×
			P60	✓	✓	×	×	×	×
		AN207 (input)*1	P61	✓	×	×	×	×	×
		AN208 (input)*1	P62	✓	×	×	✓	×	✓
		AN209 (input)*1	P63	✓	×	×	×	×	×

Table 21.1 Functions Assigned to Each Multiplexed Pin (14/15)

Module/Function	Channel	Pin Functions	Allocation Port	Package					
				With 64 Kbytes of RAM				With 48 Kbytes of RAM	
				100-pin	80-pin	64-pin	48-pin	64-pin	48-pin
12-bit A/D converter		AN210 (input)*1	P64	✓	✓	✓	×	✓	×
		AN211 (input)*1	P65	✓	✓	✓	×	✓	×
		AN216 (input)*1	P20	✓	✓	✓	✓	✓	✓
		AN217 (input)*1	P21	✓	✓	✓	✓	✓	✓
		ADTRG2# (input)	P22	✓	✓	✓	×	✓	×
			PB0	✓	✓	✓	✓	✓	✓
		ADST2 (output)	P01	✓	✓	✓	×	✓	×
12-bit D/A converter		DA0 (output)*1	P64	✓	✓	✓	×	✓	×
		DA1 (output)*1	P65	✓	✓	✓	×	✓	×
Clock frequency accuracy measurement circuit		CACREF (input)	P00	✓	✓	✓	✓	✓	✓
			P23	✓	×	×	×	×	×
			PB3	✓	✓	✓	✓	✓	✓
Comparator		COMP0 (output)	P00	✓	✓	✓	✓	✓	✓
			P24	✓	×	×	×	×	×
		COMP1 (output)	P01	✓	✓	✓	×	✓	×
			P23	✓	×	×	×	×	×
		COMP2 (output)	P22	✓	✓	✓	×	✓	×
		COMP3 (output)	P30	✓	✓	×	×	×	×
			P80	✓	×	×	×	×	×
		COMP4 (output)	P20	✓	✓	✓	✓	✓	✓
			P81	✓	×	×	×	×	×
		COMP5 (output)	P21	✓	✓	✓	✓	✓	✓
			P82	✓	×	×	×	×	×
		CVREFC0 (input)*1	P53	✓	✓	✓	✓	✓	✓
		CVREFC1 (input)*1	P54	✓	✓	✓	×	✓	×
		CMPC00 (input)*1	P40	✓	✓	✓	✓	✓	✓
		CMPC01 (input)*1	P40	✓	✓	✓	✓	×	×
			P44	×	×	×	×	✓	✓
		CMPC02 (input)*1	P52	✓	✓	✓	✓	✓	✓
		CMPC03 (input)*1	P47	×	×	×	×	✓	×
			P60	✓	✓	×	×	×	×
		CMPC10 (input)*1	P41	✓	✓	✓	✓	✓	✓
		CMPC11 (input)*1	P41	✓	✓	✓	✓	×	×
			P45	×	×	×	×	✓	×
		CMPC12 (input)*1	P53	✓	✓	✓	✓	✓	✓
		CMPC13 (input)*1	P40	×	×	×	×	✓	✓
			P61	✓	×	×	×	×	×
		CMPC20 (input)*1	P42	✓	✓	✓	✓	✓	✓
		CMPC21 (input)*1	P42	✓	✓	✓	✓	×	×
			P46	×	×	×	×	✓	×
		CMPC22 (input)*1	P54	✓	✓	✓	×	✓	×
		CMPC23 (input)*1	P43	×	×	×	×	✓	✓
			P63	✓	×	×	×	×	×
		CMPC30 (input)*1	P44	✓	✓	✓	✓	×	×
		CMPC31 (input)*1		✓	✓	✓	✓	×	×
		CMPC32 (input)*1	P55	✓	✓	×	×	×	×
		CMPC33 (input)*1	P64	✓	✓	✓	×	×	×

Table 21.1 Functions Assigned to Each Multiplexed Pin (15/15)

Module/Function	Channel	Pin Functions	Allocation Port	Package					
				With 64 Kbytes of RAM				With 48 Kbytes of RAM	
				100-pin	80-pin	64-pin	48-pin	64-pin	48-pin
Comparator		CMPC40 (input)*1	P45	✓	✓	✓	×	×	×
		CMPC41 (input)*1		✓	✓	✓	×	×	×
		CMPC42 (input)*1	P50	✓	✓	×	×	×	×
		CMPC43 (input)*1	P62	✓	×	×	✓	×	×
		CMPC50 (input)*1	P43	×	×	×	×	✓	✓
			P46	✓	✓	✓	×	×	×
		CMPC51 (input)*1	P46	✓	✓	✓	×	×	×
			P62	×	×	×	×	×	✓
		CMPC52 (input)*1	P51	✓	✓	×	×	×	×
			P64	×	×	×	×	✓	×
		CMPC53 (input)*1	P65	✓	✓	✓	×	✓	×

Note 1. To use this pin function, set the corresponding pin as general input (set the PORTm.PDR.Bn and PORTm.PMR.Bn bits to 0).
Note 2. Products with 64 Kbytes of RAM do not have this pin function.

21.2 Register Descriptions

The registers and bits of unsupported pins, depending on the package, are reserved. The write value to the reserved bits is the value after a reset.

21.2.1 Write-Protect Register (PWPR)

Address(es): 0008 C11Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	B0WI	PFSWE	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	PFSWE	PFS Register Write Enable	0: Writing to the PFS register is disabled 1: Writing to the PFS register is enabled	R/W
b7	B0WI	PFSWE Bit Write Disable	0: Writing to the PFSWE bit is enabled 1: Writing to the PFSWE bit is disabled	R/W

PFSWE Bit (PFS Register Write Enable)

Writing to PmnPFS register is enabled only when the PFSWE bit is set to 1.
To set the PFSWE bit to 1, write 1 to the PFSWE bit after writing 0 to the B0WI bit.

B0WI Bit (PFSWE Bit Write Disable)

Writing to the PFSWE bit is enabled only when the B0WI bit is set to 0.

21.2.2 P0n Pin Function Control Register (P0nPFS) (n = 0, 1)

Address(es): P00PFS 0008 C140h, P01PFS 0008 C141h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	ISEL	PSEL[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PSEL[5:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 21.2.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin P00: IRQ2 (48/64/80/100 pin) P01: IRQ4 (64/80/100 pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

The port mn pin function control register (PmnPFS) selects the pin function. Bits PSEL[5:0] select the peripheral function which is assigned to bits.
The ISEL bit is set when a pin is used as an IRQ input pin. This setting can be used with the combination of the peripheral function, though IRQn (external pin interrupt) of the same number should not be enabled by two or more pins.
The ASEL bit is set when a pin is used as an analog pin. When the pin is set as an analog pin by the ASEL bit, select the general I/O port by the port mode register (PORTm.PMR) and specify input by the port direction register (PORTm.PDR). The pin state cannot be read at this point, since the PmnPFS register is protected by the write-protect register (PWPR). Modify the register after releasing the protection.
The ISEL bit to which IRQn is not specified is reserved. The ASEL bit to which analog input/output is not specified is reserved.

Table 21.2 Register Settings for Input/Output Pin Function in 100-/80-/64-/48-Pin Products

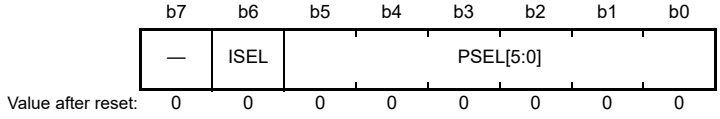
	Pin	
PSEL[5:0] Settings	**P00**	**P01*1**
000000b (initial value)	Hi-Z	
000001b	MTIOC9A	MTIOC9C
000011b	MTIOC9A#	MTIOC9C#
000111b	CACREF	POE12#
001001b	ADST1*2	ADST2
001100b	RXD12 SMISO12 SSCL12 RXDX12	TXD12 SMOSI12 SSDA12 TXDX12 SIOX12
010100b	—	GTETRGA
010101b	—	GTETRGB
010110b	—	GTETRGC
010111b	—	GTETRGD
011000b	GTIU	GTIW
011101b	TIC3	—
011110b	COMP0	COMP1
101100b*3	RXD009 SMISO009 SSCL009	TXD009 TXDA009 SMOSI009 SSDA009

—: Do not specify this value.

Note 1. P01 is not present in the 48-pin products.
Note 2. Products with 48 Kbytes of RAM do not have this pin function.
Note 3. Do not specify this value on the products with 48 Kbytes of RAM.

21.2.3 P1n Pin Function Control Registers (P1nPFS) (n = 0, 1)

Address(es): P10PFS 0008 C148h, P11PFS 0008 C149h



Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PSEL[5:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 21.3.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin P10: IRQ0 (48/80/100 pin) P11: IRQ1 (48/64/80/100 pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 21.3 Register Settings for Input/Output Pin Function in 100-/80-/64-/48-Pin Products

	Pin	
PSEL[5:0] Settings	**P10*1**	**P11**
000000b (initial value)	Hi-Z	
000001b	MTIOC9B	MTIOC3A
000010b	MTCLKD	MTCLKC
000011b	MTIOC9B#	MTIOC3A#
000100b	MTCLKD#	MTCLKC#
000101b	TMRI3	TMO3
000111b	POE12#	POE9#
001000b	—	MTIOC9D
001010b	CTS6# RTS6# SS6#	—
010100b	GTIOC3A	GTIOC3B
010101b	GTETRGB	GTETRGA
010110b	GTIOC3A#	GTIOC3B#
010111b	GTETRGD	GTETRGC
011000b	GTIV	GTCPPO0
011101b	TIC3	TOC3
101100b*2	TXD009 TXDA009 SMOSI009 SSDA009	SCK009
101101b*2	—	SCK008
101110b*2	—	TXDB009

—: Do not specify this value.
Note 1. P10 is not present in the 64-pin products.
Note 2. Do not specify this value on the products with 48 Kbytes of RAM.

21.2.4 P2n Pin Function Control Registers (P2nPFS) (n = 0 to 4, 7)

Address(es): P20PFS 0008 C150h, P21PFS 0008 C151h, P22PFS 0008 C152h, P23PFS 0008 C153h, P24PFS 0008 C154h, P27PFS 0008 C157h

	b7	b6	b5	b4	b3	b2	b1	b0
	ASEL	ISEL	PSEL[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PSEL[5:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 21.4 to Table 21.6.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin P20: IRQ7 (48/64/80/100 pin) P21: IRQ6 (48/64/80/100 pin) P22: IRQ10 (64/80/100 pin) P23: IRQ11 (100 pin) P24: IRQ4 (100 pin) P27: IRQ15 (80/100 pin)	R/W
b7	ASEL	Analog Function Select	0: Used other than as analog pin. 1: Used as analog pin. P20: AN216 (48/64/80/100 pin) P21: AN217 (48/64/80/100 pin)	R/W

Table 21.4 Register Settings for Input/Output Pin Function in 100-Pin Products

	Pin					
PSEL[5:0] Settings	**P20**	**P21**	**P22**	**P23**	**P24**	**P27**
000000b (initial value)	Hi-Z					
000001b	MTIOC9C	MTIOC9A	MTIC5W	MTIC5V	MTIC5U	MTIOC1A
000010b	MTCLKB	MTCLKA	MTCLKD	—	—	MTIOC0C
000011b	MTIOC9C#	MTIOC9A#	MTIC5W#	MTIC5V#	MTIC5U#	MTIOC1A#
000100b	MTCLKB#	MTCLKA#	MTCLKD#	—	—	MTIOC0C#
000101b	TMRI4	TMCI4	TMRI2	TMO2	TMCI2	TMO2
000110b	TMO2	TMO6	TMO4	—	TMO6	TMO6
000111b	—	—	—	CACREF	—	POE9#
001000b	—	—	MTIOC9B	—	—	—
001001b	ADTRG0#	ADTRG1#	ADTRG2#	—	—	—
001100b	—	TXD12 SMOSI12 SSDA12 TXDX12 SIOX12	RXD12 SMISO12 SSCL12 RXDX12	TXD12 SMOSI12 SSDA12 TXDX12 SIOX12	—	—
001101b	RSPCKA	MOSIA	MISOA	MOSIA	RSPCKA	RSPCKA
001110b	RSPCK0	MOSI0	MISO0	MOSI0	RSPCK0	RSPCK0
010000b	—	—	CRX0	CTX0	—	—
011000b	GTIW	GTIU	GTIV	—	—	—
011110b	COMP4	COMP5	COMP2	COMP1	COMP0	—
101100b	CTS008# RTS008# SS008#	TXD008 TXDA008 SMOSI008 SSDA008	RXD008 SMISO008 SSCL008	TXD008 TXDA008 SMOSI008 SSDA008	CTS008# RTS008# SS008#	—
101101b	RXD008 SMISO008 SSCL008	—	SCK008	—	SCK008	—
101110b	DE008	—	TXDB008	—	DE008	—

—: Do not specify this value.

Table 21.5 Register Settings for Input/Output Pin Function in 80-Pin Products

	Pin			
PSEL[5:0] Settings	**P20**	**P21**	**P22**	**P27**
000000b (initial value)	Hi-Z			
000001b	MTIOC9C	MTIOC9A	MTIC5W	MTIOC1A
000010b	MTCLKB	MTCLKA	MTCLKD	MTIOC0C
000011b	MTIOC9C#	MTIOC9A#	MTIC5W#	MTIOC1A#
000100b	MTCLKB#	MTCLKA#	MTCLKD#	MTIOC0C#
000101b	TMRI4	TMCI4	TMRI2	TMO2
000110b	TMO2	TMO6	TMO4	TMO6
000111b	—	—	—	POE9#
001000b	—	—	MTIOC9B	—
001001b	ADTRG0#	ADTRG1#	ADTRG2#	—
001100b	—	TXD12 SMOSI12 SSDA12 TXDX12 SIOX12	RXD12 SMISO12 SSCL12 RXDX12	—
001101b	RSPCKA	MOSIA	MISOA	RSPCKA
001110b	RSPCK0	MOSI0	MISO0	RSPCK0
010000b	—	—	CRX0	—
011000b	GTIW	GTIU	GTIV	—
011110b	COMP4	COMP5	COMP2	—
101100b	CTS008# RTS008# SS008#	TXD008 TXDA008 SMOSI008 SSDA008	RXD008 SMISO008 SSCL008	—
101101b	RXD008 SMISO008 SSCL008	—	SCK008	—
101110b	DE008	—	TXDB008	—

—: Do not specify this value.

Table 21.6 Register Settings for Input/Output Pin Function in 64-/48-Pin Products

	Pin		
PSEL[5:0] Settings	**P20**	**P21**	**P22*1**
000000b (initial value)	Hi-Z		
000001b	MTIOC9C	MTIOC9A	MTIC5W
000010b	MTCLKB	MTCLKA	MTCLKD
000011b	MTIOC9C#	MTIOC9A#	MTIC5W#
000100b	MTCLKB#	MTCLKA#	MTCLKD#
000101b	TMRI4	TMCI4	TMRI2
000110b	TMO2	TMO6	TMO4
001000b	—	—	MTIOC9B
001001b	ADTRG0#	ADTRG1#*2	ADTRG2#
001100b	—	TXD12 SMOSI12 SSDA12 TXDX12 SIOX12	RXD12 SMISO12 SSCL12 RXDX12
001101b	RSPCKA	MOSIA	MISOA
001110b*3	RSPCK0	MOSI0	MISO0
010000b	—	—	CRX0
011000b	GTIW	GTIU	GTIV
011110b	COMP4	COMP5	COMP2
101100b*3	CTS008# RTS008# SS008#	TXD008 TXDA008 SMOSI008 SSDA008	RXD008 SMISO008 SSCL008
101101b*3	RXD008 SMISO008 SSCL008	—	SCK008
101110b*3	DE008	—	TXDB008

Note 1. P22 is not present in the 48-pin products.
Note 2. Products with 48 Kbytes of RAM do not have this pin function.
Note 3. Do not specify this value on the products with 48 Kbytes of RAM.

21.2.5 P3n Pin Function Control Registers (P3nPFS) (n = 0 to 3, 6, 7)

Address(es): P30PFS 0008 C158h, P31PFS 0008 C159h, P32PFS 0008 C15Ah, P33PFS 0008 C15Bh, P36PFS 0008 C15Eh, P37PFS 0008 C15Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	ISEL	PSEL[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PSEL[5:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 21.7 to Table 21.9.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin P30: IRQ7 (80/100 pin) P31: IRQ6 (80/100 pin) P32: IRQ12 (100 pin) P33: IRQ13 (100 pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 21.7 Register Settings for Input/Output Pin Function in 100-Pin Products

	Pin					
PSEL[5:0] Settings	**P30**	**P31**	**P32**	**P33**	**P36**	**P37**
000000b (initial value)	Hi-Z					
000001b	MTIOC0B	MTIOC0A	MTIOC3C	MTIOC3A	—	—
000010b	MTCLKD	MTCLKC	MTCLKB	MTCLKA	—	—
000011b	MTIOC0B#	MTIOC0A#	MTIOC3C#	MTIOC3A#	—	—
000100b	MTCLKD#	MTCLKC#	MTCLKB#	MTCLKA#	—	—
000101b	TMCI6	TMRI6	TMO6	TMO0	—	—
001010b	—	—	—	—	TXD5 SMOSI5 SSDA5	RXD5 SMISO5 SSCL5
001101b	SSLA0	SSLA1	SSLA2	SSLA3	—	—
001110b	SSL00	SSL01	SSL02	SSL03	—	—
010100b	—	—	GTIOC3A	GTIOC3B	—	—
010101b	—	—	GTIOC7A	GTIOC7B	—	—
010110b	—	—	GTIOC3A#	GTIOC3B#	—	—
010111b	—	—	GTIOC7A#	GTIOC7B#	—	—
011000b	GTIV	GTIU	—	GTCPPO0	—	—
011110b	COMP3	—	—	—	—	—
101100b	SCK008	—	—	—	—	—
101101b	CTS008# RTS008# SS008#	—	—	—	—	—
101110b	DE008	—	—	—	—	—

—: Do not specify this value.

Table 21.8 Register Settings for Input/Output Pin Function in 80-Pin Products

	Pin			
PSEL[5:0] Settings	**P30**	**P31**	**P36**	**P37**
000000b (initial value)	Hi-Z			
000001b	MTIOC0B	MTIOC0A	—	—
000010b	MTCLKD	MTCLKC	—	—
000011b	MTIOC0B#	MTIOC0A#	—	—
000100b	MTCLKD#	MTCLKC#	—	—
000101b	TMCI6	TMRI6	—	—
001010b	—	—	TXD5 SMOSI5 SSDA5	RXD5 SMISO5 SSCL5
001101b	SSLA0	SSLA1	—	—
001110b	SSL00	SSL01	—	—
011000b	GTIV	GTIU	—	—
011110b	COMP3	—	—	—
101100b	SCK008	—	—	—
101101b	CTS008# RTS008# SS008#	—	—	—
101110b	DE008	—	—	—

—: Do not specify this value.

Table 21.9 Register Settings for Input/Output Pin Function in 64-/48-Pin Products

	Pin	
PSEL[5:0] Settings	**P36**	**P37**
000000b (initial value)	Hi-Z	
001010b	TXD5 SMOSI5 SSDA5	RXD5 SMISO5 SSCL5

21.2.6 P4n Pin Function Control Registers (P4nPFS) (n = 0 to 7)

Address(es): P40PFS 0008 C160h, P41PFS 0008 C161h, P42PFS 0008 C162h, P43PFS 0008 C163h, P44PFS 0008 C164h, P45PFS 0008 C165h, P46PFS 0008 C166h, P47PFS 0008 C167h

	b7	b6	b5	b4	b3	b2	b1	b0
	ASEL	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	ASEL	Analog Function Select	0: Used other than as analog pin. 1: Used as analog pin. • Products with 64 Kbytes of RAM P40: AN000, CMPC00, CMPC01 (48/64/80/100 pin) P41: AN001, CMPC10, CMPC11 (48/64/80/100 pin) P42: AN002, CMPC20, CMPC21 (48/64/80/100 pin) P43: AN003 (48/64/80/100 pin) P44: AN100, CMPC30, CMPC31 (48/64/80/100 pin) P45: AN101, CMPC40, CMPC41 (64/80/100 pin) P46: AN102, CMPC50, CMPC51 (64/80/100 pin) P47: AN103 (64/80/100 pin) • Products with 48 Kbytes of RAM P40: AN000, CMPC00, CMPC13 (48/64 pin) P41: AN001, CMPC10 (48/64 pin) P42: AN002, CMPC20 (48/64 pin) P43: AN003, CMPC23, CMPC50 (48/64 pin) P44: AN004, CMPC01 (48/64 pin) P45: AN005, CMPC11 (64 pin) P46: AN006, CMPC21 (64 pin) P47: AN206, CMPC03 (64 pin)	R/W

21.2.7 P5n Pin Function Control Registers (P5nPFS) (n = 0 to 5)

Address(es): P50PFS 0008 C168h, P51PFS 0008 C169h, P52PFS 0008 C16Ah, P53PFS 0008 C16Bh, P54PFS 0008 C16Ch, P55PFS 0008 C16Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	ASEL	ISEL	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin P52: IRQ0 (48/64/80/100 pin) P53: IRQ1 (48/64/80/100 pin) P54: IRQ2 (64/80/100 pin) P55: IRQ3 (80/100 pin)	R/W
b7	ASEL	Analog Function Select	0: Used other than as analog pin. 1: Used as analog pin. P50: AN204, CMPC42 (80/100 pin) P51: AN205, CMPC52 (80/100 pin) P52: AN200, CMPC02 (48/64/80/100 pin) P53: AN201, CMPC12 (48/64/80/100 pin) P54: AN202, CMPC22 (64/80/100 pin) P55: AN203, CMPC32 (80/100 pin)	R/W

21.2.8 P6n Pin Function Control Registers (P6nPFS) (n = 0 to 5)

Address(es): P60PFS 0008 C170h, P61PFS 0008 C171h, P62PFS 0008 C172h, P63PFS 0008 C173h, P64PFS 0008 C174h, P65PFS 0008 C175h

	b7	b6	b5	b4	b3	b2	b1	b0
	ASEL	ISEL	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin P60: IRQ4 (80/100 pin) P61: IRQ5 (100 pin) P62: IRQ6 (48/100 pin) P63: IRQ7 (100 pin) P64: IRQ8 (64/80/100 pin) P65: IRQ9 (64/80/100 pin)	R/W
b7	ASEL	Analog Function Select	0: Used other than as analog pin. 1: Used as analog pin. • Products with 64 Kbytes of RAM P60: AN206, CMPC03 (80/100 pin) P61: AN207, CMPC13 (100 pin) P62: AN208, CMPC43 (48/100 pin) P63: AN209, CMPC23 (100 pin) P64: AN210, CMPC33, DA0 (64/80/100 pin) P65: AN211, CMPC53, DA1 (64/80/100 pin) • Products with 48 Kbytes of RAM P62: AN208, CMPC51 (48 pin) P64: AN210, CMPC52, DA0 (64 pin) P65: AN211, CMPC53, DA1 (64 pin)	R/W

21.2.9 P7n Pin Function Control Registers (P7nPFS) (n = 0 to 6)

Address(es): P70PFS 0008 C178h, P71PFS 0008 C179h, P72PFS 0008 C17Ah, P73PFS 0008 C17Bh, P74PFS 0008 C17Ch, P75PFS 0008 C17Dh, P76PFS 0008 C17Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	ISEL	PSEL[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PSEL[5:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 21.10.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin P70: IRQ5 (64/80/100 pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 21.10 Register Settings for Input/Output Pin Function in 100-/80-/64-/48-Pin Products

	Pin						
PSEL[5:0] Settings	**P70*1**	**P71**	**P72**	**P73**	**P74**	**P75**	**P76**
000000b (initial value)	Hi-Z						
000001b	MTIOC0A	MTIOC3B	MTIOC4A	MTIOC4B	MTIOC3D	MTIOC4C	MTIOC4D
000010b	MTCLKC	—	—	—	—	—	—
000011b	MTIOC0A#	MTIOC3B#	MTIOC4A#	MTIOC4B#	MTIOC3D#	MTIOC4C#	MTIOC4D#
000100b	MTCLKC#	—	—	—	—	—	—
000101b	TMRI6	—	—	—	—	—	—
000111b	POE0#	—	—	—	—	—	—
001010b	SCK5	—	—	—	—	—	—
001101b	SSLA0	—	—	—	—	—	—
001110b*2	RSPCK0	MISO0	MOSI0	SSL00	SSL01	SSL02	SSL03
010100b	GTETRGA	GTIOC0A	GTIOC1A	GTIOC2A	GTIOC0B	GTIOC1B	GTIOC2B
010101b	GTETRGB	GTIOC4A	GTIOC5A	GTIOC6A	GTIOC4B	GTIOC5B	GTIOC6B
010110b	GTETRGC	GTIOC0A#	GTIOC1A#	GTIOC2A#	GTIOC0B#	GTIOC1B#	GTIOC2B#
010111b	GTETRGD	GTIOC4A#	GTIOC5A#	GTIOC6A#	GTIOC4B#	GTIOC5B#	GTIOC6B#
011000b	GTCPPO0	GTOUUP	GTOVUP	GTOWUP	GTOULO	GTOVLO	GTOWLO
101100b*2	CTS009# RTS009# SS009#	—	—	—	—	—	—
101110b*2	DE009	—	—	—	—	—	—

—: Do not specify this value.

Note 1. P70 is not present in the 48-pin products.

Note 2. Do not specify this value on the products with 48 Kbytes of RAM.

21.2.10 P8n Pin Function Control Registers (P8nPFS) (n = 0 to 2)

Address(es): P80PFS 0008 C180h, P81PFS 0008 C181h, P82PFS 0008 C182h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	ISEL	PSEL[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PSEL[5:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 21.11.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin P80: IRQ5 (100 pin) P82: IRQ3 (100 pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 21.11 Register Settings for Input/Output Pin Function in 100-Pin Products

	Pin		
PSEL[5:0] Settings	**P80**	**P81**	**P82**
000000b (initial value)	Hi-Z		
000001b	MTIC5W	MTIC5V	MTIC5U
000011b	MTIC5W#	MTIC5V#	MTIC5U#
000101b	TMRI4	TMCI4	TMO4
001010b	RXD6 SMISO6 SSCL6	TXD6 SMOSI6 SSDA6	SCK6
001100b	RXD12 SMISO12 SSCL12 RXDX12	TXD12 SMOSI12 SSDA12 TXDX12 SIOX12	SCK12
011110b	COMP3	COMP4	COMP5

21.2.11 P9n Pin Function Control Registers (P9nPFS) (n = 0 to 6)

Address(es): P90PFS 0008 C188h, P91PFS 0008 C189h, P92PFS 0008 C18Ah, P93PFS 0008 C18Bh, P94PFS 0008 C18Ch, P95PFS 0008 C18Dh, P96PFS 0008 C18Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	ISEL	PSEL[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PSEL[5:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 21.12.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin P93: IRQ14 (48/64/80/100 pin) P95: IRQ1 (48/64/80/100 pin) P96: IRQ4 (64/80/100 pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 21.12 Register Settings for Input/Output Pin Function in 100-/80-/64-/48-Pin Products

	Pin						
PSEL[5:0] Settings	**P90*1**	**P91**	**P92**	**P93**	**P94**	**P95**	**P96*1**
000000b (initial value)	Hi-Z						
000001b	MTIOC7D	MTIOC7C	MTIOC6D	MTIOC7B	MTIOC7A	MTIOC6B	—
000010b	—	—	MTIOC6C	MTIOC6A	MTIOC2A	MTIOC1A	—
000011b	MTIOC7D#	MTIOC7C#	MTIOC6D#	MTIOC7B#	MTIOC7A#	MTIOC6B#	—
000100b	—	—	MTIOC6C#	MTIOC6A#	MTIOC2A#	MTIOC1A#	—
000101b	—	—	TMO2	TMO4	TMRI7	TMCI3	—
000111b	—	—	—	—	—	—	POE4#
001001b	—	—	—	ADTRG0#	—	ADTRG1#*2	—
001010b	TXD5 SMOSI5 SSDA5	RXD5 SMISO5 SSCL5	—	—	—	RXD6 SMISO6 SSCL6	—
001101b	—	—	SSLA3	SSLA2	SSLA0	MISOA	—
001110b*3	SSL01	RSPCK0	MISO0	MOSI0	SSL00	SSL02	SSL03
010000b	—	—	CTX0	CRX0	—	—	—
010100b	GTIOC6B	GTIOC5B	GTIOC4B	GTIOC6A	GTIOC5A	GTIOC4A	GTETRGA
010101b	—	—	GTIOC7B	—	GTADSM0	GTIOC7A	GTETRGB
010110b	GTIOC6B#	GTIOC5B#	GTIOC4B#	GTIOC6A#	GTIOC5A#	GTIOC4A#	GTETRGC
010111b	—	—	GTIOC7B#	—	—	GTIOC7A#	GTETRGD
011000b	GTOWLO	GTOVLO	GTOULO	GTOWUP	GTOVUP	GTOUUP	GTCPPO4
101100b*3	—	—	SCK009	TXD009 TXDA009 SMOSI009 SSDA009	TXD009 TXDA009 SMOSI009 SSDA009	—	CTS008# RTS008# SS008#
101101b*3	—	—	TXD011 TXDA011 SMOSI011 SSDA011	RXD011 SMISO011 SSCL011	SCK008	RXD008 SMISO008 SSCL008	—
101110b*3	—	—	TXDB009	—	TXDB008	—	DE008
110011b*3	—	—	SSL03	SSL02	SSL00	MISO0	RSPCK0

—: Do not specify this value.

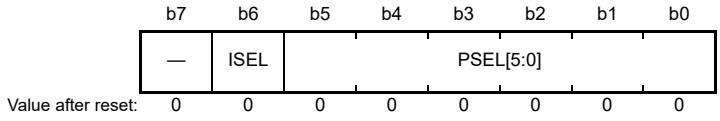
Note 1. P90 and P96 are not present in the 48-pin products.

Note 2. Products with 48 Kbytes of RAM do not have this pin function.

Note 3. Do not specify this value on the products with 48 Kbytes of RAM.

21.2.12 PAn Pin Function Control Registers (PAnPFS) (n = 0 to 5)

Address(es): PA0PFS 0008 C190h, PA1PFS 0008 C191h, PA2PFS 0008 C192h, PA3PFS 0008 C193h, PA4PFS 0008 C194h, PA5PFS 0008 C195h



Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PSEL[5:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 21.13 and Table 21.14.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin PA1: IRQ14 (100/ pin) PA5: IRQ1 (80/100 pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 21.13 Register Settings for Input/Output Pin Function in 100-Pin Products

	Pin					
PSEL[5:0] Settings	**PA0**	**PA1**	**PA2**	**PA3**	**PA4**	**PA5**
000000b (initial value)	Hi-Z					
000001b	MTIOC6C	MTIOC6A	MTIOC2B	MTIOC2A	MTIOC1B	MTIOC1A
000011b	MTIOC6C#	MTIOC6A#	MTIOC2B#	MTIOC2A#	MTIOC1B#	MTIOC1A#
000101b	TMO2	TMO4	TMO7	TMRI7	TMCI7	TMCI3
001001b	—	ADTRG0#	—	—	ADTRG0#	ADTRG1#
001010b	—	—	CTS6# RTS6# SS6#	—	SCK6	RXD6 SMISO6 SSCL6
001101b	SSLA3	SSLA2	SSLA1	SSLA0	RSPCKA	MISOA
001110b	SSL03	SSL02	SSL01	SSL00	RSPCK0	MISO0
010000b	CTX0	CRX0	—	—	—	—
010100b	—	—	GTADSM1	GTADSM0	—	—
011000b	—	GTCPPO4	—	—	—	—
101100b	SCK009	TXD009 TXDA009 SMOSI009 SSDA009	—	TXD009 TXDA009 SMOSI009 SSDA009	—	—
101101b	TXD011 TXDA011 SMOSI011 SSDA011	RXD011 SMISO011 SSCL011	RXD009 SMISO009 SSCL009	SCK008	TXD008 TXDA008 SMOSI008 SSDA008	RXD008 SMISO008 SSCL008
101110b	TXDB009	—	—	TXDB008	—	—

—: Do not specify this value.

Table 21.14 Register Settings for Input/Output Pin Function in 80-Pin Products

	Pin	
PSEL[5:0] Settings	**PA3**	**PA5**
000000b (initial value)	Hi-Z	
000001b	MTIOC2A	MTIOC1A
000011b	MTIOC2A#	MTIOC1A#
000101b	TMRI7	TMCI3
001001b	—	ADTRG1#
001010b	—	RXD6 SMISO6 SSCL6
001101b	SSLA0	MISOA
001110b	SSL00	MISO0
010100b	GTADSM0	—
101100b	TXD009 TXDA009 SMOSI009 SSDA009	—
101101b	SCK008	RXD008 SMISO008 SSCL008
101110b	TXDB008	—

—: Do not specify this value.

21.2.13 PBn Pin Function Control Registers (PBnPFS) (n = 0 to 7)

Address(es): PB0PFS 0008 C198h, PB1PFS 0008 C199h, PB2PFS 0008 C19Ah, PB3PFS 0008 C19Bh, PB4PFS 0008 C19Ch, PB5PFS 0008 C19Dh, PB6PFS 0008 C19Eh, PB7PFS 0008 C19Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	ISEL	PSEL[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PSEL[5:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 21.15 and Table 21.16.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin PB0: IRQ8 (48/64/80/100 pin) PB1: IRQ4 (48/64/80/100 pin) PB3: IRQ9 (48/64/80/100 pin) PB4: IRQ3 (48/64/80/100 pin) PB6: IRQ2 (48/64/80/100 pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 21.15 Register Settings for Input/Output Pin Function in 100-Pin Products

	Pin							
PSEL[5:0] Settings	**PB0**	**PB1**	**PB2**	**PB3**	**PB4**	**PB5**	**PB6**	**PB7**
000000b (initial value)	Hi-Z							
000001b	MTIOC0D	MTIOC0C	MTIOC0B	MTIOC0A	—	—	—	—
000011b	MTIOC0D#	MTIOC0C#	MTIOC0B#	MTIOC0A#	—	—	—	—
000101b	TMO0	TMCI0	TMRI0	—	—	—	—	—
000111b	—	—	—	CACREF	POE8#	—	—	—
001001b	ADTRG2#	ADSM1	ADSM0	—	—	—	—	—
001010b	TXD6 SMOSI6 SSDA6	RXD6 SMISO6 SSCL6	TXD6 SMOSI6 SSDA6	SCK6	CTS5# RTS5# SS5#	TXD5 SMOSI5 SSDA5	RXD5 SMISO5 SSCL5	SCK5
001100b	—	—	—	TXD12 SMOSI12 SSDA12 TXDX12 SIOX12	RXD12 SMISO12 SSCL12 RXDX12	TXD12 SMOSI12 SSDA12 TXDX12 SIOX12	RXD12 SMISO12 SSCL12 RXDX12	SCK12
001101b	MOSIA	—	—	RSPCKA	MISOA	—	—	—
001110b	MOSI0	—	—	—	SSL01	RSPCK0	MISO0	SSL03
001111b	—	SCL0	SDA0	—	—	—	—	—
010000b	—	—	—	CTX0	CRX0	CTX0	CRX0	—
010100b	—	GTADSM1	GTADSM0	—	GTETRGA	GTIOC2B	GTIOC2A	GTIOC1B
010101b	—	GTIOC7B	GTIOC7A	—	GTETRGB	GTIOC3B	GTIOC3A	—
010110b	—	—	—	—	GTETRGC	GTIOC2B#	GTIOC2A#	GTIOC1B#
010111b	—	GTIOC7B#	GTIOC7A#	—	GTETRGD	GTIOC3B#	GTIOC3A#	—
011000b	—	GTIW	GTIV	GTIU	GTCPPO0	—	—	—
011101b	TIC2	TOC2	TIC1	TOC1	—	TIC0	TOC0	—
101100b	TXD008 TXDA008 SMOSI008 SSDA008	—	—	CTS009# RTS009# SS009#	CTS011# RTS011# SS011#	—	—	—
101101b	CTS011# RTS011# SS011#	—	—	—	SCK011	TXD011 TXDA011 SMOSI011 SSDA011	RXD011 SMISO011 SSCL011	SCK011
101110b	DE011	—	—	DE009	TXDB011	—	—	TXDB011
110010b	—	SCL00	SDA00	—	—	—	—	—

—: Do not specify this value.

Table 21.16 Register Settings for Input/Output Pin Function in 80-/64-/48-Pin Products

	Pin						
PSEL[5:0] Settings	**PB0**	**PB1**	**PB2**	**PB3**	**PB4**	**PB5**	**PB6**
000000b (initial value)	Hi-Z						
000001b	MTIOC0D	MTIOC0C	MTIOC0B	MTIOC0A	—	—	—
000011b	MTIOC0D#	MTIOC0C#	MTIOC0B#	MTIOC0A#	—	—	—
000101b	TMO0	TMCI0	TMRI0	—	—	—	—
000111b	—	—	—	CACREF	POE8#	—	—
001001b	ADTRG2#	ADSM1	ADSM0	—	—	—	—
001010b	TXD6 SMOSI6 SSDA6	RXD6 SMISO6 SSCL6	TXD6 SMOSI6 SSDA6	SCK6	CTS5# RTS5# SS5#	TXD5 SMOSI5 SSDA5	RXD5 SMISO5 SSCL5
001100b	—	—	—	TXD12 SMOSI12 SSDA12 TXDX12 SIOX12	RXD12 SMISO12 SSCL12 RXDX12	TXD12 SMOSI12 SSDA12 TXDX12 SIOX12	RXD12 SMISO12 SSCL12 RXDX12
001101b	MOSIA	—	—	RSPCKA	MISOA	—	—
001110b*1	MOSI0	—	—	—	SSL01	RSPCK0	MISO0
001111b	—	SCL0	SDA0	—	—	—	—
010000b	—	—	—	CTX0	CRX0	CTX0	CRX0
010100b	—	GTADSM1	GTADSM0	—	GTETRGA	GTIOC2B	GTIOC2A
010101b	—	GTIOC7B	GTIOC7A	—	GTETRGB	GTIOC3B	GTIOC3A
010110b	—	—	—	—	GTETRGC	GTIOC2B#	GTIOC2A#
010111b	—	GTIOC7B#	GTIOC7A#	—	GTETRGD	GTIOC3B#	GTIOC3A#
011000b	—	GTIW	GTIV	GTIU	GTCPPO0	—	—
011101b	TIC2	TOC2	TIC1	TOC1	—	TIC0	TOC0
101100b*1	TXD008 TXDA008 SMOSI008 SSDA008	—	—	CTS009# RTS009# SS009#	CTS011# RTS011# SS011#	—	—
101101b*1	CTS011# RTS011# SS011#	—	—	—	SCK011	TXD011 TXDA011 SMOSI011 SSDA011	RXD011 SMISO011 SSCL011
101110b*1	DE011	—	—	DE009	TXDB011	—	—
110010b*1	—	SCL00	SDA00	—	—	—	—

—: Do not specify this value.

Note 1. Do not specify this value on the products with 48 Kbytes of RAM.

21.2.14 PDn Pin Function Control Register (PDnPFS) (n = 0 to 7)

Address(es): PD0PFS 0008 C1A8h, PD1PFS 0008 C1A9h, PD2PFS 0008 C1AAh, PD3PFS 0008 C1ABh, PD4PFS 0008 C1ACh, PD5PFS 0008 C1ADh, PD6PFS 0008 C1AEh, PD7PFS 0008 C1AFh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	ISEL	PSEL[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PSEL[5:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 21.17 to Table 21.20.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin PD4: IRQ2 (64/80/100 pin) PD5: IRQ6 (48/64/80/100 pin) PD6: IRQ5 (64/80/100 pin) PD7: IRQ8 (48/64/80/100 pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 21.17 Register Settings for Input/Output Pin Function in 100-Pin Products

	Pin							
PSEL[5:0] Settings	PD0	PD1	PD2	PD3	PD4	PD5	PD6	PD7
000000b (initial value)	Hi-Z							
000001b	—	—	—	—	—	—	MTIOC9C	MTIOC9A
000011b	—	—	—	—	—	—	MTIOC9C#	MTIOC9A#
000101b	TMO6	TMO2	TMCI1	TMO0	TMCI0	TMRI0	TMO1	TMRI1
000110b	—	—	TMO4	—	TMCI6	TMRI6	—	TMRI5
001001b	—	—	—	—	—	—	ADST0	—
001010b	—	—	SCK5	TXD1 SMOSI1 SSDA1	SCK1	RXD1 SMISO1 SSCL1	CTS1# RTS1# SS1#	TXD5 SMOSI5 SSDA5
001100b	—	—	—	—	TXD12 SMOSI12 SSDA12 TXDX12 SIOX12	—	RXD12 SMISO12 SSCL12 RXDX12	—
001101b	RSPCKA	MISOA	MOSIA	—	—	—	SSLA0	SSLA1
001110b	RSPCK0	MISO0	MOSI0	MOSI0	SSL02	SSL00	SSL00	SSL01
010000b	—	—	—	—	—	—	—	CTX0
010100b	GTIOC3B	GTIOC3A	GTIOC2B	GTIOC2A	GTIOC1B	GTIOC1A	GTIOC0B	GTIOC0A
010101b	GTIOC1A	GTIOC0B	GTIOC0A	GTETRGC	GTETRGB	GTETRGA	GTIOC3B	GTIOC3A
010110b	GTIOC3B#	GTIOC3A#	GTIOC2B#	GTIOC2A#	GTIOC1B#	GTIOC1A#	GTIOC0B#	GTIOC0A#
010111b	GTIOC1A#	GTIOC0B#	GTIOC0A#	GTIOC7B	—	GTIOC7A	GTIOC3B#	GTIOC3A#
011000b	—	—	—	—	—	—	GTIW	GTIU
101100b	—	—	—	—	—	—	—	SCK009
101101b	TXD008 TXDA008 SMOSI008 SSDA008	RXD008 SMISO008 SSCL008	SCK008	TXD011 TXDA011 SMOSI011 SSDA011	SCK011	RXD011 SMISO011 SSCL011	CTS011# RTS011# SS011#	TXD008 TXDA008 SMOSI008 SSDA008
101110b	—	—	TXDB008	—	TXDB011	—	DE011	TXDB009

—: Do not specify this value.

Table 21.18 Register Settings for Input/Output Pin Function in 80-Pin Products

	Pin					
PSEL[5:0] Settings	**PD2**	**PD3**	**PD4**	**PD5**	**PD6**	**PD7**
000000b (initial value)	Hi-Z					
000001b	—	—	—	—	MTIOC9C	MTIOC9A
000011b	—	—	—	—	MTIOC9C#	MTIOC9A#
000101b	TMCI1	TMO0	TMCI0	TMRI0	TMO1	TMRI1
000110b	TMO4	—	TMCI6	TMRI6	—	TMRI5
001001b	—	—	—	—	ADST0	—
001010b	SCK5	TXD1 SMOSI1 SSDA1	SCK1	RXD1 SMISO1 SSCL1	CTS1# RTS1# SS1#	TXD5 SMOSI5 SSDA5
001100b	—	—	TXD12 SMOSI12 SSDA12 TXDX12 SIOX12	—	RXD12 SMISO12 SSCL12 RXDX12	—
001101b	MOSIA	—	—	—	SSLA0	SSLA1
001110b	MOSI0	MOSI0	SSL02	SSL00	SSL00	SSL01
010000b	—	—	—	—	—	CTX0
010100b	GTIOC2B	GTIOC2A	GTIOC1B	GTIOC1A	GTIOC0B	GTIOC0A
010101b	GTIOC0A	GTETRGC	GTETRGB	GTETRGA	GTIOC3B	GTIOC3A
010110b	GTIOC2B#	GTIOC2A#	GTIOC1B#	GTIOC1A#	GTIOC0B#	GTIOC0A#
010111b	GTIOC0A#	GTIOC7B	—	GTIOC7A	GTIOC3B#	GTIOC3A#
011000b	—	—	—	—	GTIW	GTIU
101100b	—	—	—	—	—	SCK009
101101b	SCK008	TXD011 TXDA011 SMOSI011 SSDA011	SCK011	RXD011 SMISO011 SSCL011	CTS011# RTS011# SS011#	TXD008 TXDA008 SMOSI008 SSDA008
101110b	TXDB008	—	TXDB011	—	DE011	TXDB009

—: Do not specify this value.

Table 21.19 Register Settings for Input/Output Pin Function in 64-Pin Products

	Pin				
PSEL[5:0] Settings	**PD3**	**PD4**	**PD5**	**PD6**	**PD7**
000000b (initial value)	Hi-Z				
000001b	—	—	—	MTIOC9C	MTIOC9A
000011b	—	—	—	MTIOC9C#	MTIOC9A#
000101b	TMO0	TMCI0	TMRI0	TMO1	TMRI1
000110b	—	TMCI6	TMRI6	—	TMRI5
001001b	—	—	—	ADST0	—
001010b	TXD1 SMOSI1 SSDA1	SCK1	RXD1 SMISO1 SSCL1	CTS1# RTS1# SS1#	TXD5 SMOSI5 SSDA5
001100b	—	TXD12 SMOSI12 SSDA12 TXDX12 SIOX12	—	RXD12 SMISO12 SSCL12 RXDX12	—
001101b	—	—	—	SSLA0	SSLA1
001110b*1	MOSI0	SSL02	SSL00	SSL00	SSL01
010000b	—	—	—	—	CTX0
010100b	GTIOC2A	GTIOC1B	GTIOC1A	GTIOC0B	GTIOC0A
010101b	GTETRGC	GTETRGB	GTETRGA	GTIOC3B	GTIOC3A
010110b	GTIOC2A#	GTIOC1B#	GTIOC1A#	GTIOC0B#	GTIOC0A#
010111b	GTIOC7B	—	GTIOC7A	GTIOC3B#	GTIOC3A#
011000b	—	—	—	GTIW	GTIU
101100b*1	—	—	—	—	SCK009
101101b*1	TXD011 TXDA011 SMOSI011 SSDA011	SCK011	RXD011 SMISO011 SSCL011	CTS011# RTS011# SS011#	TXD008 TXDA008 SMOSI008 SSDA008
101110b*1	—	TXDB011	—	DE011	TXDB009

—: Do not specify this value.

Note 1. Do not specify this value on the products with 48 Kbytes of RAM.

Table 21.20 Register Settings for Input/Output Pin Function in 48-Pin Products

	Pin		
PSEL[5:0] Settings	**PD3**	**PD5**	**PD7**
000000b (initial value)	Hi-Z		
000001b	—	—	MTIOC9A
000011b	—	—	MTIOC9A#
000101b	TMO0	TMRI0	TMRI1
000110b	—	TMRI6	TMRI5
001010b	TXD1 SMOSI1 SSDA1	RXD1 SMISO1 SSCL1	TXD5 SMOSI5 SSDA5
001101b	—	—	SSLA1
001110b*1	MOSI0	SSL00	SSL01
010000b	—	—	CTX0
010100b	GTIOC2A	GTIOC1A	GTIOC0A
010101b	GTETRGC	GTETRGA	GTIOC3A
010110b	GTIOC2A#	GTIOC1A#	GTIOC0A#
010111b	GTIOC7B	GTIOC7A	GTIOC3A#
011000b	—	—	GTIU
101100b*1	—	—	SCK009
101101b*1	TXD011 TXDA011 SMOSI011 SSDA011	RXD011 SMISO011 SSCL011	TXD008 TXDA008 SMOSI008 SSDA008
101110b*1	—	—	TXDB009

—: Do not specify this value.

Note 1. Do not specify this value on the products with 48 Kbytes of RAM.

21.2.15 PEn Pin Function Control Register (PEnPFS) (n = 0 to 5)

Address(es): PE0PFS 0008 C1B0h, PE1PFS 0008 C1B1h, PE2PFS 0008 C1B2h, PE3PFS 0008 C1B3h, PE4PFS 0008 C1B4h, PE5PFS 0008 C1B5h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	ISEL	PSEL[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PSEL[5:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 21.21 to Table 21.23.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin PE0: IRQ7 (100 pin) PE1: IRQ15 (100 pin) PE2: IRQ0 (48/64/80/100 pin) PE3: IRQ2 (80/100 pin) PE4: IRQ1 (80/100 pin) PE5: IRQ0 (100 pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 21.21 Register Settings for Input/Output Pin Function in 100-Pin Products

	Pin					
PSEL[5:0] Settings	**PE0**	**PE1**	**PE2**	**PE3**	**PE4**	**PE5**
000000b (initial value)	Hi-Z					
000001b	MTIOC9B	MTIOC9D	—	—	—	MTIOC9D
000010b	—	—	—	MTCLKD	MTCLKC	—
000011b	MTIOC9B#	MTIOC9D#	—	—	—	MTIOC9D#
000100b	—	—	—	MTCLKD#	MTCLKC#	—
000101b	TMCI1	TMO5	—	—	—	—
000110b	TMCI5	—	—	—	—	—
000111b	—	—	POE10#	POE11#	POE10#	—
001001b	—	—	—	—	—	ADST0
001010b	RXD5 SMISO5 SSCL5	CTS5# RTS5# SS5#	—	—	—	—
001100b	—	CTS12# RTS12# SS12#	—	—	—	—
001101b	SSLA2	SSLA3	—	—	—	—
001110b	SSL02	SSL03	—	—	—	—
010000b	CRX0	—	—	—	—	—
010100b	—	—	—	GTETRGA	GTETRGA	GTIOC3A
010101b	—	—	—	GTETRGB	GTETRGB	GTETRGB
010110b	—	—	—	GTETRGC	GTETRGC	GTIOC3A#
010111b	—	—	—	GTETRGD	GTETRGD	GTETRGD
011000b	GTIV	—	—	—	—	—
101100b	—	—	—	CTS009# RTS009# SS009#	SCK009	SCK009
101101b	—	—	—	—	—	CTS009# RTS009# SS009#
101110b	—	—	—	DE009	TXDB009	TXDB009

—: Do not specify this value.

Table 21.22 Register Settings for Input/Output Pin Function in 80-Pin Products

	Pin		
PSEL[5:0] Settings	**PE2**	**PE3**	**PE4**
000000b (initial value)	Hi-Z		
000010b	—	MTCLKD	MTCLKC
000100b	—	MTCLKD#	MTCLKC#
000111b	POE10#	POE11#	POE10#
010100b	—	GTETRGA	GTETRGA
010101b	—	GTETRGB	GTETRGB
010110b	—	GTETRGC	GTETRGC
010111b	—	GTETRGD	GTETRGD
101100b	—	CTS009# RTS009# SS009#	SCK009
101110b	—	DE009	TXDB009

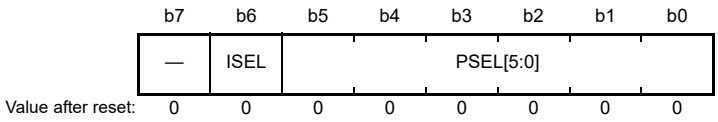
—: Do not specify this value.

Table 21.23 Register Settings for Input/Output Pin Function in 64-/48-Pin Products

	Pin
PSEL[5:0] Settings	**PE2**
000000b (initial value)	Hi-Z
000111b	POE10#

21.2.16 PN7 Pin Function Control Register (PN7PFS)

Address(es): PN7PFS 0008 C1F7h



Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PSEL[5:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 21.24.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin PN7: IRQ5 (64/80/100 pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 21.24 Register Settings for Input/Output Pin Function in 100-/80-/64-Pin Products

	Pin
PSEL[5:0] Settings	**PN7**
000000b (initial value)	Hi-Z
000001b	MTIOC9D
000011b	MTIOC9D#
001001b	ADST0

21.3 Usage Notes

21.3.1 Procedure for Specifying Input/Output Pin Function

Use the following procedure to specify the input/output pin functions.

(1) Clear the port mode register (PMR) for the target pin to 0 to select the general I/O port.
(2) Specify the assignments of input/output signals for peripheral modules to the desired pins.
(3) Enable writing to the Pmn pin function control register (PmnPFS) through the write-protect register (PWPR) setting. (m = 0 to 9, A, B, D, E, and N, n = 0 to 7)
(4) Specify the input/output function for the pin through the PSEL[5:0] bit settings in the PmnPFS register.
(5) Clear the PFSWE bit in the PWPR register to 0 to disable writing to the PmnPFS register.
(6) Set the PMR to 1 as necessary to switch to the selected input/output function for the pin.

21.3.2 Notes on MPC Register Setting

(1) Settings of the Pmn pin function control register (PmnPFS) should be made only while the PMR register for the target pin is set to 0. If the PmnPFS register is set while the PMR register is 1, unexpected edges may be input through the input pin or unexpected pulses are output through the output pin.
(2) Only the allowed values (functions) should be specified in the Pmn pin function control registers. If a value that is not allowed for the register is specified, correct operation is not guaranteed.
(3) Do not assign a single function to multiple pins through the MPC settings.
(4) Ports 2, 4 to 6 also function as analog Input/output pins for the A/D converter, comparator, and D/A converter. When using these ports as analog Input/output pins, set them to general input pins by clearing the corresponding bits in the port mode register (PMR) and port direction register (PDR) to 0 and set the PmnPFS.ASEL bit to 1, to avoid degradation of accuracy.
(5) Points to note regarding the port mode register (PMR), port direction register (PDR), and the PmnPFS register settings for pins that have multiplexed pin functions are listed in Table 21.25. The pin states are readable if the value of the ASEL bit is 0. Ensure that the PMR.Bn bit is 0 when changes to the PSEL[5:0] bits are made.

Table 21.25 Register Settings

Item	PMR.Bn	PDR.Bn	PmnPFS			Point to Note
			ASEL	ISEL	PSEL[5:0]	
After a reset	0	0	0	0	000000b	Pins function as general input port pins after release from the reset state.
General input ports	0	0	0	0/1	x	Set the PmnPFS.ISEL bit to 1 if these are multiplexed with interrupt inputs.
General output ports	0	1	0	0	x	
Peripheral functions	1	x	0	0/1	Peripheral functions (see Table 21.2 to Table 21.24)	Set the PmnPFS.ISEL bit to 1 if these are multiplexed with interrupt inputs.
Interrupt inputs	0	0	0	1	x	
NMI	x	x	x	x*1	x	Register settings are not required.
Analog inputs and outputs	0	0	1	x*1	x	Set these as general input port pins so that the output buffers are turned off.
JTAG interface	0	x	x	0	x	Set the PMR.Bn bit and the PmnPFS.ISEL bit to 0 and switch the input buffers off.
FINE interface	0	x	x	0	x	Set the PMR.Bn bit and the PmnPFS.ISEL bit to 0 and switch the input buffers off.
EXTAL/XTAL	0	0	x	x*1	x	Set these as general input port pins so that the output buffers are turned off.

x: Setting not required.
0/1: Setting the PmnPFS.ISEL bit to 0 makes the pin incapable of functioning as an IRQ pin.
Setting the PmnPFS.ISEL bit to 1 makes the pin capable of functioning as an IRQ pin (When a IRQ is assigned).
Note 1. Even if the PmnPFS.ISEL bit is set to 1, the pin will not function as an IRQn input pin.

Note:
- The pin state is readable when the PmnPFS.ASEL bit is 0.
- If the value of the PmnPFS.PSEL[5:0] bits is to be changed, do so while the PMR.Bn bit is 0.
- If an RIIC or RI3C function is assigned to a port pin, clear the PMR.Bn (to 0); pulling up is automatically turned off for outputs from peripheral modules other than the RIIC or RI3C.

21.3.3 Notes on the Use of Analog Functions

To use an analog function, set the corresponding bits in both the port mode register (PMR) and port direction register (PDR) to 0 so that the pin acts as a general input port. After that, set the pin function select bit in the Pmn pin function control register (PmnPFS.ASEL) to 1.

21.3.4 Notes on Controlling the Switch to General I/O Port Pin Operation by POE3

On the generation of requests for disabling of output specified by the POE3, the pins for which the setting is 1 in the corresponding PMMCRn register (n = 0 to 3) of the POE3 are switched to general I/O port pins. Set the bits of the corresponding POECRn register (n = 0 to 3) to 0 beforehand.

21.3.5 Notes on Inverting Input/Output Function for the MTU and GPTW Input/Output Pins

The MTU and GPTW input/output pins shown in Table 21.26 can invert and output the signal captured by inverting the input signal when the PmnPFS.PSEL[5:0] bit for the target pins is set. Setting for switching the state between the normal input/output and the inverting input/output must be done while the PMR register for the target pins is set as 0.

Table 21.26 MTU and GPTW Input/Output Pins (1/2)

Module/Function	Channel	Normal Input/Output	Inverting Input/Output
Multi-function timer unit 3	MTU0	MTIOC0A	MTIOC0A#
		MTIOC0B	MTIOC0B#
		MTIOC0C	MTIOC0C#
		MTIOC0D	MTIOC0D#
	MTU1	MTIOC1A	MTIOC1A#
		MTIOC1B	MTIOC1B#
	MTU2	MTIOC2A	MTIOC2A#
		MTIOC2B	MTIOC2B#
	MTU3	MTIOC3A	MTIOC3A#
		MTIOC3B	MTIOC3B#
		MTIOC3C	MTIOC3C#
		MTIOC3D	MTIOC3D#
	MTU4	MTIOC4A	MTIOC4A#
		MTIOC4B	MTIOC4B#
		MTIOC4C	MTIOC4C#
		MTIOC4D	MTIOC4D#
	MTU5	MTIC5U	MTIC5U#
		MTIC5V	MTIC5V#
		MTIC5W	MTIC5W#
	MTU6	MTIOC6A	MTIOC6A#
		MTIOC6B	MTIOC6B#
		MTIOC6C	MTIOC6C#
		MTIOC6D	MTIOC6D#
	MTU7	MTIOC7A	MTIOC7A#
		MTIOC7B	MTIOC7B#
		MTIOC7C	MTIOC7C#
		MTIOC7D	MTIOC7D#
	MTU9	MTIOC9A	MTIOC9A#
		MTIOC9B	MTIOC9B#
		MTIOC9C	MTIOC9C#
		MTIOC9D	MTIOC9D#
	MTU	MTCLKA	MTCLKA#
		MTCLKB	MTCLKB#
		MTCLKC	MTCLKC#
		MTCLKD	MTCLKD#

Table 21.26 MTU and GPTW Input/Output Pins (2/2)

Module/Function	Channel	Normal Input/Output	Inverting Input/Output
General PWM timer	GPTW0	GTIOC0A	GTIOC0A#
		GTIOC0B	GTIOC0B#
	GPTW1	GTIOC1A	GTIOC1A#
		GTIOC1B	GTIOC1B#
	GPTW2	GTIOC2A	GTIOC2A#
		GTIOC2B	GTIOC2B#
	GPTW3	GTIOC3A	GTIOC3A#
		GTIOC3B	GTIOC3B#
	GPTW4	GTIOC4A	GTIOC4A#
		GTIOC4B	GTIOC4B#
	GPTW5	GTIOC5A	GTIOC5A#
		GTIOC5B	GTIOC5B#
	GPTW6	GTIOC6A	GTIOC6A#
		GTIOC6B	GTIOC6B#
	GPTW7	GTIOC7A	GTIOC7A#
		GTIOC7B	GTIOC7B#

22. Multi-Function Timer Pulse Unit 3 (MTU3d)

22.1 Overview

This MCU has an on-chip multi-function timer pulse unit 3 (MTU3d), consisting of nine 16-bit timer channels. Table 22.1 shows the specifications of the MTU and Table 22.2 lists the functions of the MTU. Figure 22.1 and Figure 22.2 show block diagrams of the MTU.

Table 22.1 MTU Specifications

Item	Description
Pulse input/output	28 lines max.
Pulse input	3 lines
Count clock	11 clocks for each channel (14 clocks for MTU0 and MTU9, 12 clocks for MTU2, 10 clocks for MTU5, and four clocks for MTU1 & MTU2 (LWA = 1))
Available operations	[MTU0 to MTU4, MTU6, MTU7, MTU9] • Waveform output on compare match • Input capture function (noise filter setting available) • Counter-clearing operation • Simultaneous writing to multiple timer counters (TCNT) • Simultaneous clearing on compare match or input capture • Simultaneous input and output to registers in synchronization with counter operations • Up to 14-phase PWM output in combination with synchronous operation
	[MTU0, MTU3, MTU4, MTU6, MTU7, MTU9] • Buffer operation specifiable
	[MTU1, MTU2] • Phase counting mode can be specified independently • 32-bit phase counting mode can be specified for interlocked operation of MTU1 and MTU2 (when TMDR3.LWA = 1) • Cascade connection operation available
	[MTU3, MTU4, MTU6, MTU7] • Through interlocked operation of MTU3/4 and MTU6/7, the positive and negative signals in six phases (12 phases in total) can be output in complementary PWM and reset-synchronized PWM operation. • In complementary PWM mode, transfer of values from buffer registers to temporary registers on crests or troughs of the timer-counter values or writing to the buffer registers (MTU4.TGRD and MTU7.TGRD) • Double-buffering selectable in complementary PWM mode
	[MTU3, MTU4] • Through interlocking with MTU0, a mode for driving AC synchronous motors (brushless DC motors) by using complementary PWM output and reset-synchronized PWM output is settable and allows the selection of two types of waveform output (chopping or level)
	[MTU5] • Capable of operation as a dead-time compensation counter
	[MTU6, MTU7] • Through interlocking with MTU9, a mode for driving AC synchronous motors (brushless DC motors) by using complementary PWM output and reset-synchronized PWM output is settable and allows the selection of two types of waveform output (chopping or level)
Interrupt skipping function	• In complementary PWM mode, interrupts on crests and troughs of counter values and triggers to start conversion by the A/D converter can be skipped
Interrupt sources	45 sources
Buffer operation	Automatic transfer of register data (transfer from the buffer register to the timer register)
Trigger generation	A/D conversion start triggers can be generated
	A/D conversion start request delaying function enables A/D converter to be started with any desired timing and to be synchronized with PWM output
Low power consumption function	Module stop mode can be set

Table 22.2 MTU Functions (1/2)

Item		MTU0	MTU1	MTU2	MTU1 & MTU2 (LWA = 1)	MTU3	MTU4	MTU5	MTU6	MTU7	MTU9
Count clock		PCLKC/1 PCLKC/2 PCLKC/4 PCLKC/8 PCLKC/16 PCLKC/32 PCLKC/64 PCLKC/256 PCLKC/1024 MTCLKA MTCLKB MTCLKC MTCLKD MTIOC1A	PCLKC/1 PCLKC/2 PCLKC/4 PCLKC/8 PCLKC/16 PCLKC/32 PCLKC/64 PCLKC/256 PCLKC/1024 MTCLKA MTCLKB	PCLKC/1 PCLKC/2 PCLKC/4 PCLKC/8 PCLKC/16 PCLKC/32 PCLKC/64 PCLKC/256 PCLKC/1024 MTCLKA MTCLKB MTCLKC	MTCLKA MTCLKB MTCLKC MTCLKD	PCLKC/1 PCLKC/2 PCLKC/4 PCLKC/8 PCLKC/16 PCLKC/32 PCLKC/64 PCLKC/256 PCLKC/1024 MTCLKA MTCLKB	PCLKC/1 PCLKC/2 PCLKC/4 PCLKC/8 PCLKC/16 PCLKC/32 PCLKC/64 PCLKC/256 PCLKC/1024 MTCLKA MTCLKB	PCLKC/1 PCLKC/2 PCLKC/4 PCLKC/8 PCLKC/16 PCLKC/32 PCLKC/64 PCLKC/256 PCLKC/1024 MTIOC1A	PCLKC/1 PCLKC/2 PCLKC/4 PCLKC/8 PCLKC/16 PCLKC/32 PCLKC/64 PCLKC/256 PCLKC/1024 MTCLKA MTCLKB	PCLKC/1 PCLKC/2 PCLKC/4 PCLKC/8 PCLKC/16 PCLKC/32 PCLKC/64 PCLKC/256 PCLKC/1024 MTCLKA MTCLKB	PCLKC/1 PCLKC/2 PCLKC/4 PCLKC/8 PCLKC/16 PCLKC/32 PCLKC/64 PCLKC/256 PCLKC/1024 MTCLKA MTCLKB MTCLKC MTCLKD MTIOC1A
External clock for phase counting mode		—	MTCLKA MTCLKB	MTCLKA MTCLKB MTCLKC MTCLKD	MTCLKA MTCLKB MTCLKC MTCLKD	—	—	—	—	—	—
General registers (TGR)		TGRA TGRB TGRE	TGRA TGRB	TGRA TGRB	TGRALW TGRBLW	TGRA TGRB	TGRA TGRB	TGRU TGRV TGRW	TGRA TGRB	TGRA TGRB	TGRA TGRB TGRE
General registers/ buffer registers		TGRC TGRD TGRF	—	—	—	TGRC TGRD TGRE	TGRC TGRD TGRE TGRF	—	TGRC TGRD TGRE	TGRC TGRD TGRE TGRF	TGRC TGRD TGRF
I/O pins		MTIOC0A MTIOC0B MTIOC0C MTIOC0D	MTIOC1A MTIOC1B	MTIOC2A MTIOC2B	MTIOC1A MTIOC1B	MTIOC3A MTIOC3B MTIOC3C MTIOC3D	MTIOC4A MTIOC4B MTIOC4C MTIOC4D	MTIC5U MTIC5V MTIC5W	MTIOC6A MTIOC6B MTIOC6C MTIOC6D	MTIOC7A MTIOC7B MTIOC7C MTIOC7D	MTIOC9A MTIOC9B MTIOC9C MTIOC9D
Counter clear function		TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGRALW/ TGRBLW compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture
Compare match output	0 output	✓	✓	✓	✓	✓	✓	—	✓	✓	✓
	1 output	✓	✓	✓	✓	✓	✓	—	✓	✓	✓
	Toggle output	✓	✓	✓	✓	✓	✓	—	✓	✓	✓
Input capture function		✓	✓	✓	✓*1	✓	✓	✓	✓	✓	✓
Synchronous operation		✓	✓	✓	—	✓	✓	—	✓	✓	✓
PWM mode 1		✓	✓	✓	—	✓	✓	—	✓	✓	✓
PWM mode 2		✓	✓	✓	—	—	—	—	—	—	✓
Complementary PWM mode		—	—	—	—	✓	✓	—	✓	✓	—
Reset-synchronized PWM mode		—	—	—	—	✓	✓	—	✓	✓	—
AC synchronous motor drive mode		✓	—	—	—	✓	✓	—	✓	✓	✓
Phase counting mode		—	✓	✓	✓	—	—	—	—	—	—
Buffer operation		✓	—	—	—	✓	✓	—	✓	✓	✓
Dead time compensation counter function		—	—	—	—	—	—	✓	—	—	—
DMAC/DTC trigger sources		TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGRALW/ TGRBLW compare match or input capture	TGR compare match or input capture	TGR compare match or input capture and TCNT overflow/ underflow*2	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture and TCNT overflow/ underflow*2	TGR compare match or input capture
A/D conversion start trigger		TGRA compare match or input capture TGRE compare match	TGRA compare match or input capture	TGRA compare match or input capture	TGRALW compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture, or TCNT underflow (trough) in complementary PWM mode	—	TGRA compare match or input capture	TGRA compare match or input capture, or TCNT underflow (trough) in complementary PWM mode	TGRA compare match or input capture TGRE compare match

Table 22.2 MTU Functions (2/2)

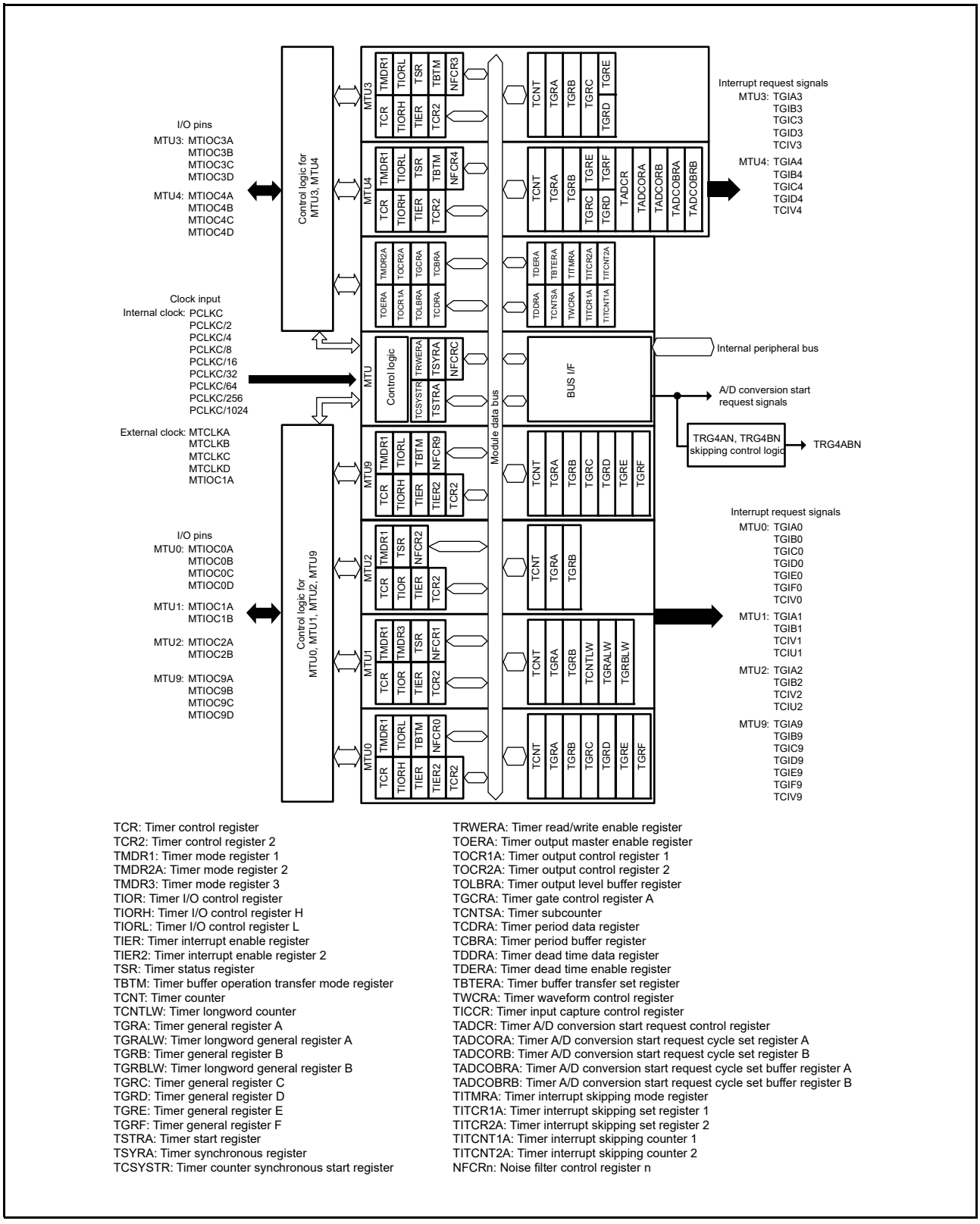
Item	MTU0	MTU1	MTU2	MTU1 & MTU2 (LWA = 1)	MTU3	MTU4	MTU5	MTU6	MTU7	MTU9
Interrupt sources	Seven sources • Compare match or input capture 0A • Compare match or input capture 0B • Compare match or input capture 0C • Compare match or input capture 0D • Compare match 0E • Compare match 0F • Overflow	Four sources • Compare match or input capture 1A • Compare match or input capture 1B • Overflow • Underflow	Four sources • Compare match or input capture 2A • Compare match or input capture 2B • Overflow • Underflow	Four sources • Compare match or input capture 1A • Compare match or input capture 1B • Overflow • Underflow	Five sources • Compare match or input capture 3A • Compare match or input capture 3B • Compare match or input capture 3C • Compare match or input capture 3D • Overflow	Five sources • Compare match or input capture 4A • Compare match or input capture 4B • Compare match or input capture 4C • Compare match or input capture 4D • Overflow or underflow*2	Three sources • Compare match or input capture 5U • Compare match or input capture 5V • Compare match or input capture 5W	Five sources • Compare match or input capture 6A • Compare match or input capture 6B • Compare match or input capture 6C • Compare match or input capture 6D • Overflow	Five sources • Compare match or input capture 7A • Compare match or input capture 7B • Compare match or input capture 7C • Compare match or input capture 7D • Overflow or underflow*2	Seven sources • Compare match or input capture 9A • Compare match or input capture 9B • Compare match or input capture 9C • Compare match or input capture 9D • Compare match 9E • Compare match 9F • Overflow
Event link function (output)	Seven sources • Compare match 0A • Compare match 0B • Compare match 0C • Compare match 0D • Compare match 0E • Compare match 0F • Overflow	—	—	—	Five sources • Compare match 3A • Compare match 3B • Compare match 3C • Compare match 3D • Overflow	Six sources • Compare match 4A • Compare match 4B • Compare match 4C • Compare match 4D • Overflow • Underflow*2	—	Five sources • Compare match 6A • Compare match 6B • Compare match 6C • Compare match 6D • Overflow	Six sources • Compare match 7A • Compare match 7B • Compare match 7C • Compare match 7D • Overflow • Underflow*2	Seven sources • Compare match 9A • Compare match 9B • Compare match 9C • Compare match 9D • Compare match 9E • Compare match 9F • Overflow
Event link function (input)	• Start counting • Input capture (to be captured in the TGRA) • Restart counting (clear counter)	—	—	—	• Start counting • Input capture (to be captured in the TGRA) • Restart counting (clear counter)	• Start counting • Input capture (to be captured in the TGRA) • Restart counting (clear counter)	—	• Start counting • Input capture (to be captured in the TGRA) • Restart counting (clear counter)	• Start counting • Input capture (to be captured in the TGRA) • Restart counting (clear counter)	• Start counting • Input capture (to be captured in the TGRA) • Restart counting (clear counter)
A/D conversion start request delaying function	—	—	—	—	—	A/D conversion start request at a match between TADCORA and TCNT or A/D conversion start request at a match between TADCORB and TCNT	—	—	A/D conversion start request at a match between TADCORA and TCNT or A/D conversion start request at a match between TADCORB and TCNT	—
Interrupt skipping 1	—	—	—	—	Skips TGRA compare match interrupts	Skips TCIV interrupts	—	Skips TGRA compare match interrupts	Skips TCIV interrupts	—
Interrupt skipping 2	—	—	—	—	—	Skipping in compare count between TADCORA and TCNT, and TADCORB and TCNT	—	—	Skipping in compare count between TADCORA and TCNT, and TADCORB and TCNT	—
Module stop function	MSTPCRA.MSTPA9*3									

✓: Possible —: Not possible

Note 1. When LWA is 1, the TGRALW capture source can be selected from either of the following: an input from MTIOC1A or MTU0.TGRA compare match/input capture event.
The TGRBLW capture source can be selected from any of the following: an input from MTIOC1B, MTU0.TGRC compare match/input capture event.

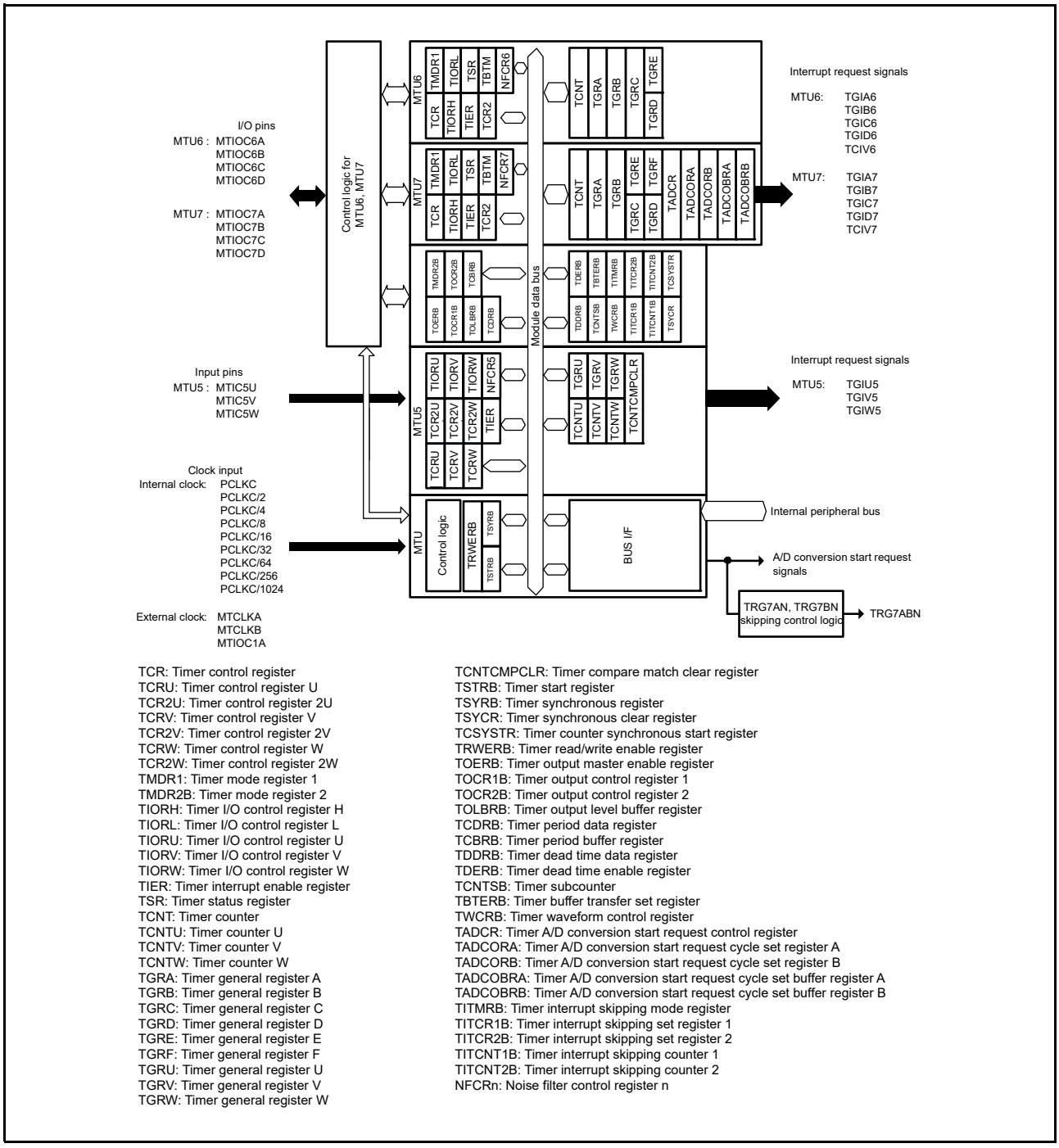
Note 2. Underflow is available only in complementary PWM mode.

Note 3. For details on the module stop function, refer to section 11, Low Power Consumption.



TCR: Timer control register
TCR2: Timer control register 2
TMDR1: Timer mode register 1
TMDR2A: Timer mode register 2
TMDR3: Timer mode register 3
TIOR: Timer I/O control register
TIORH: Timer I/O control register H
TIORL: Timer I/O control register L
TIER: Timer interrupt enable register
TIER2: Timer interrupt enable register 2
TSR: Timer status register
TBTM: Timer buffer operation transfer mode register
TCNT: Timer counter
TCNTLW: Timer longword counter
TGRA: Timer general register A
TGRALW: Timer longword general register A
TGRB: Timer general register B
TGRBLW: Timer longword general register B
TGRC: Timer general register C
TGRD: Timer general register D
TGRE: Timer general register E
TGRF: Timer general register F
TSTRA: Timer start register
TSYRA: Timer synchronous register
TCSYSTR: Timer counter synchronous start register
TRWERA: Timer read/write enable register
TOERA: Timer output master enable register
TOCR1A: Timer output control register 1
TOCR2A: Timer output control register 2
TOLBRA: Timer output level buffer register
TGCRA: Timer gate control register A
TCNTSA: Timer subcounter
TCDRA: Timer period data register
TCBRA: Timer period buffer register
TDDRA: Timer dead time data register
TDERA: Timer dead time enable register
TBTERA: Timer buffer transfer set register
TWCRA: Timer waveform control register
TICCR: Timer input capture control register
TADCR: Timer A/D conversion start request control register
TADCORA: Timer A/D conversion start request cycle set register A
TADCORB: Timer A/D conversion start request cycle set register B
TADCOBRA: Timer A/D conversion start request cycle set buffer register A
TADCOBRB: Timer A/D conversion start request cycle set buffer register B
TITMRA: Timer interrupt skipping mode register
TITCR1A: Timer interrupt skipping set register 1
TITCR2A: Timer interrupt skipping set register 2
TITCNT1A: Timer interrupt skipping counter 1
TITCNT2A: Timer interrupt skipping counter 2
NFCRn: Noise filter control register n

Figure 22.1 Block Diagram of MTU (MTU0 to MTU4, MTU9)



TCR: Timer control register
TCRU: Timer control register U
TCR2U: Timer control register 2U
TCRV: Timer control register V
TCR2V: Timer control register 2V
TCRW: Timer control register W
TCR2W: Timer control register 2W
TMDR1: Timer mode register 1
TMDR2B: Timer mode register 2
TIORH: Timer I/O control register H
TIORL: Timer I/O control register L
TIORU: Timer I/O control register U
TIORV: Timer I/O control register V
TIORW: Timer I/O control register W
TIER: Timer interrupt enable register
TSR: Timer status register
TCNT: Timer counter
TCNTU: Timer counter U
TCNTV: Timer counter V
TCNTW: Timer counter W
TGRA: Timer general register A
TGRB: Timer general register B
TGRC: Timer general register C
TGRD: Timer general register D
TGRE: Timer general register E
TGRF: Timer general register F
TGRU: Timer general register U
TGRV: Timer general register V
TGRW: Timer general register W

TCNTCMPCLR: Timer compare match clear register
TSTRB: Timer start register
TSYRB: Timer synchronous register
TSYCR: Timer synchronous clear register
TCSYSTR: Timer counter synchronous start register
TRWERB: Timer read/write enable register
TOERB: Timer output master enable register
TOCR1B: Timer output control register 1
TOCR2B: Timer output control register 2
TOLBRB: Timer output level buffer register
TCDRB: Timer period data register
TCBRB: Timer period buffer register
TDDRB: Timer dead time data register
TDERB: Timer dead time enable register
TCNTSB: Timer subcounter
TBTERB: Timer buffer transfer set register
TWCRB: Timer waveform control register
TADCR: Timer A/D conversion start request control register
TADCORA: Timer A/D conversion start request cycle set register A
TADCORB: Timer A/D conversion start request cycle set register B
TADCOBRA: Timer A/D conversion start request cycle set buffer register A
TADCOBRB: Timer A/D conversion start request cycle set buffer register B
TITMRB: Timer interrupt skipping mode register
TITCR1B: Timer interrupt skipping set register 1
TITCR2B: Timer interrupt skipping set register 2
TITCNT1B: Timer interrupt skipping counter 1
TITCNT2B: Timer interrupt skipping counter 2
NFCRn: Noise filter control register n

Figure 22.2 Block Diagram of MTU (MTU5 to MTU7)

Table 22.3 shows the configuration of pins for the MTU.

Table 22.3 Pin Configuration of the MTU

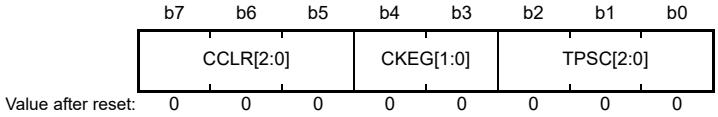
Channel	Pin Name	I/O	Function
MTU	MTCLKA	Input	External clock A input pin (MTU1 and MTU2 phase counting mode A phase input)
	MTCLKB	Input	External clock B input pin (MTU1 and MTU2 phase counting mode B phase input)
	MTCLKC	Input	External clock C input pin (MTU2 phase counting mode A phase input)
	MTCLKD	Input	External clock D input pin (MTU2 phase counting mode B phase input)
	ADSM0	Output	A/D conversion start request frame synchronization signal 0 output pin
	ADSM1	Output	A/D conversion start request frame synchronization signal 1 output pin
MTU0	MTIOC0A	I/O	MTU0 TGRA input capture input/output compare output/PWM output pin
	MTIOC0B	I/O	MTU0 TGRB input capture input/output compare output/PWM output pin
	MTIOC0C	I/O	MTU0 TGRC input capture input/output compare output/PWM output pin
	MTIOC0D	I/O	MTU0 TGRD input capture input/output compare output/PWM output pin
MTU1	MTIOC1A	I/O	MTU1 TGRA input capture input/output compare output/PWM output pin
	MTIOC1B	I/O	MTU1 TGRB input capture input/output compare output/PWM output pin
MTU2	MTIOC2A	I/O	MTU2 TGRA input capture input/output compare output/PWM output pin
	MTIOC2B	I/O	MTU2 TGRB input capture input/output compare output/PWM output pin
MTU3	MTIOC3A	I/O	MTU3 TGRA input capture input/output compare output/PWM output pin
	MTIOC3B	I/O	MTU3 TGRB input capture input/output compare output/PWM output pin
	MTIOC3C	I/O	MTU3 TGRC input capture input/output compare output/PWM output pin
	MTIOC3D	I/O	MTU3 TGRD input capture input/output compare output/PWM output pin
MTU4	MTIOC4A	I/O	MTU4 TGRA input capture input/output compare output/PWM output pin
	MTIOC4B	I/O	MTU4 TGRB input capture input/output compare output/PWM output pin
	MTIOC4C	I/O	MTU4 TGRC input capture input/output compare output/PWM output pin
	MTIOC4D	I/O	MTU4 TGRD input capture input/output compare output/PWM output pin
MTU5	MTIC5U	Input	MTU5 TGRU input capture input/external pulse input pin
	MTIC5V	Input	MTU5 TGRV input capture input/external pulse input pin
	MTIC5W	Input	MTU5 TGRW input capture input/external pulse input pin
MTU6	MTIOC6A	I/O	MTU6 TGRA input capture input/output compare output/PWM output pin
	MTIOC6B	I/O	MTU6 TGRB input capture input/output compare output/PWM output pin
	MTIOC6C	I/O	MTU6 TGRC input capture input/output compare output/PWM output pin
	MTIOC6D	I/O	MTU6 TGRD input capture input/output compare output/PWM output pin
MTU7	MTIOC7A	I/O	MTU7 TGRA input capture input/output compare output/PWM output pin
	MTIOC7B	I/O	MTU7 TGRB input capture input/output compare output/PWM output pin
	MTIOC7C	I/O	MTU7 TGRC input capture input/output compare output/PWM output pin
	MTIOC7D	I/O	MTU7 TGRD input capture input/output compare output/PWM output pin
MTU9	MTIOC9A	I/O	MTU9 TGRA input capture input/output compare output/PWM output pin
	MTIOC9B	I/O	MTU9 TGRB input capture input/output compare output/PWM output pin
	MTIOC9C	I/O	MTU9 TGRC input capture input/output compare output/PWM output pin
	MTIOC9D	I/O	MTU9 TGRD input capture input/output compare output/PWM output pin

22.2 Register Descriptions

22.2.1 Timer Control Register (TCR)

- MTU0.TCR, MTU1.TCR, MTU2.TCR, MTU3.TCR, MTU4.TCR, MTU6.TCR, MTU7.TCR, MTU9.TCR

Address(es): MTU0.TCR 000C 1300h, MTU1.TCR 000C 1380h, MTU2.TCR 000C 1400h, MTU3.TCR 000C 1200h, MTU4.TCR 000C 1201h, MTU6.TCR 000C 1A00h, MTU7.TCR 000C 1A01h, MTU9.TCR 000C 1580h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	TPSC[2:0]	Time Prescaler Select	Refer to Table 22.6 to Table 22.9.	R/W
b4, b3	CKEG[1:0]	Clock Edge Select	b4 b3 0 0: Count at rising edge 0 1: Count at falling edge 1 x: Count at both edges	R/W
b7 to b5	CCLR[2:0]	Counter Clear Source Select	Refer to Table 22.4 and Table 22.5.	R/W

x: Don't care

The TCR register controls the TCNT operation for each channel in combination with the TCR2 register. The MTU has a total of 11 TCR registers, one each for MTU0 to MTU4, MTU6, MTU7, and MTU9 and three (TCRU, TCRV, and TCRW) for MTU5. TCR values should be specified only while TCNT operation is stopped.

TPSC[2:0] Bits (Time Prescaler Select)

These bits select the TCNT count clock source. The count clock source can be selected independently for each channel. Refer to Table 22.6 to Table 22.9 for details.

CKEG[1:0] Bits (Clock Edge Select)

These bits select the clock edge, including the MTIOC1A pin. When the internal clock is counted at both edges, the count clock period is halved (e.g. PCLKC/4 at both edges = PCLKC/2 at rising edge). If phase counting mode is used on MTU1 and MTU2, the setting of these bits is ignored and the phase counting mode setting has priority. Internal clock edge selection is valid when the count clock source is PCLKC/2 or slower. When PCLKC/1 or the overflow/underflow in another channel is selected for the count clock source, a value can be written to these bits but counter operation compiles with the initial value.

CCLR[2:0] Bits (Counter Clear Source Select)

These bits select the TCNT counter clearing source. Refer to Table 22.4 and Table 22.5 for details.

Table 22.4 CCLR[2:0] (MTU0, MTU3, MTU4, MTU6, MTU7, MTU9)

Channel	Bit 7 CCLR[2]	Bit 6 CCLR[1]	Bit 5 CCLR[0]	Description
MTU0 MTU3 MTU4 MTU6 MTU7 MTU9	0	0	0	TCNT clearing disabled
	0	0	1	TCNT cleared by TGRA compare match/input capture
	0	1	0	TCNT cleared by TGRB compare match/input capture
	0	1	1	TCNT cleared by counter clearing in another channel performing synchronous clearing/ synchronous operation*1
	1	0	0	TCNT clearing disabled
	1	0	1	TCNT cleared by TGRC compare match/input capture*2
	1	1	0	TCNT cleared by TGRD compare match/input capture*2
	1	1	1	TCNT cleared by counter clearing in another channel performing synchronous clearing/ synchronous operation*1

Note 1. Synchronous operation is selected by setting the TSYRA.SYNCn or TSYRB.SYNCn bit to 1
Note 2. When TGRC or TGRD is used as a buffer register, TCNT is not cleared because the buffer register setting has priority and compare match/input capture does not occur.

Table 22.5 CCLR[2:0] (MTU1 and MTU2)

Channel	Bit 7 Reserved*2	Bit 6 CCLR[1]	Bit 5 CCLR[0]	Description
MTU1 MTU2	0	0	0	TCNT clearing disabled
	0	0	1	TCNT cleared by TGRA compare match/input capture (when LWA = 0) TCNTLW cleared by TGRALW compare match/input capture (when LWA = 1)
	0	1	0	TCNT cleared by TGRB compare match/input capture (when LWA = 0) TCNTLW cleared by TGRBLW compare match/input capture (when LWA = 1)
	0	1	1	TCNT cleared by counter clearing in another channel performing synchronous clearing/synchronous operation*1

Note 1. Synchronous operation is selected by setting the TSYRA.SYNCn and TSYRB.SYNCn bits to 1.
Note 2. Bit 7 is reserved in MTU1 and MTU2. It is read as 0. The write value is ignored.

- MTU5.TCRU, MTU5.TCRV, MTU5.TCRW

Address(es): MTU5.TCRU 000C 1C84h, MTU5.TCRV 000C 1C94h, MTU5.TCRW 000C 1CA4h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	TPSC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TPSC[1:0]	Time Prescaler Select	Refer to Table 22.10.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TPSC[1:0] Bits (Time Prescaler Select)

These bits select the TCNT count clock source. Refer to Table 22.10 for details.

22.2.2 Timer Control Register 2 (TCR2)

- MTU0.TCR2, MTU3.TCR2, MTU4.TCR2, MTU6.TCR2, MTU7.TCR2, MTU9.TCR2

Address(es): MTU0.TCR2 000C 1328h, MTU3.TCR2 000C 124Ch, MTU4.TCR2 000C 124Dh, MTU6.TCR2 000C 1A4Ch, MTU7.TCR2 000C 1A4Dh, MTU9.TCR2 000C 15A8h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TPSC2[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

- MTU1.TCR2, MTU2.TCR2

Address(es): MTU1.TCR2 000C 1394h, MTU2.TCR2 000C 140Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	PCB[1:0]		TPSC2[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	TPSC2[2:0]	Time Prescaler Select	Refer to Table 22.6 to Table 22.9.	R/W
b4, b3	PCB[1:0]	Phase Counting Mode Function Expansion Control	Functional Expansion Control for Phase Counting Modes 2, 3, and 5	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The TCR2 register controls the TCNT operation for each channel in combination with the TCR register. The MTU has a total of 11 TCR2 registers, one each for MTU0 to MTU4, MTU6, MTU7, and MTU9 and three (TCR2U, TCR2V, and TCR2W) for MTU5. TCR2 values should be specified only while TCNT operation is stopped.

TPSC2[2:0] Bits (Time Prescaler Select)

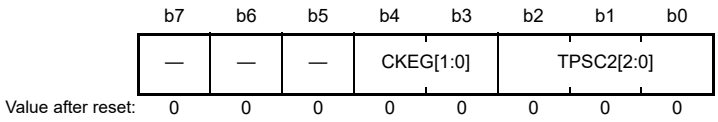
These bits select the TCNT count clock source. The count clock source can be selected independently for each channel. Refer to Table 22.6 to Table 22.9 for details.

PCB[1:0] Bits (Phase Counting Mode Function Expansion Control)

These bits control extended functions for phase counting mode 2, 3, and 5 in MTU1 and MTU2. Refer to section 22.3.6, Phase Counting Mode.

- MTU5.TCR2U, MTU5.TCR2V, MTU5.TCR2W

Address(es): MTU5.TCR2U 000C 1C85h, MTU5.TCR2V 000C 1C95h, MTU5.TCR2W 000C 1CA5h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	TPSC2[2:0]	Time Prescaler Select	Refer to Table 22.10.	R/W
b4, b3	CKEG[1:0]	Clock Edge Select	b4 b3 0 0: Counts at the rising edge. 0 1: Counts at the falling edge. 1 x: Counts at both edges.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

x: Don't care

TPSC2[2:0] Bits (Time Prescaler Select)

These bits select the TCNT count clock source. Refer to Table 22.10 for details.

CKEG[1:0] Bits (Clock Edge Select)

These bits select the edge of the count clock signal input from the MTIOC1A pin.

Table 22.6 TPSC[2:0], TPSC2[2:0] (MTU0 and MTU9)

	TCR2 register			TCR register			
	Bit 2	Bit 1	Bit 0	Bit 2	Bit 1	Bit 0	
Channel	TPSC2[2]	TPSC2[1]	TPSC2[0]	TPSC[2]	TPSC[1]	TPSC[0]	Description
MTU0 MTU9	0	0	0	0	0	0	Internal clock: counts on PCLKC/1
	0	0	0	0	0	1	Internal clock: counts on PCLKC/4
	0	0	0	0	1	0	Internal clock: counts on PCLKC/16
	0	0	0	0	1	1	Internal clock: counts on PCLKC/64
	0	0	0	1	0	0	External clock: counts on MTCLKA pin input
	0	0	0	1	0	1	External clock: counts on MTCLKB pin input
	0	0	0	1	1	0	External clock: counts on MTCLKC pin input
	0	0	0	1	1	1	External clock: counts on MTCLKD pin input
	0	0	1	x	x	x	Internal clock: counts on PCLKC/2
	0	1	0	x	x	x	Internal clock: counts on PCLKC/8
	0	1	1	x	x	x	Internal clock: counts on PCLKC/32
	1	0	0	x	x	x	Internal clock: counts on PCLKC/256
	1	0	1	x	x	x	Internal clock: counts on PCLKC/1024
	1	1	0	x	x	x	Setting prohibited
	1	1	1	x	x	x	External clock: counts on MTIOC1A pin input

x: Don't care

Table 22.7 TPSC[2:0], TPSC2[2:0] (MTU1)

	TCR2 register			TCR register			
	Bit 2	Bit 1	Bit 0	Bit 2	Bit 1	Bit 0	
Channel	TPSC2[2]	TPSC2[1]	TPSC2[0]	TPSC[2]	TPSC[1]	TPSC[0]	Description
MTU1	0	0	0	0	0	0	Internal clock: counts on PCLKC/1
	0	0	0	0	0	1	Internal clock: counts on PCLKC/4
	0	0	0	0	1	0	Internal clock: counts on PCLKC/16
	0	0	0	0	1	1	Internal clock: counts on PCLKC/64
	0	0	0	1	0	0	External clock: counts on MTCLKA pin input
	0	0	0	1	0	1	External clock: counts on MTCLKB pin input
	0	0	0	1	1	0	Internal clock: counts on PCLKC/256
	0	0	0	1	1	1	Overflow/underflow of MTU2.TCNT
	0	0	1	x	x	x	Internal clock: counts on PCLKC/2
	0	1	0	x	x	x	Internal clock: counts on PCLKC/8
	0	1	1	x	x	x	Internal clock: counts on PCLKC/32
	1	0	0	x	x	x	Internal clock: counts on PCLKC/1024
	1	0	1	x	x	x	Setting prohibited
	1	1	0	x	x	x	Setting prohibited
	1	1	1	x	x	x	Setting prohibited

x: Don't care

Note: This setting has no effect when MTU1 is in phase counting mode.

Table 22.8 TPSC[2:0], TPSC2[2:0] (MTU2)

	TCR2 register			TCR register			
	Bit 2	Bit 1	Bit 0	Bit 2	Bit 1	Bit 0	
Channel	TPSC2[2]	TPSC2[1]	TPSC2[0]	TPSC[2]	TPSC[1]	TPSC[0]	Description
MTU2	0	0	0	0	0	0	Internal clock: counts on PCLKC/1
	0	0	0	0	0	1	Internal clock: counts on PCLKC/4
	0	0	0	0	1	0	Internal clock: counts on PCLKC/16
	0	0	0	0	1	1	Internal clock: counts on PCLKC/64
	0	0	0	1	0	0	External clock: counts on MTCLKA pin input
	0	0	0	1	0	1	External clock: counts on MTCLKB pin input
	0	0	0	1	1	0	External clock: counts on MTCLKC pin input
	0	0	0	1	1	1	Internal clock: counts on PCLKC/1024
	0	0	1	x	x	x	Internal clock: counts on PCLKC/2
	0	1	0	x	x	x	Internal clock: counts on PCLKC/8
	0	1	1	x	x	x	Internal clock: counts on PCLKC/32
	1	0	0	x	x	x	Internal clock: counts on PCLKC/256
	1	0	1	x	x	x	Setting prohibited
	1	1	0	x	x	x	Setting prohibited
	1	1	1	x	x	x	Setting prohibited

x: Don't care

Note: This setting has no effect when the MTU2 is in phase counting mode.

Table 22.9 TPSC[2:0], TPSC2[2:0] (MTU3, MTU4, MTU6, MTU7)

	TCR2 register			TCR register			
	Bit 2	Bit 1	Bit 0	Bit 2	Bit 1	Bit 0	
Channel	TPSC2[2]	TPSC2[1]	TPSC2[0]	TPSC[2]	TPSC[1]	TPSC[0]	Description
MTU3 MTU4 MTU6 MTU7	0	0	0	0	0	0	Internal clock: counts on PCLKC/1
	0	0	0	0	0	1	Internal clock: counts on PCLKC/4
	0	0	0	0	1	0	Internal clock: counts on PCLKC/16
	0	0	0	0	1	1	Internal clock: counts on PCLKC/64
	0	0	0	1	0	0	Internal clock: counts on PCLKC/256
	0	0	0	1	0	1	Internal clock: counts on PCLKC/1024
	0	0	0	1	1	0	External clock: counts on MTCLKA pin input
	0	0	0	1	1	1	External clock: counts on MTCLKB pin input
	0	0	1	x	x	x	Internal clock: counts on PCLKC/2
	0	1	0	x	x	x	Internal clock: counts on PCLKC/8
	0	1	1	x	x	x	Internal clock: counts on PCLKC/32
	1	0	0	x	x	x	Setting prohibited
	1	0	1	x	x	x	Setting prohibited
	1	1	0	x	x	x	Setting prohibited
	1	1	1	x	x	x	Setting prohibited

x: Don't care

Table 22.10 TPSC[1:0], TPSC2[2:0] (MTU5)

	TCR2 register			TCR register		
	Bit 2	Bit 1	Bit 0	Bit 1	Bit 0	
Channel	TPSC2[2]	TPSC2[1]	TPSC2[0]	TPSC[1]	TPSC[0]	Description
MTU5	0	0	0	0	0	Internal clock: counts on PCLKC/1
	0	0	0	0	1	Internal clock: counts on PCLKC/4
	0	0	0	1	0	Internal clock: counts on PCLKC/16
	0	0	0	1	1	Internal clock: counts on PCLKC/64
	0	0	1	x	x	Internal clock: counts on PCLKC/2
	0	1	0	x	x	Internal clock: counts on PCLKC/8
	0	1	1	x	x	Internal clock: counts on PCLKC/32
	1	0	0	x	x	Internal clock: counts on PCLKC/256
	1	0	1	x	x	Internal clock: counts on PCLKC/1024
	1	1	0	x	x	Setting prohibited
	1	1	1	x	x	External clock: counts on MTIOC1A pin input

x: Don't care

22.2.3 Timer Mode Register 1 (TMDR1)

- MTU0.TMDR1, MTU9.TMDR1

Address(es): MTU0.TMDR1 000C 1301h, MTU9.TMDR1 000C 1581h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	BFE	BFB	BFA	MD[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

- MTU1.TMDR1, MTU2.TMDR1

Address(es): MTU1.TMDR1 000C 1381h, MTU2.TMDR1 000C 1401h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	MD[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

- MTU3.TMDR1, MTU4.TMDR1, MTU6.TMDR1, MTU7.TMDR1

Address(es): MTU3.TMDR1 000C 1202h, MTU4.TMDR1 000C 1203h, MTU6.TMDR1 000C 1A02h, MTU7.TMDR1 000C 1A03h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	BFB	BFA	MD[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MD[3:0]	Mode Select	These bits specify the timer operating mode. Refer to Table 22.11 for details.	R/W
b4	BFA	Buffer Operation A	0: TGRA and TGRC operate normally 1: TGRA and TGRC used together for buffer operation	R/W
b5	BFB	Buffer Operation B	0: TGRB and TGRD operate normally 1: TGRB and TGRD used together for buffer operation	R/W
b6	BFE	Buffer Operation E	0: MTU0.TGRE and MTU0.TGRF, and MTU9.TGRE and MTU9.TGRF operate normally 1: MTU0.TGRE and MTU0.TGRF, and MTU9.TGRE and MTU9.TGRF used together for buffer operation	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

The TMDR1 register specifies the operating mode of each channel. The MTU has a total of eight TMDR1 registers, one each for MTU0 to MTU4, MTU6, MTU7, and MTU9. TMDR1 register values should be specified only while TCNT operation is stopped.

Table 22.11 Operating Mode Setting by MD[3:0] Bits (MTU0 to MTU4, MTU6, MTU7, and MTU9)

Bit 3	Bit 2	Bit 1	Bit 0										
MD[3]	MD[2]	MD[1]	MD[0]	Description	MTU0	MTU1	MTU2	MTU1 & MTU2 (LWA = 1)	MTU3	MTU4	MTU6	MTU7	MTU9
0	0	0	0	Normal mode	✓	✓	✓		✓	✓	✓	✓	✓
0	0	0	1	Setting prohibited									
0	0	1	0	PWM mode 1	✓	✓	✓		✓	✓	✓	✓	✓
0	0	1	1	PWM mode 2	✓	✓	✓						✓
0	1	0	0	Phase counting mode 1		✓	✓	✓					
0	1	0	1	Phase counting mode 2		✓	✓	✓					
0	1	1	0	Phase counting mode 3		✓	✓	✓					
0	1	1	1	Phase counting mode 4		✓	✓	✓					
1	0	0	0	Reset-synchronized PWM mode*1					✓		✓		
1	0	0	1	Phase counting mode 5		✓	✓	✓					
1	0	1	x	Setting prohibited									
1	1	0	0	Setting prohibited									
1	1	0	1	Complementary PWM mode 1 (transfer at crest)*1					✓		✓		
1	1	1	0	Complementary PWM mode 2 (transfer at trough)*1					✓		✓		
1	1	1	1	Complementary PWM mode 3 (transfer at crest and trough)*1					✓		✓		

x: Don't care

Note: Only set the corresponding operating mode listed above for each channel.

Note 1. Reset-synchronized PWM mode and complementary PWM mode can only be set for MTU3 and MTU6.
When MTU3 or MTU6 is set to reset-synchronized PWM mode or complementary PWM mode, the MTU4 or MTU7 settings become ineffective and automatically conform to the MTU3 or MTU6 setting, respectively. MTU4 and MTU7 should be set to the initial values (normal mode).

BFA Bit (Buffer Operation A)

This bit specifies whether to operate TGRA in the normal way or to use TGRA and TGRC together for buffer operation. When TGRC is used as a buffer register, TGRC input capture/output compare does not take place in modes other than complementary PWM mode, but compare match with TGRC occurs in complementary PWM mode.

If a compare match occurs on MTU4 in the Tb interval in complementary PWM mode, the TGIEC bit in timer interrupt enable register (MTU4.TIER) should be set to 0.

In reset-synchronized PWM mode or complementary PWM mode, buffer operation in MTU3 and MTU4 (or MTU6 and MTU7) depends on the settings in the BFA bit of MTU3.TMDR1 (MTU6.TMDR1). The BFA bit of MTU4.TMDR1 (MTU7.TMDR1) should be set to 0.

In MTU1 and MTU2, which have no TGRC, this bit is reserved. It is read as 0. The write value should be 0.

Refer to Figure 22.49 for an illustration of the Tb interval in complementary PWM mode.

BFB Bit (Buffer Operation B)

This bit specifies whether to operate TGRB in the normal way or to use TGRB and TGRD together for buffer operation. When TGRD is used as a buffer register, TGRD input capture/output compare does not take place in modes other than complementary PWM mode, but compare match with TGRD occurs in complementary PWM mode.

If a compare match occurs on MTU4 in the Tb interval in complementary PWM mode, the TGIED bit in timer interrupt enable register (MTU4.TIER) should be set to 0.

In reset-synchronized PWM mode or complementary PWM mode, buffer operation in MTU3 and MTU4 (or MTU6 and MTU7) depends on the settings in the BFB bit of MTU3.TMDR1 (MTU6.TMDR1). The BFB bit of MTU4.TMDR1

(MTU7.TMDR1) should be set to 0.
In MTU1 and MTU2, which have no TGRD, this bit is reserved. It is read as 0. The write value should be 0.
Refer to Figure 22.49 for an illustration of the Tb interval in complementary PWM mode.

BFE Bit (Buffer Operation E)

This bit specifies whether to operate MTU0.TGRE and MTU0.TGRF, and MTU9.TGRE and MTU9.TGRF in the normal way or to use them together for buffer operation. Compare match with TGRF occurs even when TGRF is used as a buffer register.
In MTU0 to MTU4, MTU6, MTU7, this bit is reserved. It is read as 0. The write value should be 0.

22.2.4 Timer Mode Register 2m (TMDR2m) (m = A, B)

Address(es): MTU.TMDR2A 000C 1270h, MTU.TMDR2B 000C 1A70h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DRS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DRS	Double Buffer Select	0: Double buffer function is disabled 1: Double buffer function is enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TMDR2A and TMDR2B specify the double buffer function in complementary PWM mode 3 (transfer at the crest and trough of the counter value). The MTU has two TMDR2m registers: one each for MTU3 (TMDR2A) and MTU6 (TMDR2B). TMDR2A and TMDR2B values should be specified only while TCNT operation is stopped.

DRS Bit (Double Buffer Select)

This bit enables or disables the double buffer function in complementary PWM mode.

22.2.5 Timer Mode Register 3 (TMDR3)

Address(es): MTU1.TMDR3 000C 1391h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PHCKSEL	LWA
Value after reset:	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	LWA	MTU1/MTU2 Combination Longword Access Control	0: 16-bit access is enabled. 1: 32-bit access is enabled.	R/W
b1	PHCKSEL	External Input Phase Clock Select	0: MTCLKA and MTCLKB are selected for the external phase clock. 1: MTCLKC and MTCLKD are selected for the external phase clock.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The TMDR3 register controls longword access to a 32-bit register or counter in a combination of MTU1 and MTU2. There is only one TMDR3 register in MTU1. The counter (TCNTLW), general register A (TGRALW), and general register B (TGRBLW) of MTU1 and MTU2 are accessed in the combinations listed in Table 22.12.

LWA Bit (MTU1/MTU2 Combination Longword Access Control)

This bit selects a 32-bit access in a combination of MTU1 and MTU2.

When LWA is set to 0, the MTU1 and MTU2 independently operate as a 16-bit timer. therefore registers TCNTLW, TGRALW, and TGRBLW cannot be accessed.

When LWA is set to 1, MTU1 and MTU2 operate as a 32-bit cascaded timer and the timer is controlled by registers MTU1.TCR, MTU1.TCR2, MTU1.TIOR, and MTU1.TMDR1. The settings of registers MTU2.TCR, MTU2.TCR2, MTU2.TIOR, and MTU2.TMDR1 are disabled and the 16-bit registers (TCNT, TGRA, and TGRB) in MTU1 and MTU2 cannot be accessed. Furthermore, MTU2 input capture and compare match are also disabled, which in turn disables any linked operation with the ELC.

The cascaded connection of MTU1 and MTU2 with the LWA bit set to 1 can only be used in phase counting mode, but not in normal mode, PWM1 mode, or PWM2 mode. Select phase counting mode when setting the LWA bit to 1.

Initialize the registers TCNT, TGRA, and TGRB in MTU1 and MTU2 in advance before setting the LWA bit to 1.

PHCKSEL Bit (External Input Phase Clock Select)

When the MTU1 and MTU2 registers are combined for 32-bit phase counting mode or MTU2 phase counting mode, this bit selects either the A- or B-phase signal from the external clock. Refer to Table 22.66, Clock Input Pins in Phase Counting Mode for details.

Table 22.12 Setting and Combination of the TMDR3 Register

	TMDR3.LWA = 0		TMDR3.LWA = 1	
Register	**Symbol**	**Access mode**	**Symbol**	**Access mode**
Counter in MTU1*1	MTU1.TCNT	Word	MTU1.TCNTLW	Longword
Counter in MTU2	MTU2.TCNT	Word		
General register A in MTU1	MTU1.TGRA	Word	MTU1.TGRALW	Longword
General register A in MTU2	MTU2.TGRA	Word		
General register B in MTU1	MTU1.TGRB	Word	MTU1.TGRBLW	Longword
General register B in MTU2	MTU2.TGRB	Word		

Note 1. When the LWA bit is set to 1, setting the count clock for MTU1 as MTU2.TCNT overflow/underflow is not required.

22.2.6 Timer I/O Control Register (TIOR)

- MTU0.TIORH, MTU1.TIOR, MTU2.TIOR, MTU3.TIORH, MTU4.TIORH, MTU6.TIORH, MTU7.TIORH, MTU9.TIORH

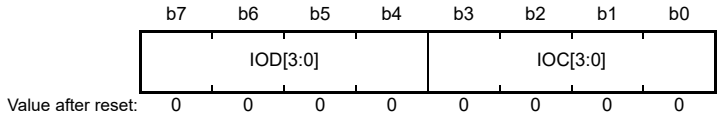
Address(es): MTU0.TIORH 000C 1302h, MTU1.TIOR 000C 1382h, MTU2.TIOR 000C 1402h, MTU3.TIORH 000C 1204h, MTU4.TIORH 000C 1206h, MTU6.TIORH 000C 1A04h, MTU7.TIORH 000C 1A06h, MTU9.TIORH 000C 1582h

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	IOA[3:0]	I/O Control A*1	Refer to the following tables. MTU0.TIORH: Table 22.27 MTU1.TIOR: Table 22.29 MTU2.TIOR: Table 22.30 MTU3.TIORH: Table 22.31 MTU4.TIORH: Table 22.33 MTU6.TIORH: Table 22.35 MTU7.TIORH: Table 22.37 MTU9.TIORH: Table 22.39	R/W
b7 to b4	IOB[3:0]	I/O Control B*1	Refer to the following tables. MTU0.TIORH: Table 22.13 MTU1.TIOR: Table 22.15 MTU2.TIOR: Table 22.16 MTU3.TIORH: Table 22.17 MTU4.TIORH: Table 22.19 MTU6.TIORH: Table 22.21 MTU7.TIORH: Table 22.23 MTU9.TIORH: Table 22.25	R/W

Note 1. When the value of IOm[3:0] (m = A, B) is changed to the output-prohibited state (0000b or 0100b) during low, high, or toggle output at compare match, this register is in Hi-Z.

- MTU0.TIORL, MTU3.TIORL, MTU4.TIORL, MTU6.TIORL, MTU7.TIORL, MTU9.TIORL

Address(es): MTU0.TIORL 000C 1303h, MTU3.TIORL 000C 1205h, MTU4.TIORL 000C 1207h, MTU6.TIORL 000C 1A05h, MTU7.TIORL 000C 1A07h, MTU9.TIORL 000C 1583h

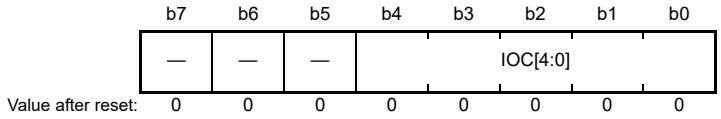


Bit	Symbol	Bit Name	Description	R/W
b3 to b0	IOC[3:0]	I/O Control C*1	Refer to the following tables. MTU0.TIORL: Table 22.28 MTU3.TIORL: Table 22.32 MTU4.TIORL: Table 22.34 MTU6.TIORL: Table 22.36 MTU7.TIORL: Table 22.38 MTU9.TIORL: Table 22.40	R/W
b7 to b4	IOD[3:0]	I/O Control D*1	Refer to the following tables. MTU0.TIORL: Table 22.14 MTU3.TIORL: Table 22.18 MTU4.TIORL: Table 22.20 MTU6.TIORL: Table 22.22 MTU7.TIORL: Table 22.24 MTU9.TIORL: Table 22.26	R/W

Note 1. When the value of IOm[3:0] (m = C, D) is changed to the output-prohibited state (0000b or 0100b) during low, high, or toggle output at compare match, this register is in Hi-Z.

- MTU5.TIORU, MTU5.TIORV, MTU5.TIORW

Address(es): MTU5.TIORU 000C 1C86h, MTU5.TIORV 000C 1C96h, MTU5.TIORW 000C 1CA6h



Bit	Symbol	Bit Name	Description	R/W
b4 to b0	IOC[4:0]	I/O Control C	Refer to the following table. MTU5.TIORU, MTU5.TIORV, MTU5.TIORW: Table 22.41	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The TIOR register controls the TGRm register. The MTU has a total of 17 TIOR registers, two each for MTU0, MTU3, MTU4, MTU6, MTU7, and MTU9, one each for MTU1 and MTU2, and three (MTU5.TIORU/TIORV/TIORW) for MTU5. The TIOR register should be set when the TMDR register setting is normal mode, PWM mode, or phase counting mode.

Note that TIOR is affected by the TMDR1 setting.

The initial output specified by TIOR is valid when the counter is stopped (the CSTn bit in TSTRA and the CSTn bit in TSTRB are set to 0). Note also that, in PWM mode 2, the output at the point at which the counter becomes 0000h is specified.

When TGRC or TGRD is designated for buffer operation, this setting is invalid and the register operates as a buffer register.

Table 22.13 TIORH (MTU0)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	TGRB Register Function	MTIOC0B Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	x	x		Capture input source is the clock source for counting in MTU1. Input capture on counting up or down by MTU1.TCNT (LWA = 0) or MTU1.TCNTLW (LWA = 1).*1

x: Don't care

Note 1. When PCLKC/1 is selected as the count clock for MTU1, MTU0 input capture is not generated. Do not select PCLKC/1 as the count clock for MTU1.

Table 22.14 TIORL (MTU0)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOD[3]	IOD[2]	IOD[1]	IOD[0]	TGRD Register Function	MTIOC0D Pin Function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register*1	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	x	x		Capture input source is the clock source for counting in MTU1. Input capture on counting up or down by MTU1.TCNT (LWA = 0) or MTU1.TCNTLW (LWA = 1).*2

x: Don't care

Note 1. When the MTU0.TMDR1.BFB bit is set to 1 and MTU0.TGRD register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Note 2. When PCLKC/1 is selected as the count clock for MTU1, MTU0 input capture is not generated. Do not select PCLKC/1 as the count clock for MTU1.

Table 22.15 TIOR (MTU1)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	TGRB/TGRBLW Register Function	MTIOC1B Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	0	x		Input capture at occurrence of compare match or input capture in the MTU0.TGRC register

x: Don't care

Table 22.16 TIOR (MTU2)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	TGRB Register Function	MTIOC2B Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Table 22.17 TIORH (MTU3)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	TGRB Register Function	MTIOC3B Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Table 22.18 TIORL (MTU3)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOD[3]	IOD[2]	IOD[1]	IOD[0]	TGRD Register Function	MTIOC3D Pin Function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Note 1. When the MTU3.TMDR1.BFB bit is set to 1 and MTU3.TGRD register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 22.19 TIORH (MTU4)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	TGRB Register Function	MTIOC4B Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Table 22.20 TIORL (MTU4)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOD[3]	IOD[2]	IOD[1]	IOD[0]	TGRD Register Function	MTIOC4D Pin Function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Note 1. When the MTU4.TMDR1.BFB bit is set to 1 and MTU4.TGRD register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 22.21 TIORH (MTU6)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	TGRB Register Function	MTIOC6B Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Table 22.22 TIORL (MTU6)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOD[3]	IOD[2]	IOD[1]	IOD[0]	TGRD Register Function	MTIOC6D Pin Function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Note 1. When the MTU6.TMDR1.BFB bit is set to 1 and MTU6.TGRD register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 22.23 TIORH (MTU7)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	TGRB Register Function	MTIOC7B Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Table 22.24 TIORL (MTU7)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOD[3]	IOD[2]	IOD[1]	IOD[0]	TGRD Register Function	MTIOC7D Pin Function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Note 1. When the MTU7.TMDR1.BFB bit is set to 1 and MTU7.TGRD register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 22.25 TIORH (MTU9)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	TGRB Register Function	MTIOC9B Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	x	x		Capture input source is the clock source for counting in MTU2. Input capture on counting up or down by MTU2.TCNT.*1

x: Don't care

Note 1. When PCLKC/1 is selected as the count clock for MTU2, MTU9 input capture is not generated. Do not select PCLKC/1 as the count clock for MTU2.

Table 22.26 TIORL (MTU9)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOD[3]	IOD[2]	IOD[1]	IOD[0]	TGRD Register Function	MTIOC9D Pin Function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register*1	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	x	x		Capture input source is the clock source for counting in MTU2. Input capture on counting up or down by MTU2.TCNT.*2

x: Don't care

Note 1. When the MTU9.TMDR1.BFB bit is set to 1 and the MTU9.TGRD register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Note 2. When PCLKC/1 is selected as the count clock for MTU2, MTU9 input capture is not generated. Do not select PCLKC/1 as the count clock for MTU2.

Table 22.27 TIORH (MTU0)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	TGRA Register Function	MTIOC0A Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	0	x		Capture input source is the clock source for counting in MTU1. Input capture on counting up or down by MTU1.TCNT (LWA = 0) or MTU1.TCNTLW (LWA = 1).*1

x: Don't care

Note 1. When PCLKC/1 is selected as the count clock for MTU1, MTU0 input capture is not generated. Do not select PCLKC/1 as the count clock for MTU1.

Table 22.28 TIORL (MTU0)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC[3]	IOC[2]	IOC[1]	IOC[0]	TGRC Register Function	MTIOC0C Pin Function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register*1	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	x	x		Capture input source is the clock source for counting in MTU1. Input capture on counting up or down by MTU1.TCNT (LWA = 0) or MTU1.TCNTLW (LWA = 1).*2

x: Don't care

Note 1. When the MTU0.TMDR1.BFB bit is set to 1 and MTU0.TGRC register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Note 2. When PCLKC/1 is selected as the count clock for MTU1, MTU0 input capture is not generated. Do not select PCLKC/1 as the count clock for MTU1.

Table 22.29 TIOR (MTU1)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	TGRA/TGRALW Register Function	MTIOC1A Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	x	x		Input capture at generation of MTU0.TGRA compare match/input capture.

x: Don't care

Table 22.30 TIOR (MTU2)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	TGRA Register Function	MTIOC2A Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Table 22.31 TIORH (MTU3)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	TGRA Register Function	MTIOC3A Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Table 22.32 TIORL (MTU3)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC[3]	IOC[2]	IOC[1]	IOC[0]	TGRC Register Function	MTIOC3C Pin Function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Note 1. When the MTU3.TMDR1.BFB bit is set to 1 and MTU3.TGRC register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 22.33 TIORH (MTU4)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	TGRA Register Function	MTIOC4A Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Table 22.34 TIORL (MTU4)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC[3]	IOC[2]	IOC[1]	IOC[0]	TGRC Register Function	MTIOC4C Pin Function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Note 1. When the MTU4.TMDR1.BFB bit is set to 1 and MTU4.TGRC register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 22.35 TIORH (MTU6)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	TGRA Register Function	MTIOC6A Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Table 22.36 TIORL (MTU6)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC[3]	IOC[2]	IOC[1]	IOC[0]	TGRC Register Function	MTIOC6C Pin Function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Note 1. When the MTU6.TMDR1.BFB bit is set to 1 and MTU6.TGRC register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 22.37 TIORH (MTU7)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	TGRA Register Function	MTIOC7A Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Table 22.38 TIORL (MTU7)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC[3]	IOC[2]	IOC[1]	IOC[0]	TGRC Register Function	MTIOC7C Pin Function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Note 1. When the MTU7.TMDR1.BFB bit is set to 1 and MTU7.TGRC register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 22.39 TIORH (MTU9)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	TGRA Register Function	MTIOC9A Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	0	x		Capture input source is the clock source for counting in MTU2. Input capture on counting up or down by MTU2.TCNT.*1

x: Don't care

Note 1. When PCLKC/1 is selected as the count clock for MTU2, MTU9 input capture is not generated. Do not select PCLKC/1 as the count clock for MTU2.

Table 22.40 TIORL (MTU9)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC[3]	IOC[2]	IOC[1]	IOC[0]	TGRC Register Function	MTIOC9C Pin Function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register*1	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	x	x		Capture input source is the clock source for counting in MTU2. Input capture on counting up or down by MTU2.TCNT.*2

x: Don't care

Note 1. When the MTU9.TMDR1.BFA bit is set to 1 and the MTU9.TGRC register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Note 2. When PCLKC/1 is selected as the count clock for MTU2, MTU9 input capture is not generated. Do not select PCLKC/1 as the count clock for MTU2.

Table 22.41 TIORU, TIORV, and TIORW (MTU5)

Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC[4]	IOC[3]	IOC[2]	IOC[1]	IOC[0]	TGRU, TGRV, TGRW Registers Function	MTIC5U, MTIC5V, MTIC5W Pin Function
0	0	0	0	0	Output compare register	No function
0	0	0	0	1		Setting prohibited
0	0	0	1	x		Setting prohibited
0	0	1	x	x		Setting prohibited
0	1	x	x	x		Setting prohibited
1	0	0	0	0	Input capture register*1	Setting prohibited
1	0	0	0	1		Input capture at rising edge.
1	0	0	1	0		Input capture at falling edge.
1	0	0	1	1		Input capture at both edges.
1	0	1	x	x		Setting prohibited
1	1	0	0	0		Setting prohibited
1	1	0	0	1		Measurement of low pulse width of external input signal. Capture at trough in complementary PWM mode.
1	1	0	1	0		Measurement of low pulse width of external input signal. Capture at crest of complementary PWM mode.
1	1	0	1	1		Measurement of low pulse width of external input signal. Capture at crest and trough of complementary PWM mode.
1	1	1	0	0		Setting prohibited
1	1	1	0	1		Measurement of high pulse width of external input signal. Capture at trough in complementary PWM mode.
1	1	1	1	0		Measurement of high pulse width of external input signal. Capture at crest of complementary PWM mode.
1	1	1	1	1		Measurement of high pulse width of external input signal. Capture at crest and trough of complementary PWM mode.

x: Don't care

Note 1. Set the IOC[4:0] bits to 19h, 1Ah, 1Bh, 1Dh, 1Eh, or 1Fh only when using external pulse width measurement or only when using dead time compensation linked with MTU6 and MTU7. For details, refer to section 22.3.11, External Pulse Width Measurement and section 22.3.12, Dead Time Compensation.

22.2.7 Timer Compare Match Clear Register (TCNTCMPCLR)

Address(es): MTU5.TCNTCMPCLR 000C 1CB6h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	CMPCLR5U	CMPCLR5V	CMPCLR5W
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMPCLR5W	TCNT Compare Clear 5W	0: Disables MTU5.TCNTW to be cleared to 0000h at MTU5.TCNTW and MTU5.TGRW compare match or input capture 1: Enables MTU5.TCNTW to be cleared to 0000h at MTU5.TCNTW and MTU5.TGRW compare match or input capture	R/W
b1	CMPCLR5V	TCNT Compare Clear 5V	0: Disables MTU5.TCNTV to be cleared to 0000h at MTU5.TCNTV and MTU5.TGRV compare match or input capture 1: Enables MTU5.TCNTV to be cleared to 0000h at MTU5.TCNTV and MTU5.TGRV compare match or input capture	R/W
b2	CMPCLR5U	TCNT Compare Clear 5U	0: Disables MTU5.TCNTU to be cleared to 0000h at MTU5.TCNTU and MTU5.TGRU compare match or input capture 1: Enables MTU5.TCNTU to be cleared to 0000h at MTU5.TCNTU and MTU5.TGRU compare match or input capture	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TCNTCMPCLR specifies requests to clear MTU5.TCNTU, MTU5.TCNTV, and MTU5.TCNTW. The MTU has one TCNTCMPCLR (on MTU5).

22.2.8 Timer Interrupt Enable Register (TIER)

- MTU1.TIER, MTU2.TIER

Address(es): MTU1.TIER 000C 1384h, MTU2.TIER 000C 1404h

	b7	b6	b5	b4	b3	b2	b1	b0
	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
Value after reset:	0	0	0	0	0	0	0	0

- MTU0.TIER, MTU3.TIER, MTU6.TIER, MTU9.TIER

Address(es): MTU0.TIER 000C 1304h, MTU3.TIER 000C 1208h, MTU6.TIER 000C 1A08h, MTU9.TIER 000C 1584h

	b7	b6	b5	b4	b3	b2	b1	b0
	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
Value after reset:	0	0	0	0	0	0	0	0

- MTU4.TIER, MTU7.TIER

Address(es): MTU4.TIER 000C 1209h, MTU7.TIER 000C 1A09h

	b7	b6	b5	b4	b3	b2	b1	b0
	TTGE	TTGE2	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TGIEA	TGR Interrupt Enable A	0: Interrupt requests (TGIA) disabled 1: Interrupt requests (TGIA) enabled	R/W
b1	TGIEB	TGR Interrupt Enable B	0: Interrupt requests (TGIB) disabled 1: Interrupt requests (TGIB) enabled	R/W
b2	TGIEC	TGR Interrupt Enable C	0: Interrupt requests (TGIC) disabled 1: Interrupt requests (TGIC) enabled	R/W
b3	TGIED	TGR Interrupt Enable D	0: Interrupt requests (TGID) disabled 1: Interrupt requests (TGID) enabled	R/W
b4	TCIEV	Overflow Interrupt Enable	0: Interrupt requests (TCIV) disabled 1: Interrupt requests (TCIV) enabled	R/W
b5	TCIEU	Underflow Interrupt Enable	0: Interrupt requests (TCIU) disabled 1: Interrupt requests (TCIU) enabled	R/W
b6	TTGE2	A/D Conversion Start Request Enable 2	0: A/D conversion start request generation by MTUn.TCNT underflow (trough) disabled 1: A/D conversion start request generation by MTUn.TCNT underflow (trough) enabled	R/W
b7	TTGE	A/D Conversion Start Request Enable	0: A/D conversion start request generation disabled 1: A/D conversion start request generation enabled	R/W

n = 4, 7

The TIER register enables or disables interrupt requests from each channel. The MTU has a total of eleven TIER registers, two for MTU0 and MTU9 and one each for MTU1 to MTU7.

TGIEA and TGIEB Bits (TGR Interrupt Enable A and B)

Each bit enables or disables interrupt requests (TGIm) (m = A, B).

TGIEC and TGIED Bits (TGR Interrupt Enable C and D)

Each bit enables or disables an interrupt request (TGIm) (m = C, D).
In MTU1 and MTU2, these bits are reserved. They are read as 0. The write value should be 0.

TCIEV Bit (Overflow Interrupt Enable)

This bit enables or disables interrupt requests (TCIV).

TCIEU Bit (Underflow Interrupt Enable)

This bit enables or disables interrupt requests (TCIU).
In MTU0, MTU3, MTU4, MTU6, MTU7, and MTU9, this bit is reserved. It is read as 0. The write value should be 0.

TTGE2 Bit (A/D Conversion Start Request Enable 2)

This bit enables or disables generation of A/D conversion start requests by MTUn.TCNT underflow (trough) in complementary PWM mode (n = 4, 7).
In MTU0 to MTU3, MTU6, and MTU9, this bit is reserved. It is read as 0. The write value should be 0.

TTGE Bit (A/D Conversion Start Request Enable)

This bit enables or disables generation of A/D conversion start requests by TGRA input capture/compare match.

- MTU0.TIER2, MTU9.TIER2

Address(es): MTU0.TIER2 000C 1324h, MTU9.TIER2 000C 15A4h

	b7	b6	b5	b4	b3	b2	b1	b0
	TTGE2	—	—	—	—	—	TGIEF	TGIEE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TGIEE	TGR Interrupt Enable E	0: Interrupt requests (TGIE) disabled 1: Interrupt requests (TGIE) enabled	R/W
b1	TGIEF	TGR Interrupt Enable F	0: Interrupt requests (TGIF) disabled 1: Interrupt requests (TGIF) enabled	R/W
b6 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	TTGE2	A/D Conversion Start Request Enable 2	0: A/D conversion start request generation by compare match between MTU0.TCNT and MTU0.TGRE, and MTU9.TCNT and MTU9.TGRE disabled 1: A/D conversion start request generation by compare match between MTU0.TCNT and MTU0.TGRE, and MTU9.TCNT and MTU9.TGRE enabled	R/W

TGIEE and TGIEF Bits (TGR Interrupt Enable E and F)

Each bit enables or disables interrupt requests by compare match between MTU0.TCNT and MTU0.TGRm, and MTU9.TCNT and MTU9.TGRm (m = E, F).

TTGE2 Bit (A/D Conversion Start Request Enable 2)

Each bit enables or disables A/D conversion start requests by compare match between MTU0.TCNT and MTU0.TGRE, and MTU9.TCNT and MTU9.TGRE.

- MTU5.TIER

Address(es): MTU5.TIER 000C 1CB2h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TGIE5U	TGIE5V	TGIE5W
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TGIE5W	TGR Interrupt Enable 5W	0: Interrupt requests TGIW5 disabled 1: Interrupt requests TGIW5 enabled	R/W
b1	TGIE5V	TGR Interrupt Enable 5V	0: Interrupt requests TGIV5 disabled 1: Interrupt requests TGIV5 enabled	R/W
b2	TGIE5U	TGR Interrupt Enable 5U	0: Interrupt requests TGIU5 disabled 1: Interrupt requests TGIU5 enabled	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TGIE5m Bits (TGR Interrupt Enable 5m)

Each bit enables or disables interrupt requests (TGIm5) (m = U, V, W).

22.2.9 Timer Status Register (TSR)

- MTU1.TSR, MTU2.TSR

Address(es): MTU1.TSR 000C 1385h, MTU2.TSR 000C 1405h

	b7	b6	b5	b4	b3	b2	b1	b0
	TCFD	—	—	—	—	—	—	—
Value after reset:	1	1	0	0	0	0	0	0

- MTU3.TSR, MTU4.TSR, MTU6.TSR, MTU7.TSR

Address(es): MTU3.TSR 000C 122Ch, MTU4.TSR 000C 122Dh, MTU6.TSR 000C 1A2Ch, MTU7.TSR 000C 1A2Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	TCFD	—	—	—	—	—	—	—
Value after reset:	1	1	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	The read value is undefined.The write value should be 1.	R/W
b6	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b7	TCFD	Count Direction Flag	0: TCNT counts down 1: TCNT counts up	R

TSR indicates the states of each of the channels. The MTU has a total of six TSR registers, one each for MTU1 to MTU4, MTU6, and MTU7.

TCFD Flag (Count Direction Flag)

Status flag that indicates the direction in which TCNT is counting in MTU1 to MTU4, MTU6, and MTU7.

22.2.10 Timer Buffer Operation Transfer Mode Register (TBTM)

- MTU0.TBTM, MTU9.TBTM

Address(es): MTU0.TBTM 000C 1326h, MTU9.TBTM 000C 15A6h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TTSE	TTSB	TTSA
Value after reset:	0	0	0	0	0	0	0	0

- MTU3.TBTM, MTU4.TBTM, MTU6.TBTM, MTU7.TBTM

Address(es): MTU3.TBTM 000C 1238h, MTU4.TBTM 000C 1239h, MTU6.TBTM 000C 1A38h, MTU7.TBTM 000C 1A39h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	TTSB	TTSA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TTSA	Timing Select A	0: When compare match A occurs in each channel, data is transferred from TGRC to TGRA 1: When TCNT is cleared in each channel, data is transferred from TGRC to TGRA	R/W
b1	TTSB	Timing Select B	0: When compare match B occurs in each channel, data is transferred from TGRD to TGRB 1: When TCNT is cleared in each channel, data is transferred from TGRD to TGRB	R/W
b2	TTSE	Timing Select E	0: When compare match E occurs in MTU0 or MTU9, data is transferred from MTU0.TGRF to MTU0.TGRE or MTU9.TGRF to MTU9.TGRE 1: When MTU0.TCNT or MTU9.TCNT is cleared, data is transferred from MTU0.TGRF to MTU0.TGRE or MTU9.TGRF to MTU9.TGRE	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TBTM specifies the timing for transferring data from the buffer register to the timer general register in PWM mode. The MTU has a total of six TBTM registers, one each for MTU0, MTU3, MTU4, MTU6, MTU7, and MTU9.

TTSA Bit (Timing Select A)

This bit specifies the timing for transferring data from TGRC to TGRA in each channel when they are used together for buffer operation. When a channel is not set to PWM mode, do not set the TTSA bit in the channel to 1.

TTSB Bit (Timing Select B)

This bit specifies the timing for transferring data from TGRD to TGRB in each channel when they are used together for buffer operation. When a channel is not set to PWM mode, do not set the TTSB bit in the channel to 1.

TTSE Bit (Timing Select E)

This bit specifies the timing for transferring data from MTU0.TGRF to MTU0.TGRE or MTU9.TGRF to MTU9.TGRE when they are used together for buffer operation.
In MTU3, MTU4, MTU6, and MTU7, this bit is reserved. It is read as 0 and the write value should be 0. When a channel is not set to PWM mode, do not set the TTSE bit in the channel to 1.

22.2.11 Timer Input Capture Control Register (TICCR)

Address(es): MTU1.TICCR 000C 1390h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	I2BE	I2AE	I1BE	I1AE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	I1AE	Input Capture Enable	0: Does not include the MTIOC1A pin in the MTU2.TGRA input capture conditions 1: Includes the MTIOC1A pin in the MTU2.TGRA input capture conditions	R/W
b1	I1BE	Input Capture Enable	0: Does not include the MTIOC1B pin in the MTU2.TGRB input capture conditions 1: Includes the MTIOC1B pin in the MTU2.TGRB input capture conditions	R/W
b2	I2AE	Input Capture Enable	0: Does not include the MTIOC2A pin in the MTU1.TGRA input capture conditions 1: Includes the MTIOC2A pin in the MTU1.TGRA input capture conditions	R/W
b3	I2BE	Input Capture Enable	0: Does not include the MTIOC2B pin in the MTU1.TGRB input capture conditions 1: Includes the MTIOC2B pin in the MTU1.TGRB input capture conditions	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TICCR specifies input capture conditions when MTU1.TCNT and MTU2.TCNT are cascaded. The MTU has one TICCR for MTU1.

22.2.12 Timer Synchronous Clear Register (TSYCR)

Address(es): MTU6.TSYCR 000C 1A50h

	b7	b6	b5	b4	b3	b2	b1	b0
	CE0A	CE0B	CE0C	CE0D	CE1A	CE1B	CE2A	CE2B
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CE2B	Clear Enable 2B	0: Disables counter clearing by the MTU2.TGIB2 interrupt generation timing. 1: Enables counter clearing by the MTU2.TGIB2 interrupt generation timing.	R/W
b1	CE2A	Clear Enable 2A	0: Disables counter clearing by the MTU2.TGIA2 interrupt generation timing*1. 1: Enables counter clearing by the MTU2.TGIA2 interrupt generation timing*1.	R/W
b2	CE1B	Clear Enable 1B	0: Disables counter clearing by the MTU1.TGIB1 interrupt generation timing*1. 1: Enables counter clearing by the MTU1.TGIB1 interrupt generation timing*1.	R/W
b3	CE1A	Clear Enable 1A	0: Disables counter clearing by the MTU1.TGIA1 interrupt generation timing*1. 1: Enables counter clearing by the MTU1.TGIA1 interrupt generation timing*1.	R/W
b4	CE0D	Clear Enable 0D	0: Disables counter clearing by the MTU0.TGID0 interrupt generation timing*1. 1: Enables counter clearing by the MTU0.TGID0 interrupt generation timing*1.	R/W
b5	CE0C	Clear Enable 0C	0: Disables counter clearing by the MTU0.TGIC0 interrupt generation timing*1. 1: Enables counter clearing by the MTU0.TGIC0 interrupt generation timing*1.	R/W
b6	CE0B	Clear Enable 0B	0: Disables counter clearing by the MTU0.TGIB0 interrupt generation timing*1. 1: Enables counter clearing by the MTU0.TGIB0 interrupt generation timing*1.	R/W
b7	CE0A	Clear Enable 0A	0: Disables counter clearing by the MTU0.TGIA0 interrupt generation timing*1. 1: Enables counter clearing by the MTU0.TGIA0 interrupt generation timing*1.	R/W

Note 1. This does not depend on the TIERn.TGIEm bit setting. (n = 0, 1, 2; m = A, B, C, D)

TSYCR specifies synchronous clear conditions for MTU6.TCNT and MTU7.TCNT. The MTU has one TSYCR for MTU1.

CEnm Bits (Clear Enable nm; n = 0, 1, 2; m = A, B, C, D)

These bits enable or disable counter clearing by the MTUn.TGImn interrupt generation timing.

22.2.13 Timer Counter (TCNT)

Address(es): MTU0.TCNT 000C 1306h, MTU1.TCNT 000C 1386h, MTU2.TCNT 000C 1406h, MTU3.TCNT 000C 1210h, MTU4.TCNT 000C 1212h, MTU5.TCNTU 000C 1C80h, MTU5.TCNTV 000C 1C90h, MTU5.TCNTW 000C 1CA0h, MTU6.TCNT 000C 1A10h, MTU7.TCNT 000C 1A12h, MTU9.TCNT 000C 1586h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: TCNT must not be accessed in 8 bits; it should be accessed in 16 bits.

TCNT is a 16-bit readable/writable counter. The MTU has a total of 11 TCNT counters, one each for MTU0 to MTU4, MTU6, MTU7, and MTU9 and three (MTU5.TCNTU, TCNTV, and TCNTW) for MTU5. The TCNT counters in MTU0 to MTU4, MTU6, MTU7, and MTU9 are initialized to 0000h by a reset. MTU5.TCNTU, MTU5.TCNTV, and MTU5.TCNTW are initialized to 0000h by a reset.

In MTU0 to MTU4, MTU6, MTU7, and MTU9, the TCNT counters must not be accessed in 8-bit units; they should be accessed in 16-bit units.

The MTU1.TCNT and MTU2.TCNT counters are read as 0000h when TMDR3.LWA is 1. Refer to section 22.2.5, Timer Mode Register 3 (TMDR3) for details.

22.2.14 Timer Longword Counter (TCNTLW)

Address(es): MTU1.TCNTLW 000C 13A0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: TCNTLW must not be accessed in 8 or 16 bits; it should be accessed in 32 bits.

The TCNTLW counter is a 32-bit readable/writable counter. Only one counter of this type is provided, and is formed by combining MTU1.TCNT and MTU2.TCNT. Such operation is only effective when TMDR3.LWA is 1. The TCNTLW counter is initialized to 0000 0000h by a reset. This counter is read as 0000 0000h when TMDR3.LWA is 0. Refer to section 22.2.5, Timer Mode Register 3 (TMDR3) for details. This register can only be used in 32-bit phase counting mode.

22.2.15 Timer General Register m (TGRm) (m = A, B, C, D, E, F, U, V, W)

Address(es): MTU0.TGRA 000C 1308h, MTU0.TGRB 000C 130Ah, MTU0.TGRC 000C 130Ch, MTU0.TGRD 000C 130Eh, MTU0.TGRE 000C 1320h, MTU0.TGRF 000C 1322h,
MTU1.TGRA 000C 1388h, MTU1.TGRB 000C 138Ah,
MTU2.TGRA 000C 1408h, MTU2.TGRB 000C 140Ah,
MTU3.TGRA 000C 1218h, MTU3.TGRB 000C 121Ah, MTU3.TGRC 000C 1224h, MTU3.TGRD 000C 1226h, MTU3.TGRE 000C 1272h,
MTU4.TGRA 000C 121Ch, MTU4.TGRB 000C 121Eh, MTU4.TGRC 000C 1228h, MTU4.TGRD 000C 122Ah, MTU4.TGRE 000C 1274h, MTU4.TGRF 000C 1276h,
MTU5.TGRU 000C 1C82h, MTU5.TGRV 000C 1C92h, MTU5.TGRW 000C 1CA2h,
MTU6.TGRA 000C 1A18h, MTU6.TGRB 000C 1A1Ah, MTU6.TGRC 000C 1A24h, MTU6.TGRD 000C 1A26h, MTU6.TGRE 000C 1A72h,
MTU7.TGRA 000C 1A1Ch, MTU7.TGRB 000C 1A1Eh, MTU7.TGRC 000C 1A28h, MTU7.TGRD 000C 1A2Ah, MTU7.TGRE 000C 1A74h, MTU7.TGRF 000C 1A76h,
MTU9.TGRA 000C 1588h, MTU9.TGRB 000C 158Ah, MTU9.TGRC 000C 158Ch, MTU9.TGRD 000C 158Eh, MTU9.TGRE 000C 15A0h, MTU9.TGRF 000C 15A2h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Note: The TGRm registers must not be accessed in 8 bits; They should be accessed in 16 bits. The initial value of the TGRm register is FFFFh.

The TGRm register is 16-bit readable/writable register. The MTU has a total of 41 TGRm registers, six for MTU0 and MTU9, two each for MTU1 and MTU2, five each for MTU3 and MTU6, six each for MTU4 and MTU7, and three for MTU5.

The TGRA, TGRB, TGRC, and TGRD registers function as either output compare or input capture registers. The TGRC and TGRD registers for MTU0, MTU3, MTU4, MTU6, MTU7, and MTU9 can also be designated for operation as buffer registers. TGRm buffer register combinations are TGRA and TGRC, and TGRB and TGRD.

MTU0.TGRE and MTU0.TGRF, and MTU9.TGRE and MTU9.TGRF function as compare registers. When the MTU0.TCNT count matches the MTU0.TGRE value or the MTU9.TCNT count matches the MTU9.TGRE value, an A/D conversion start request can be issued. The TGRF register can also be designated for operation as a buffer register. TGRm buffer register combination is TGRE and TGRF. MTU5.TGRU, MTU5.TGRV, and MTU5.TGRW function as compare match, input capture, or external pulse width measurement registers.

The MTU1.TGRA, MTU2.TGRA, MTU1.TGRB, and MTU2.TGRB registers are read as 0000h when TMDR3.LWA is 1. Refer to section 22.2.5, Timer Mode Register 3 (TMDR3) for details.

22.2.16 Timer Longword General Register m (TGRmLW) (m = A, B)

Address(es): MTU1.TGRALW 000C 13A4h, MTU1.TGRBLW 000C 13A8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Note: TGRALW and TGRBLW must not be accessed in 8 or 16 bits; it should be accessed in 32 bits.

The TGRmLW register is a 32-bit readable/writable register. Two general registers of this type are provided, and are formed by combining MTU1.TGRm and MTU2.TGRm. Such operation is only effective when TMDR3.LWA is 1.
The TGRmLW register is initialized to FFFF FFFFh by a reset, but it is read as 0000 0000h when TMDR3.LWA is 0. Refer to section 22.2.5, Timer Mode Register 3 (TMDR3) for details.
The TGRALW and TGRBLW registers function as compare match or input capture registers which can only be used in 32-bit phase counting mode.

22.2.17 Timer Start Registers (TSTRA, TSTRB, TSTR)

- MTU.TSTRA (for MTU0, MTU1, MTU2, MTU3, MTU4, and MTU9)

Address(es): MTU.TSTRA 000C 1280h

	b7	b6	b5	b4	b3	b2	b1	b0
	CST4	CST3	—	CST9	—	CST2	CST1	CST0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CST0	Counter Start 0	0: MTU0.TCNT counting is stopped 1: MTU0.TCNT performs count operation	R/W
b1	CST1	Counter Start 1	0: MTU1.TCNT counting is stopped 1: MTU1.TCNT performs count operation	R/W
b2	CST2	Counter Start 2	0: MTU2.TCNT counting is stopped 1: MTU2.TCNT performs count operation	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	CST9	Counter start 9	0: MTU9.TCNT counting is stopped 1: MTU9.TCNT performs count operation	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	CST3	Counter Start 3	0: MTU3.TCNT counting is stopped 1: MTU3.TCNT performs count operation	R/W
b7	CST4	Counter Start 4	0: MTU4.TCNT counting is stopped 1: MTU4.TCNT performs count operation	R/W

Note: When 1 is written to a bit in TCSYSTR, the corresponding bit in TSTRA is also set to 1 automatically.

The TSTRA register starts or stops TCNT operation in MTU0 to MTU4 and MTU9.
TSTRB starts or stops TCNT operation in MTU6 and MTU7.
TSTR starts or stops TCNT operation in MTU5.
Before setting the operating mode in TMDR1 or setting the TCNT count clock in TCR, be sure to stop the TCNT counter.

CSTn Bits (Counter Start n) (n = 0, 1, 2, 3, 4, 9)

Each bit starts or stops TCNT in the corresponding channel.
If 0 is written to the CSTn bit during operation with the MTIOC pin designated for output, the counter stops. At this time, initial output level specified in the TOCR1A or TOCR2A register is output from the MTIOC pin in complementary PWM mode or reset-synchronized PWM mode. In any mode other than complementary PWM mode and reset synchronous PWM mode, the output compare signal level from the MTIOC pin is retained. If TIOR is written to while the CSTn bit is 0, the pin output level will be changed to the specified initial output value.

- MTU.TSTRB (for MTU6 and MTU7)

Address(es): MTU.TSTRB 000C 1A80h

	b7	b6	b5	b4	b3	b2	b1	b0
	CST7	CST6	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	CST6	Counter Start 6	0: MTU6.TCNT counting is stopped 1: MTU6.TCNT performs count operation	R/W
b7	CST7	Counter Start 7	0: MTU7.TCNT counting is stopped 1: MTU7.TCNT performs count operation	R/W

Note: When 1 is written to a bit in TCSYSTR, the corresponding bit in TSTRB is also set to 1 automatically.

CSTn Bits (Counter Start n) (n = 6, 7)

Each bit starts or stops TCNT in the corresponding channel.
If 0 is written to the CSTn bit during operation with the MTIOC pin designated for output, the counter stops. At this time the MTIOC pin output the initial output level set in the TOCR1B or TOCR2B register in complementary PWM mode or reset-synchronized PWM mode, but the output compare signal level from the MTIOC pin is retained in the other modes.
If TIOR is written to while the CSTn bit is 0, the pin output level will be changed to the specified initial output value.

- MTU5.TSTR

Address(es): MTU5.TSTR 000C 1CB4h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	CSTU5	CSTV5	CSTW5
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CSTW5	Counter Start W5	0: MTU5.TCNTW counting is stopped 1: MTU5.TCNTW performs count operation	R/W
b1	CSTV5	Counter Start V5	0: MTU5.TCNTV counting is stopped 1: MTU5.TCNTV performs count operation	R/W
b2	CSTU5	Counter Start U5	0: MTU5.TCNTU counting is stopped 1: MTU5.TCNTU performs count operation	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

22.2.18 Timer Synchronous Register m (TSYRm) (m = A, B)

- MTU.TSYRA (for MTU0 to MTU4 and MTU9)

Address(es): MTU.TSYRA 000C 1281h

	b7	b6	b5	b4	b3	b2	b1	b0
	SYNC4	SYNC3	—	—	SYNC9	SYNC2	SYNC1	SYNC0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SYNC0	Timer Synchronous Operation 0	0: MTU0.TCNT operates independently (TCNT setting/clearing is not related to other channels). 1: MTU0.TCNT performs synchronous operation. (TCNT synchronous setting/synchronous clearing is enabled.)	R/W
b1	SYNC1	Timer Synchronous Operation 1	0: MTU1.TCNT operates independently (TCNT setting/clearing is not related to other channels). 1: MTU1.TCNT performs synchronous operation. (TCNT synchronous setting/synchronous clearing is enabled.)	R/W
b2	SYNC2	Timer Synchronous Operation 2	0: MTU2.TCNT operates independently (TCNT setting/clearing is not related to other channels). 1: MTU2.TCNT performs synchronous operation. (TCNT synchronous setting/synchronous clearing is enabled.)	R/W
b3	SYNC9	Timer Synchronous Operation 9	0: MTU9.TCNT operates independently (TCNT setting/clearing is not related to other channels). 1: MTU9.TCNT performs synchronous operation. (TCNT synchronous setting/synchronous clearing is enabled.)	R/W
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	SYNC3	Timer Synchronous Operation 3	0: MTU3.TCNT operates independently (TCNT setting/clearing is not related to other channels). 1: MTU3.TCNT performs synchronous operation. (TCNT synchronous setting/synchronous clearing is enabled.)	R/W
b7	SYNC4	Timer Synchronous Operation 4	0: MTU4.TCNT operates independently (TCNT setting/clearing is not related to other channels). 1: MTU4.TCNT performs synchronous operation. (TCNT synchronous setting/synchronous clearing is enabled.)	R/W

TSYRA selects independent operation or synchronous operation of TCNT in MTU0 to MTU4 and MTU9.
TSYRB selects independent operation or synchronous operation of TCNT in MTU6 and MTU7.
A channel performs synchronous operation when the corresponding bit in TSYR is set to 1.

SYNCn Bits (Timer Synchronous Operation n) (n = 0, 1, 2, 3, 4, 9)

Each bit selects whether operation is independent of or synchronized with other channels.
When synchronous operation is selected, the TCNT synchronous setting of multiple channels and synchronous clearing by counter clearing on another channel are possible.
To set synchronous operation, the SYNC bits for at least two channels must be set to 1. To set synchronous clearing, in addition to the SYNC bit, the TCNT clearing source must also be set by means of the TCR.CCLR[2:0] bits.

- MTU.TSYRB (for MTU6 and MTU7)

Address(es): MTU.TSYRB 000C 1A81h

	b7	b6	b5	b4	b3	b2	b1	b0
	SYNC7	SYNC6	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	SYNC6	Timer Synchronous Operation 6	0: MTU6.TCNT operates independently (TCNT setting/clearing is not related to other channels). 1: MTU6.TCNT performs synchronous operation. (TCNT synchronous setting/synchronous clearing is enabled.)	R/W
b7	SYNC7	Timer Synchronous Operation 7	0: MTU7.TCNT operates independently (TCNT setting/clearing is not related to other channels). 1: MTU7.TCNT performs synchronous operation. (TCNT synchronous setting/synchronous clearing is enabled.)	R/W

SYNCn Bits (Timer Synchronous Operation n) (n = 6, 7)

Each bit selects whether operation is independent of or synchronized with other channels.

When synchronous operation is selected, the TCNT synchronous setting of multiple channels and synchronous clearing by counter clearing on another channel are possible.

To set synchronous operation, the SYNC bits for at least two channels must be set to 1. To set synchronous clearing, in addition to the SYNC bit, the TCNT clearing source must also be set by means of bits TCR.CCLR[2:0].

22.2.19 Timer Counter Synchronous Start Register (TCSYSTR)

Address(es): MTU.TCSYSTR 000C 1282h

	b7	b6	b5	b4	b3	b2	b1	b0
	SCH0	SCH1	SCH2	SCH3	SCH4	SCH9	SCH6	SCH7
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SCH7	Synchronous Start 7	0: Does not specify synchronous start for MTU7.TCNT 1: Specifies synchronous start for MTU7.TCNT	R/(W)*1
b1	SCH6	Synchronous Start 6	0: Does not specify synchronous start for MTU6.TCNT 1: Specifies synchronous start for MTU6.TCNT	R/(W)*1
b2	SCH9	Synchronous Start 9	0: Does not specify synchronous start for MTU9.TCNT 1: Specifies synchronous start for MTU9.TCNT	R/(W)*1
b3	SCH4	Synchronous Start 4	0: Does not specify synchronous start for MTU4.TCNT 1: Specifies synchronous start for MTU4.TCNT	R/(W)*1
b4	SCH3	Synchronous Start 3	0: Does not specify synchronous start for MTU3.TCNT 1: Specifies synchronous start for MTU3.TCNT	R/(W)*1
b5	SCH2	Synchronous Start 2	0: Does not specify synchronous start for MTU2.TCNT 1: Specifies synchronous start for MTU2.TCNT	R/(W)*1
b6	SCH1	Synchronous Start 1	0: Does not specify synchronous start for MTU1.TCNT 1: Specifies synchronous start for MTU1.TCNT	R/(W)*1
b7	SCH0	Synchronous Start 0	0: Does not specify synchronous start for MTU0.TCNT 1: Specifies synchronous start for MTU0.TCNT	R/(W)*1

Note 1. Only 1 can be written to this bit. This bit is automatically cleared when the corresponding counter starts.

TCSYSTR specifies synchronous start of the counters.

SCH7 Bit (Synchronous Start 7)

This bit controls synchronous start of MTU7.TCNT.

[Clearing condition]

- When 1 is set to the TSTRB.CST7 bit while SCH7 = 1

SCH6 Bit (Synchronous Start 6)

This bit controls synchronous start of MTU6.TCNT.

[Clearing condition]

- When 1 is set to the TSTRB.CST6 bit while SCH6 = 1

SCH9 Bit (Synchronous Start 9)

This bit controls synchronous start of MTU9.TCNT.

[Clearing condition]

- When 1 is set to the TSTRA.CST9 bit while SCH9 = 1

SCH4 Bit (Synchronous Start 4)

This bit controls synchronous start of MTU4.TCNT.

[Clearing condition]

- When 1 is set to the TSTRA.CST4 bit while SCH4 = 1

SCH3 Bit (Synchronous Start 3)

This bit controls synchronous start of MTU3.TCNT.
[Clearing condition]

- When 1 is set to the TSTRA.CST3 bit while SCH3 = 1

SCH2 Bit (Synchronous Start 2)

This bit controls synchronous start of MTU2.TCNT.
[Clearing condition]

- When 1 is set to the TSTRA.CST2 bit while SCH2 = 1

SCH1 Bit (Synchronous Start 1)

This bit controls synchronous start of MTU1.TCNT.
[Clearing condition]

- When 1 is set to the TSTRA.CST1 bit while SCH1 = 1

SCH0 Bit (Synchronous Start 0)

This bit controls synchronous start of MTU0.TCNT.
[Clearing condition]

- When 1 is set to the TSTRA.CST0 bit while SCH0 = 1

22.2.20 Timer Read/Write Enable Register m (TRWERm) (m = A, B)

Address(es): MTU.TRWERA 000C 1284h, MTU.TRWERB 000C 1A84h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	RWE
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	RWE	Read/Write Enable	0: Read/write access to the registers is disabled 1: Read/write access to the registers is enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TRWERA enables or disables access to the registers and counters that have write-protection capability against accidental modification in MTU3 and MTU4.
TRWERB enables or disables access to the registers and counters that have write-protection capability against accidental modification in MTU6 and MTU7.

RWE Bit (Read/Write Enable)

This bit enables or disables access to the registers that have write-protection capability against accidental modification.
[Clearing condition]

- When 0 is written to the RWE bit after reading RWE = 1

- Registers and Counters having Write-Protection Capability against Accidental Modification (TRWERA)

24 registers: MTUn.TCR, MTUn.TCR2, MTUn.TMDR1, MTUn.TIORH, MTUn.TIORL, MTUn.TIER, MTUn.TGRA, MTUn.TGRB, MTU.TOERA, MTU.TOCR1A, MTU.TOCR2A, MTU.TGCRA, MTU.TCDRA, MTU.TDDRA, and MTUn.TCNT (n = 3, 4)

- Registers and Counters having Write-Protection Capability against Accidental Modification (TRWERB)

24 registers: MTUn.TCR, MTUn.TCR2, MTUn.TMDR1, MTUn.TIORH, MTUn.TIORL, MTUn.TIER, MTUn.TGRA, MTUn.TGRB, MTU.TOERB, MTU.TOCR1B, MTU.TOCR2B, MTU.TGCRB, MTU.TCDRB, MTU.TDDRB, and MTUn.TCNT (n = 6, 7)

22.2.21 Timer Output Master Enable Register m (TOERm) (m = A, B)

- MTU.TOERA

Address(es): MTU.TOERA 000C 120Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	OE4D	OE4C	OE3D	OE4B	OE4A	OE3B
Value after reset:	1	1	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OE3B	Master Enable MTIOC3B	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b1	OE4A	Master Enable MTIOC4A	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b2	OE4B	Master Enable MTIOC4B	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b3	OE3D	Master Enable MTIOC3D	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b4	OE4C	Master Enable MTIOC4C	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b5	OE4D	Master Enable MTIOC4D	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b7, b6	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

Note 1. To output the inactive level from each pin when the MTU output is set to disabled, first set the data direction register (PDR) and port output data register (PODR) of I/O ports to output the inactive level from general I/O ports, and then set the port mode register (PMR) to use general I/O ports. For details, refer to section 20, I/O Ports.

TOERA enables or disables output settings for output pins MTIOC4D, MTIOC4C, MTIOC3D, MTIOC4B, MTIOC4A, and MTIOC3B.

These pins do not output correctly if the bits in the TOERA register have not been set. In MTU3 and MTU4, set TOERA prior to setting TIOR.

Set MTU.TOERA after setting the CST3 and CST4 bits in MTU.TSTRA to 0 (refer to Figure 22.43 and Figure 22.47).

- MTU.TOERB

Address(es): MTU.TOERB 000C 1A0Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	OE7D	OE7C	OE6D	OE7B	OE7A	OE6B
Value after reset:	1	1	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OE6B	Master Enable MTIOC6B	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b1	OE7A	Master Enable MTIOC7A	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b2	OE7B	Master Enable MTIOC7B	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b3	OE6D	Master Enable MTIOC6D	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b4	OE7C	Master Enable MTIOC7C	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b5	OE7D	Master Enable MTIOC7D	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b7, b6	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

Note 1. To output the inactive level from each pin when the MTU output is set to disabled, first set the data direction register (PDR) and port output data register (PODR) of I/O ports to output the inactive level from general I/O ports, and then set the port mode register (PMR) to use general I/O ports. For details, refer to section 20, I/O Ports.

TOERB enables or disables output settings for output pins MTIOC7D, MTIOC7C, MTIOC6D, MTIOC7B, MTIOC7A, and MTIOC6B.

These pins do not output correctly if the bits in the TOERB register have not been set. In MTU6, and MTU7, set TOERB prior to setting TIOR.

Set MTU.TOERB after setting the CST6 and CST7 bits in MTU.TSTRB to 0 (refer to Figure 22.43 and Figure 22.47).

22.2.22 Timer Output Control Register 1m (TOCR1m) (m = A, B)

Address(es): MTU.TOCR1A 000C 120Eh, MTU.TOCR1B 000C 1A0Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	PSYE	—	—	TOCL	TOCS	OLSN	OLSP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OLSP	Output Level Select P*1, *3	Refer to Table 22.42.	R/W
b1	OLSN	Output Level Select N*1, *3	Refer to Table 22.43.	R/W
b2	TOCS	TOC Select	0: TOCR1m setting is selected (m = A, B) 1: TOCR2m setting is selected	R/W
b3	TOCL	TOC Register Write Protection*2, *4	0: Write access to the TOCS, OLSN, and OLSP bits is enabled 1: Write access to the TOCS, OLSN, and OLSP bits is disabled	R/W
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	PSYE	PWM Synchronous Output Enable	0: Toggle output is disabled 1: Toggle output is enabled	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. Setting the TOCR1m.TOCS bit to 0 makes this bit setting valid.
Note 2. Setting the TOCR1m.TOCL bit to 1 prevents accidental modification when the CPU goes out of control.
Note 3. If dead-time is not generated, the negative-phase output is the exact inverse of the positive-phase output. In this case, only the OLSP bit is valid.
Note 4. This bit can be set to 1 only once after a reset. After 1 is written, 0 cannot be written to the bit.

TOCR1A and TOCR1B enable or disable PWM-synchronized toggle output in complementary PWM mode and reset-synchronized PWM mode, and control inversion of PWM output level.

OLSP Bit (Output Level Select P)

This bit selects the positive-phase output level in reset-synchronized PWM mode and complementary PWM mode. The initial output is selected while the counter is stopped.

OLSN Bit (Output Level Select N)

This bit selects the negative-phase output level in reset-synchronized PWM mode and complementary PWM mode. The initial output is selected while the counter is stopped.

TOCS Bit (TOC Select)

This bit selects either the TOCR1m or TOCR2m (m = A, B) setting to be used for the output level in complementary PWM mode and reset-synchronized PWM mode.

TOCL Bit (TOC Register Write Protection)

This bit enables or disables write access to the TOCS, OLSN, and OLSP bits in TOCR1m (m = A, B).

PSYE Bit (PWM Synchronous Output Enable)

This bit enables or disables toggle output synchronized with the PWM period from the MTIOC3A or MTIOC6A pin.

Table 22.42 Output Level Select Function

Bit 0	Function			
			Compare Match Output	
OLSP	Initial Output	Active Level	Up-Counting	Down-Counting
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

Table 22.43 Output Level Select Function

Bit 1	Function			
			Compare Match Output	
OLSN	Initial Output	Active Level	Up-Counting	Down-Counting
0	High level	Low level	High level	Low level
1	Low level	High level	Low level	High level

Note: The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

Figure 22.3 shows an example of output in complementary PWM mode (one phase) when OLSN = 1 and OLSP = 1.

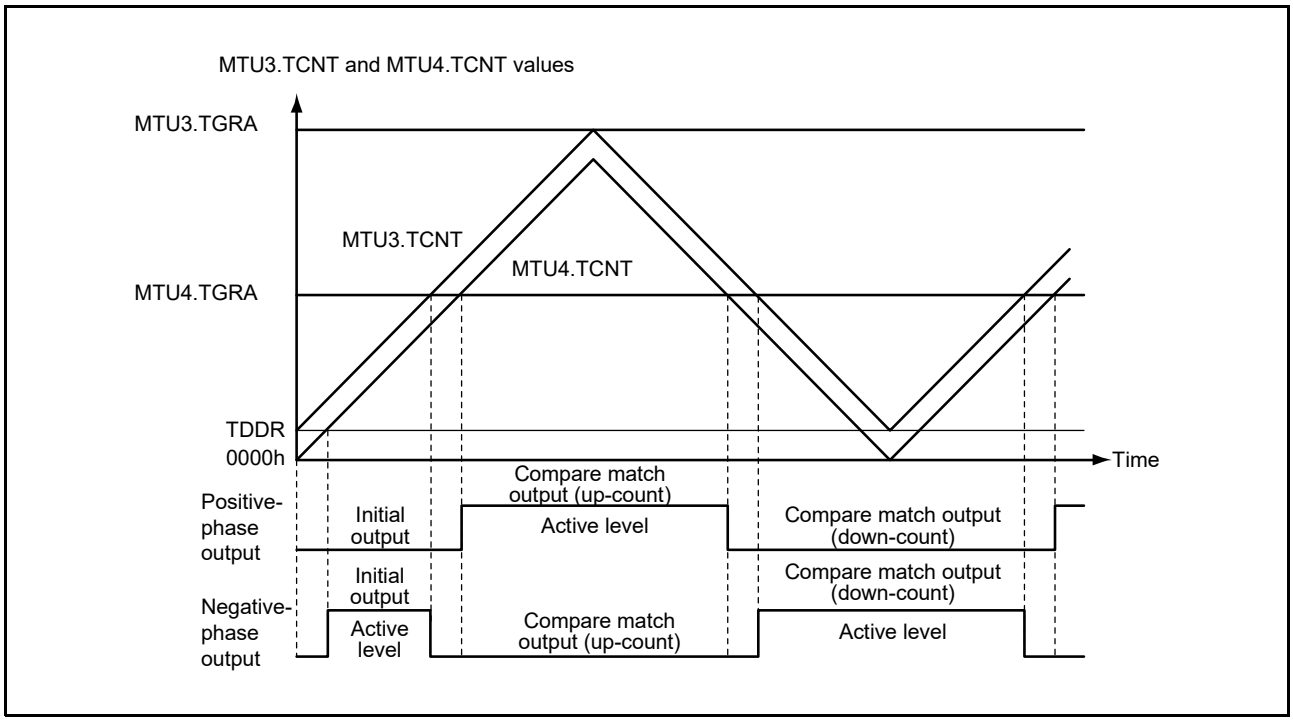


Figure 22.3 Example of Output in Complementary PWM Mode

22.2.23 Timer Output Control Register 2m (TOCR2m) (m = A, B)

Address(es): MTU.TOCR2A 000C 120Fh, MTU.TOCR2B 000C 1A0Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	BF[1:0]		OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OLS1P	Output Level Select 1P*1, *2	This bit selects the output level on MTIOC3B or MTIOC6B in reset-synchronized PWM mode and complementary PWM mode. Refer to Table 22.44.	R/W
b1	OLS1N	Output Level Select 1N*1, *2	This bit selects the output level on MTIOC3D or MTIOC6D in reset-synchronized PWM mode and complementary PWM mode. Refer to Table 22.45.	R/W
b2	OLS2P	Output Level Select 2P*1, *2	This bit selects the output level on MTIOC4A or MTIOC7A in reset-synchronized PWM mode and complementary PWM mode. Refer to Table 22.46.	R/W
b3	OLS2N	Output Level Select 2N*1, *2	This bit selects the output level on MTIOC4C or MTIOC7C in reset-synchronized PWM mode and complementary PWM mode. Refer to Table 22.47.	R/W
b4	OLS3P	Output Level Select 3P*1, *2	This bit selects the output level on MTIOC4B or MTIOC7B in reset-synchronized PWM mode and complementary PWM mode. Refer to Table 22.48.	R/W
b5	OLS3N	Output Level Select 3N*1, *2	This bit selects the output level on MTIOC4D or MTIOC7D in reset-synchronized PWM mode and complementary PWM mode. Refer to Table 22.49.	R/W
b7, b6	BF[1:0]	TOLBR Buffer Transfer Timing Select	These bits select the timing for transferring data from TOLBRm to TOCR2m. Refer to Table 22.50 for details.	R/W

m = A, B

Note 1. Setting the TOCR1m.TOCS bit to 1 makes this bit setting valid.

Note 2. If dead-time is not generated, the negative-phase output is the exact inverse of the positive-phase output. In this case, only the OLSiP bits are valid (i = 1 to 3).

TOCR2A and TOCR2B control inversion of PWM output level in complementary PWM mode and reset-synchronized PWM mode.

The initial output is selected while the counter is stopped.

Table 22.44 MTIOCnB Output Level Select Function

Bit 0	Function			
			Compare Match Output	
OLS1P	Initial Output	Active Level	Up-Counting	Down-Counting
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

n = 3, 6

Table 22.45 MTIOCnD Output Level Select Function

Bit 1	Function			
			Compare Match Output	
OLS1N	Initial Output	Active Level	Up-Counting	Down-Counting
0	High level	Low level	High level	Low level
1	Low level	High level	Low level	High level

n = 3, 6

Note: The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

Table 22.46 MTIOCnA Output Level Select Function

Bit 2	Function			
			Compare Match Output	
OLS2P	Initial Output	Active Level	Up-Counting	Down-Counting
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

n = 4, 7

Table 22.47 MTIOCnC Output Level Select Function

Bit 3	Function			
			Compare Match Output	
OLS2N	Initial Output	Active Level	Up-Counting	Down-Counting
0	High level	Low level	High level	Low level
1	Low level	High level	Low level	High level

n = 4, 7

Note: The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

Table 22.48 MTIOCnB Output Level Select Function

Bit 4	Function			
			Compare Match Output	
OLS3P	Initial Output	Active Level	Up-Counting	Down-Counting
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

n = 4, 7

Table 22.49 MTIOCnD Output Level Select Function

Bit 5	Function			
			Compare Match Output	
OLS3N	Initial Output	Active Level	Up-Counting	Down-Counting
0	High level	Low level	High level	Low level
1	Low level	High level	Low level	High level

n = 4, 7

Note: The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

Table 22.50 Setting of TOCR2m.BF[1:0] Bits

Bit 7	Bit 6	Description	
BF[1]	**BF[0]**	**Complementary PWM Mode**	**Reset-Synchronized PWM Mode**
0	0	Does not transfer data from the buffer register (TOLBRm) to TOCR2m.	Does not transfer data from the buffer register (TOLBRm) to TOCR2m.
0	1	Transfers data from the buffer register (TOLBRm) to TOCR2m at the crest of the MTUn.TCNT count.	Transfers data from the buffer register (TOLBRm) to TOCR2m when MTUk.TCNT or MTUn.TCNT is cleared.
1	0	Transfers data from the buffer register (TOLBRm) to TOCR2m at the trough of the MTUn.TCNT count.	Setting prohibited
1	1	Transfers data from the buffer register (TOLBRm) to TOCR2m at the crest and trough of the MTUn.TCNT count.	Setting prohibited

n = 4, 7; k = 3, 6; m = A, B

22.2.24 Timer Output Level Buffer Register m (TOLBRm) (m = A, B)

Address(es): MTU.TOLBRA 000C 1236h, MTU.TOLBRB 000C 1A36h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OLS1P	Output Level Select 1P	Specify the buffer value to be transferred to the OLS1P bit in TOCR2m.	R/W
b1	OLS1N	Output Level Select 1N	Specify the buffer value to be transferred to the OLS1N bit in TOCR2m.	R/W
b2	OLS2P	Output Level Select 2P	Specify the buffer value to be transferred to the OLS2P bit in TOCR2m.	R/W
b3	OLS2N	Output Level Select 2N	Specify the buffer value to be transferred to the OLS2N bit in TOCR2m.	R/W
b4	OLS3P	Output Level Select 3P	Specify the buffer value to be transferred to the OLS3P bit in TOCR2m.	R/W
b5	OLS3N	Output Level Select 3N	Specify the buffer value to be transferred to the OLS3N bit in TOCR2m.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

m = A, B

TOLBRA and TOLBRB are buffer registers for TOCR2A and TOCR2B and specify the PWM output level in complementary PWM mode and reset-synchronized PWM mode.

Figure 22.4 shows an example of the PWM output level setting procedure in buffer operation.

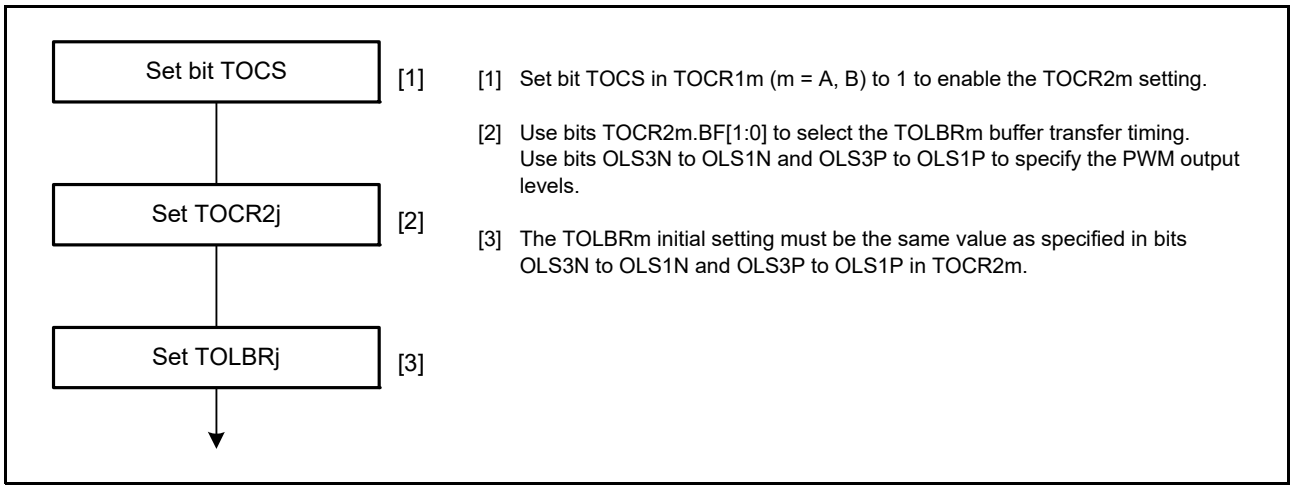


Figure 22.4 Example of PWM Output Level Setting Procedure in Buffer Operation

22.2.25 Timer Gate Control Register m (TGCRm) (m = A, B)

Address(es): MTU.TGCRA 000C 120Dh, MTU.TGCRB 000C 1A0Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	BDC	N	P	FB	WF	VF	UF
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	UF	Output Phase Switch	These bits turn on or off the positive-phase/negative-phase output. The setting of these bits is valid only when the FB bit is set to 1. In this case, the setting of b0 to b2 is used instead of the external input. Refer to Table 22.51.	R/W
b1	VF			R/W
b2	WF			R/W
b3	FB	External Feedback Signal Enable	0: Output is switched by external input (input sources are TGRA, TGRB, and TGRC input capture signals in MTU0 and MTU9) 1: Output is switched by software (TGCRA and TGCRB's UF, VF, and WF settings)	R/W
b4	P	Positive-Phase Output (P) Control	0: Level output 1: Reset-synchronized PWM or complementary PWM output	R/W
b5	N	Negative-Phase Output (N) Control	0: Level output 1: Reset-synchronized PWM or complementary PWM output	R/W
b6	BDC	Brushless DC Motor	0: Ordinary output 1: Functions of this register are made effective	R/W
b7	—	Reserved	This bit is read as 1. The write value should be 1.	R/W

TGCRA and TGCRB control the output waveform necessary for brushless DC motor control in reset-synchronized PWM mode and complementary PWM mode. TGCRA and TGCRB register settings are ineffective for anything other than complementary PWM mode and reset-synchronized PWM mode.

UF, VF, and WF Bits (Output Phase Switch)

The setting of these bits is valid only when the FB bit is set to 1. In this case, the setting of b0 to b2 is used instead of the external input. Refer to Table 22.51 for details.

FB Bit (External Feedback Signal Enable)

This bit selects whether the positive-/negative-phase output is switched automatically with the TGRA, TGRB, and TGRC input capture signals in MTU0 and MTU9 or by writing 0 or 1 to bits 2 to 0 in TGCRA and TGCRB.
When the TGCRA.FB bit is 0, output of MTU3 and MTU4 can be switched with the TGRA, TGRB, and TGRC input capture signals in MTU0.
When the TGCRB.FB bit is 0, output of MTU6 and MTU7 can be switched with the TGRA, TGRB, and TGRC input capture signals in MTU9.

P Bit (Positive-Phase Output (P) Control)

This bit selects the level output or the reset-synchronized PWM/complementary PWM output for the positive-phase output pins (MTIOC3B, MTIOC4A, MTIOC4B, MTIOC6B, MTIOC7A, and MTIOC7B pins).

N Bit (Negative-Phase Output (N) Control)

This bit selects the level output or the reset-synchronized PWM/complementary PWM output for the negative-phase output pins (MTIOC3D, MTIOC4C, MTIOC4D, MTIOC6D, MTIOC7C, and MTIOC7D pins).

BDC Bit (Brushless DC Motor)

This bit selects whether to make the functions of TGCRA and TGCRB effective or ineffective.

Table 22.51 Output Level Select Function

Bit 2	Bit 1	Bit 0	Function					
			MTIOC3B, MTIOC6B	MTIOC4A, MTIOC7A	MTIOC4B, MTIOC7B	MTIOC3D, MTIOC6D	MTIOC4C, MTIOC7C	MTIOC4D, MTIOC7D
WF	VF	UF	U Phase	V Phase	W Phase	U Phase	V Phase	W Phase
0	0	0	OFF	OFF	OFF	OFF	OFF	OFF
0	0	1	ON	OFF	OFF	OFF	OFF	ON
0	1	0	OFF	ON	OFF	ON	OFF	OFF
0	1	1	OFF	ON	OFF	OFF	OFF	ON
1	0	0	OFF	OFF	ON	OFF	ON	OFF
1	0	1	ON	OFF	OFF	OFF	ON	OFF
1	1	0	OFF	OFF	ON	ON	OFF	OFF
1	1	1	OFF	OFF	OFF	OFF	OFF	OFF

22.2.26 Timer Subcounter m (TCNTSm) (m = A, B)

Address(es): MTU.TCNTSA 000C 1220h, MTU.TCNTSB 000C 1A20h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: TCNTSA and TCNTSB must not be accessed in 8 bits; it should be accessed in 16 bits.

TCNTSA and TCNTSB are 16-bit read-only counters used only in complementary PWM mode.
The initial value of TCNTSA and TCNTSB after a reset is 0000h.

22.2.27 Timer Period Data Register m (TCDRm) (m = A, B)

Address(es): MTU.TCDRA 000C 1214h, MTU.TCDRB 000C 1A14h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Note: TCDRA and TCDRB must not be accessed in 8 bits; it should be accessed in 16 bits.

TCDRA and TCDRB are 16-bit readable/writable registers used only in complementary PWM mode. Set half the PWM carrier period as the TCDRA and TCDRB values. The TCDRA and TCDRB registers are constantly compared with the TCNTSA and TCNTSB counters in complementary PWM mode, respectively. When a match occurs, the TCNTSA and TCNTSB counters switch the count direction (down-count to up-count). The initial value of TCDRA and TCDRB after a reset is FFFFh.

22.2.28 Timer Period Buffer Register m (TCBRm) (m = A, B)

Address(es): MTU.TCBRA 000C 1222h, MTU.TCBRB 000C 1A22h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Note: TCBRA and TCBRB must not be accessed in 8 bits; it should be accessed in 16 bits.

TCBRA and TCBRB are 16-bit readable/writable registers, used only in complementary PWM mode, that function as buffer registers for TCDRA and TCDRB. The TCBRA and TCBRB values are transferred to TCDRA and TCDRB with the transfer timing set in TMDR1. The initial value of TCBRA and TCBRB after a reset is FFFFh.

22.2.29 Timer Dead Time Data Register m (TDDRm) (m = A, B)

Address(es): MTU.TDDRA 000C 1216h, MTU.TDDRB 000C 1A16h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Note: TDDRA and TDDRB must not be accessed in 8 bits; it should be accessed in 16 bits.

TDDRA and TDDRB are 16-bit readable/writable registers, used only in complementary PWM mode, that specify the MTU3.TCNT (MTU6.TCNT) and MTU4.TCNT (MTU7.TCNT) counter offset value. In complementary PWM mode, when the MTU3.TCNT (MTU6.TCNT) and MTU4.TCNT (MTU7.TCNT) counters are cleared and then restarted, the TDDRA (TDDRB) value is loaded into the MTU3.TCNT (MTU6.TCNT) counter and the count operation starts. The initial value of TDDRA and TDDRB after a reset is FFFFh.

22.2.30 Timer Dead Time Enable Register m (TDERm) (m = A, B)

Address(es): MTU.TDERA 000C 1234h, MTU.TDERB 000C 1A34h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	TDER
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	TDER	Dead Time Enable	0: No dead time is generated 1: Dead time is generated*1	R/(W)
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. TDDRA and TDDRB must be set to 1 or a larger value.

TDERA and TDERB control dead time generation in complementary PWM mode. The MTU has one TDER each for MTU3 and MTU6. TDERA and TDERB should be modified only while TCNT stops.

TDER Bit (Dead Time Enable)

This bit specifies whether to generate dead time.

[Clearing condition]

- When 0 is written to TDER after reading TDER = 1

22.2.31 Timer Buffer Transfer Set Register m (TBTERm) (m = A, B)

Address(es): MTU.TBTERA 000C 1232h, MTU.TBTERB 000C 1A32h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	BTE[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	BTE[1:0]	Buffer Transfer Disable and Interrupt Skipping Link Setting	These bits enable or disable transfer from the buffer registers*1 used in complementary PWM mode to the temporary registers, and specify whether to link the transfer with interrupt skipping function 1. For details, refer to Table 22.52.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Applicable buffer registers (TBTERA):
MTU3.TGRC, MTU3.TGRD, MTU4.TGRC, MTU4.TGRD, and TCBRA
Applicable buffer registers (TBTERB):
MTU6.TGRC, MTU6.TGRD, MTU7.TGRC, MTU7.TGRD, and TCBRB

TBTERA and TBTERB enable or disable transfer from the buffer registers used in complementary PWM mode to the temporary registers, and specify whether to link the transfer with interrupt skipping 1 operation.

Table 22.52 Setting of TBTERA.BTE[1:0] Bits and TBTERB.BTE[1:0] Bits

Bit 1	Bit 0	
BTE[1]	BTE[0]	Description
0	0	Enables transfer from the buffer registers to the temporary registers*1 and does not link the transfer with interrupt skipping function 1.
0	1	Disables transfer from the buffer registers to the temporary registers.
1	0	Links transfer from the buffer registers to the temporary registers with interrupt skipping function 1.*2
1	1	Setting prohibited

Note 1. Data is transferred according to the MD3 to MD0 bit setting in TMDR1. For details, refer to section 22.3.8, Complementary PWM Mode.

Note 2. When interrupt skipping is disabled the T3AEN and T4VEN (T6AEN and T7VEN) bits are set to 0 in the timer interrupt skipping set register (TITCR1A (TITCR1B)) or the skipping count set bits (T3ACOR and T4VCOR (T6ACOR and T7VCOR)) in TITCR1A (TITCR1B) are set to 0), be sure to disable link of buffer transfer with interrupt skipping (set the BTE1 bit in the timer buffer transfer set register (TBTERA (TBTERB)) to 0). If link with interrupt skipping is enabled while interrupt skipping is disabled, buffer transfer will not be performed.

22.2.32 Timer Waveform Control Register m (TWCRm) (m = A, B)

Address(es): MTU.TWCRA 000C 1260h, MTU.TWCRB 000C 1A60h

	b7	b6	b5	b4	b3	b2	b1	b0
	CCE	—	—	—	—	—	SCC	WRE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	WRE	Waveform Retain Enable	0: Initial values specified in TOCR1A and TOCR2A (TOCR1B and TOCR2B) are output 1: Initial output is inhibited	R/(W) *3
b1	SCC	Synchronous Clearing Control *1, *3	(Only valid in TWCRB) 0: Clearing of MTU6.TCNT and MTU7.TCNT in response to synchronous clearing for MTU0, MTU1, MTU2–MTU6, MTU7 is enabled. 1: Clearing of MTU6.TCNT and MTU7.TCNT in response to synchronous clearing for MTU0, MTU1, MTU2–MTU6, MTU7 is disabled.	R/(W)
b6 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	CCE	Compare Match Clear Enable *2	0: Counters are not cleared at MTU3.TGRA (MTU6.TGRA) compare match 1: Counters are cleared at MTU3.TGRA (MTU6.TGRA) compare match	R/(W)

Note 1. This bit is only valid in register TWCRB and is a reserved bit in register TWCRA.
Note 2. Do not set to 1 when complementary PWM mode 1 is not selected.
Note 3. Do not set to 1 when complementary PWM mode is not selected.

TWCRA and TWCRB control the output waveform when synchronous counter clearing occurs in MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT) in complementary PWM mode and specifies whether to clear the counters at MTU3.TGRA (MTU6.TGRA) compare match.
The CCE bit and WRE bit in TWCRA and TWCRB should be modified only while TCNT stops.

WRE Bit (Waveform Retain Enable)

This bit selects the waveform output when synchronous counter clearing occurs in complementary PWM mode.
The initial output is inhibited with this function only when synchronous clearing occurs within the Tb interval at the trough in complementary PWM mode. When synchronous clearing occurs outside this interval, the initial values specified in TOCR1A and TOCR2A (TOCR1B and TOCR2B) are output regardless of the WRE bit setting. The initial values specified in TOCR1A and TOCR2A (TOCR1B and TOCR2B) are also output when synchronous clearing occurs in the Tb interval at the trough immediately after MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT) start operation.
For the Tb interval at the trough in complementary PWM mode, refer to Figure 22.49.
[Setting condition]

- When 1 is written to the WRE bit after reading WRE = 0

SCC Bit (Synchronous Clearing Control)

The setting of this bit selects whether MTU6.TCNT and MTU7.TCNT are or are not cleared when counter-synchronous clearing is generated for MTU0, MTU1, MTU2–MTU6, MTU7 in complementary PWM mode.
Make the complementary PWM mode settings for MTU6 and MTU7 when this function is in use. When writing a new value to the SCC bit while the counter is operating, do so in such a way that the values of the CCE and WRE bits are not changed.
Synchronous clearing from the MTU module only becomes disabled due to the setting of the SCC bit when synchronous clearing is generated outside the Tb interval in the trough. If synchronous clearing is generated within the Tb interval in the trough including immediately after the value at which MTU6.TCNT and MTU7.TCNT start, MTU6.TCNT and MTU7.TCNT are cleared.
Regarding the Tb interval in the trough in complementary PWM mode, refer to Figure 22.49.
[Setting condition]

- Writing of 1 to the SCC bit after reading it as 0

The corresponding bit in register TWCRA is reserved and is read as 0. When writing to TWCRA, write 0 to this bit.

CCE Bit (Compare Match Clear Enable)

This bit specifies whether to clear counters at MTU3.TGRA (MTU6.TGRA) compare match in complementary PWM mode.
[Setting condition]

- When 1 is written to CCE after reading CCE = 0

22.2.33 Noise Filter Control Register n (NFCRn) (n = 0 to 4, 6, 7, 9, C)

- MTU0.NFCR0, MTU1.NFCR1, MTU2.NFCR2, MTU3.NFCR3, MTU4.NFCR4, MTU6.NFCR6, MTU7.NFCR7, MTU9.NFCR9

Address(es): MTU0.NFCR0 000C 1290h, MTU1.NFCR1 000C 1291h, MTU2.NFCR2 000C 1292h, MTU3.NFCR3 000C 1293h, MTU4.NFCR4 000C 1294h, MTU6.NFCR6 000C 1A93h, MTU7.NFCR7 000C 1A94h, MTU9.NFCR9 000C 1296h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	NFCS[1:0]		NFDEN	NFCEN	NFBEN	NFAEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	NFAEN	Noise Filter A Enable	0: The noise filter for the MTIOCnA pin is disabled. 1: The noise filter for the MTIOCnA pin is enabled.	R/W
b1	NFBEN	Noise Filter B Enable	0: The noise filter for the MTIOCnB pin is disabled. 1: The noise filter for the MTIOCnB pin is enabled.	R/W
b2	NFCEN	Noise Filter C Enable*1	0: The noise filter for the MTIOCnC pin is disabled. 1: The noise filter for the MTIOCnC pin is enabled.	R/W
b3	NFDEN	Noise Filter D Enable*1	0: The noise filter for the MTIOCnD pin is disabled. 1: The noise filter for the MTIOCnD pin is enabled.	R/W
b5, b4	NFCS[1:0]	Noise Filter Clock Select	b5 b4 0 0: PCLKC/1 0 1: PCLKC/8 1 0: PCLKC/32 1 1: Clock source for counting	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. These bits are reserved in MTU1 and MTU2. These bits are read as 0 and writing to them has no effect.

The NFCRn register (n = 0 to 4, 6, 7, 9) sets the noise filter function of input capture pins for the corresponding channel.

NFAEN Bit (Noise Filter A Enable)

This bit disables or enables the noise filter for input from the MTIOCnA pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, select the output compare function for the relevant pin in the timer I/O control register or set the TMDR.MD[3:0] bits to a value other than that for normal mode (0000b) before doing so.

NFBEN Bit (Noise Filter B Enable)

This bit disables or enables the noise filter for input from the MTIOCnB pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, select the output compare function for the relevant pin in the timer I/O control register or set the TMDR.MD[3:0] bits to a value other than that for normal mode (0000b) before doing so.

NFCEN Bit (Noise Filter C Enable)

This bit disables or enables the noise filter for input from the MTIOCnC pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, select the output compare function for the relevant pin in the timer I/O control register or set the TMDR.MD[3:0] bits to a value other than that for normal mode (0000b) before doing so.

NFDEN Bit (Noise Filter D Enable)

This bit disables or enables the noise filter for input from the MTIOCnD pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, select the output compare function for the relevant pin in the timer I/O control register or set the TMDR.MD[3:0] bits to a value other than that for normal mode (0000b) before doing so.

NFCS[1:0] Bits (Noise Filter Clock Select)

These bits set the sampling interval for the noise filters. When setting the NFCS[1:0] bits, wait for two cycles of the selected sampling interval before setting the input-capture function. When the NFCS[1:0] bits are set to 11b, i.e. selecting the external clock as the source to drive counting, wait for two cycles of the external clock before setting the input capture function.

- MTU0.NFCRC

Address(es): MTU0.NFCRC 000C 1299h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	NFCS[1:0]		NFDEN	NFCEN	NFBEN	NFAEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	NFAEN	Noise Filter A Enable	0: The noise filter for the MTCLKA pin is disabled. 1: The noise filter for the MTCLKA pin is enabled.	R/W
b1	NFBEN	Noise Filter B Enable	0: The noise filter for the MTCLKB pin is disabled. 1: The noise filter for the MTCLKB pin is enabled.	R/W
b2	NFCEN	Noise Filter C Enable	0: The noise filter for the MTCLKC pin is disabled. 1: The noise filter for the MTCLKC pin is enabled.	R/W
b3	NFDEN	Noise Filter D Enable	0: The noise filter for the MTCLKD pin is disabled. 1: The noise filter for the MTCLKD pin is enabled.	R/W
b5, b4	NFCS[1:0]	Noise Filter Clock Select	b5 b4 0 0: PCLKC/1 0 1: PCLKC/2 1 0: PCLKC/8 1 1: PCLKC/32	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The NFCRC register sets the noise filter function of external clock pins common to each channel.

NFAEN Bit (Noise Filter A Enable)

This bit disables or enables the noise filter for input from the MTCLKA pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, do so after stopping the internal counter.

NFBEN Bit (Noise Filter B Enable)

This bit disables or enables the noise filter for input from the MTCLKB pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, do so after stopping the internal counter.

NFCEN Bit (Noise Filter C Enable)

This bit disables or enables the noise filter for input from the MTCLKC pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, do so after stopping the internal counter.

NFDEN Bit (Noise Filter D Enable)

This bit disables or enables the noise filter for input from the MTCLKD pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, do so after stopping the internal counter.

NFCS[1:0] Bits (Noise Filter Clock Select)

These bits set the sampling interval for the noise filters. After setting the NFCS[1:0] bits, wait for two cycles of the selected sampling interval to set the input capture function.

22.2.34 Noise Filter Control Register 5 (NFCR5)

Address(es): MTU5.NFCR5 000C 1A95h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	NFCS[1:0]		—	NFWEN	NFVEN	NFUEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	NFUEN	Noise Filter U Enable	0: The noise filter for the MTIC5U pin is disabled. 1: The noise filter for the MTIC5U pin is enabled.	R/W
b1	NFVEN	Noise Filter V Enable	0: The noise filter for the MTIC5V pin is disabled. 1: The noise filter for the MTIC5V pin is enabled.	R/W
b2	NFWEN	Noise Filter W Enable	0: The noise filter for the MTIC5W pin is disabled. 1: The noise filter for the MTIC5W pin is enabled.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5, b4	NFCS[1:0]	Noise Filter Clock Select	b5 b4 0 0: PCLKC/1 0 1: PCLKC/8 1 0: PCLKC/32 1 1: Clock source for counting	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

NFUEN Bit (Noise Filter U Enable)

This bit disables or enables the noise filter for input from the MTIC5U pin. Since unexpected edges may be internally generated when the value of this bit is changed, select the output compare function for the relevant pin in the timer I/O control register.

NFVEN Bit (Noise Filter V Enable)

This bit disables or enables the noise filter for input from the MTIC5V pin. Since unexpected edges may be internally generated when the value of this bit is changed, select the output compare function for the relevant pin in the timer I/O control register.

NFWEN Bit (Noise Filter W Enable)

This bit disables or enables the noise filter for input from the MTIC5W pin. Since unexpected edges may be internally generated when the value of this bit is changed, select the output compare function for the relevant pin in the timer I/O control register.

NFCS[1:0] Bits (Noise Filter Clock Select)

These bits set the sampling interval for the noise filters. When setting the NFCS[1:0] bits, wait for two cycles of the selected sampling interval before setting the input-capture function.

22.2.35 Timer A/D Conversion Start Request Control Register (TADCR)

- MTU4.TADCR

Address(es): MTU4.TADCR 000C 1240h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BF[1:0]		—	—	—	—	—	—	UT4AE	DT4AE	UT4BE	DT4BE	ITA3AE	ITA4VE	ITB3AE	ITB4VE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ITB4VE	TCIV4 Interrupt Skipping Link Enable*1, *2, *3	0: A/D conversion start request TRG4BN and TCIV4 interrupt skipping 1 are not linked 1: A/D conversion start request TRG4BN and TCIV4 interrupt skipping 1 are linked	R/W
b1	ITB3AE	TGIA3 Interrupt Skipping Link Enable*1, *2, *3	0: A/D conversion start request TRG4BN and TGIA3 interrupt skipping 1 are not linked 1: A/D conversion start request TRG4BN and TGIA3 interrupt skipping 1 are linked	R/W
b2	ITA4VE	TCIV4 Interrupt Skipping Link Enable*1, *2, *3	0: A/D conversion start request TRG4AN and TCIV4 interrupt skipping 1 are not linked 1: A/D conversion start request TRG4AN and TCIV4 interrupt skipping 1 are linked	R/W
b3	ITA3AE	TGIA3 Interrupt Skipping Link Enable*1, *2, *3	0: A/D conversion start request TRG4AN and TGIA3 interrupt skipping 1 are not linked 1: A/D conversion start request TRG4AN and TGIA3 interrupt skipping 1 are linked	R/W
b4	DT4BE	Down-Count TRG4BN Enable*3	0: A/D conversion start requests (TRG4BN) disabled during MTU4.TCNT down-count operation 1: A/D conversion start requests (TRG4BN) enabled during MTU4.TCNT down-count operation	R/W
b5	UT4BE	Up-Count TRG4BN Enable	0: A/D conversion start requests (TRG4BN) disabled during MTU4.TCNT up-count operation 1: A/D conversion start requests (TRG4BN) enabled during MTU4.TCNT up-count operation	R/W
b6	DT4AE	Down-Count TRG4AN Enable*3	0: A/D conversion start requests (TRG4AN) disabled during MTU4.TCNT down-count operation 1: A/D conversion start requests (TRG4AN) enabled during MTU4.TCNT down-count operation	R/W
b7	UT4AE	Up-Count TRG4AN Enable	0: A/D conversion start requests (TRG4AN) disabled during MTU4.TCNT up-count operation 1: A/D conversion start requests (TRG4AN) enabled during MTU4.TCNT up-count operation	R/W
b13 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15, b14	BF[1:0]	MTU4.TADCOBRA/TADCOBRB Transfer Timing Select	Refer to Table 22.53 for details. These bits specify the transfer timing from MTU4.TADCOBRA and MTU4.TADCOBRB to MTU4.TADCORA and MTU4.TADCORB.	R/W

Note: MTU4.TADCR must not be accessed in 8 bits; it should be accessed in 16 bits.
Note 1. Set to 0 when interrupt skipping is disabled (the T3AEN and T4VEN bits in TITCR1A are set to 0 or the T3ACOR and T4VCOR bits in TITCR1A are set to 0).
Note 2. If link with interrupt skipping is enabled while interrupt skipping is disabled, A/D conversion start requests will not be issued.
Note 3. Set to 0 when complementary PWM mode is not selected.

TADCR enables or disables A/D conversion start requests and specifies whether to link A/D conversion start requests with interrupt skipping function. The MTU has one TADCR each for MTU4 and MTU7.

Table 22.53 Setting of Transfer Timing by TADCR.BF[1:0] Bits (MTU4)

Bit 15	Bit 14	Description			
BF[1]	**BF[0]**	**In Complementary PWM Mode**	**In Reset-Synchronized PWM Mode**	**In PWM Mode 1**	**In Normal Mode**
0	0	Data is not transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB).	Data is not transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB).	Data is not transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB).	Data is not transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB).
0	1	Data is transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB) at the crest of the MTU4.TCNT.	Data is transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB) when a compare match occurs between MTU3.TCNT and MTU3.TGRA.	Data is transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB) when a compare match occurs between MTU4.TCNT and MTU4.TGRA.	Data is transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB) when a compare match occurs between MTU4.TCNT and MTU4.TGRA.
1	0	Data is transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB) at the trough of the MTU4.TCNT.	Setting prohibited	Setting prohibited	Setting prohibited
1	1	Data is transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB) at the crest and trough of the MTU4.TCNT.	Setting prohibited	Setting prohibited	Setting prohibited

- MTU7.TADCR

Address(es): MTU7.TADCR 000C 1A40h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BF[1:0]		—	—	—	—	—	—	UT7AE	DT7AE	UT7BE	DT7BE	ITA6AE	ITA7VE	ITB6AE	ITB7VE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ITB7VE	TCIV7 Interrupt Skipping Link Enable*1, *2, *3	0: A/D conversion start request TRG7BN and TCIV7 interrupt skipping 1 are not linked 1: A/D conversion start request TRG7BN and TCIV7 interrupt skipping 1 are linked	R/W
b1	ITB6AE	TGIA6 Interrupt Skipping Link Enable*1, *2, *3	0: A/D conversion start request TRG7BN and TGIA6 interrupt skipping 1 are not linked 1: A/D conversion start request TRG7BN and TGIA6 interrupt skipping 1 are linked	R/W
b2	ITA7VE	TCIV7 Interrupt Skipping Link Enable*1, *2, *3	0: A/D conversion start request TRG7AN and TCIV7 interrupt skipping 1 are not linked 1: A/D conversion start request TRG7AN and TCIV7 interrupt skipping 1 are linked	R/W
b3	ITA6AE	TGIA6 Interrupt Skipping Link Enable*1, *2, *3	0: A/D conversion start request TRG7AN and TGIA6 interrupt skipping 1 are not linked 1: A/D conversion start request TRG7AN and TGIA6 interrupt skipping 1 are linked	R/W
b4	DT7BE	Down-Count TRG7BN Enable*3	0: A/D conversion start requests (TRG7BN) disabled during MTU7.TCNT down-count operation 1: A/D conversion start requests (TRG7BN) enabled during MTU7.TCNT down-count operation	R/W
b5	UT7BE	Up-Count TRG7BN Enable	0: A/D conversion start requests (TRG7BN) disabled during MTU7.TCNT up-count operation 1: A/D conversion start requests (TRG7BN) enabled during MTU7.TCNT up-count operation	R/W
b6	DT7AE	Down-Count TRG7AN Enable*3	0: A/D conversion start requests (TRG7AN) disabled during MTU7.TCNT down-count operation 1: A/D conversion start requests (TRG7AN) enabled during MTU7.TCNT down-count operation	R/W
b7	UT7AE	Up-Count TRG7AN Enable	0: A/D conversion start requests (TRG7AN) disabled during MTU7.TCNT up-count operation 1: A/D conversion start requests (TRG7AN) enabled during MTU7.TCNT up-count operation	R/W
b13 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15, b14	BF[1:0]	MTU7.TADCOBRA/TADCOBRB Transfer Timing Select	Refer to Table 22.54 for details. These bits specify the transfer timing from MTU7.TADCOBRA and MTU7.TADCOBRB to MTU7.TADCORA and MTU7.TADCORB.	R/W

Note: MTU7.TADCR must not be accessed in 8 bits; it should be accessed in 16 bits.
Note 1. Set to 0 when interrupt skipping is disabled (the T6AEN and T7VEN bits in TITCR1B are set to 0 or the T6ACOR and T7VCOR bits in TITCR1B are set to 0).
Note 2. If link with interrupt skipping is enabled while interrupt skipping is disabled, A/D conversion start requests will not be issued.
Note 3. Set to 0 when complementary PWM mode is not selected.

Table 22.54 Setting of Transfer Timing by TADCR.BF[1:0] Bits (MTU7)

Bit 15	Bit 14	Description			
BF[1]	BF[0]	In Complementary PWM Mode	In Reset-Synchronized PWM Mode	In PWM Mode 1	In Normal Mode
0	0	Data is not transferred from the cycle set buffer register (MTU7.TADCOBRA, MTU7.TADCOBRB) to the cycle set register (MTU7.TADCORA, MTU7.TADCORB).	Data is not transferred from the cycle set buffer register (MTU7.TADCOBRA, MTU7.TADCOBRB) to the cycle set register (MTU7.TADCORA, MTU7.TADCORB).	Data is not transferred from the cycle set buffer register (MTU7.TADCOBRA, MTU7.TADCOBRB) to the cycle set register (MTU7.TADCORA, MTU7.TADCORB).	Data is not transferred from the cycle set buffer register (MTU7.TADCOBRA, MTU7.TADCOBRB) to the cycle set register (MTU7.TADCORA, MTU7.TADCORB).
0	1	Data is transferred from the cycle set buffer register (MTU7.TADCOBRA, MTU7.TADCOBRB) to the cycle set register (MTU7.TADCORA, MTU7.TADCORB) at the crest of the MTU7.TCNT.	Data is transferred from the cycle set buffer register (MTU7.TADCOBRA, MTU7.TADCOBRB) to the cycle set register (MTU7.TADCORA, MTU7.TADCORB) when a compare match occurs between MTU6.TCNT and MTU6.TGRA.	Data is transferred from the cycle set buffer register (MTU7.TADCOBRA, MTU7.TADCOBRB) to the cycle set register (MTU7.TADCORA, MTU7.TADCORB) when a compare match occurs between MTU7.TCNT and MTU7.TGRA.	Data is transferred from the cycle set buffer register (MTU7.TADCOBRA, MTU7.TADCOBRB) to the cycle set register (MTU7.TADCORA, MTU7.TADCORB) when a compare match occurs between MTU7.TCNT and MTU7.TGRA.
1	0	Data is transferred from the cycle set buffer register (MTU7.TADCOBRA, MTU7.TADCOBRB) to the cycle set register (MTU7.TADCORA, MTU7.TADCORB) at the trough of the MTU7.TCNT.	Setting prohibited	Setting prohibited	Setting prohibited
1	1	Data is transferred from the cycle set buffer register (MTU7.TADCOBRA, MTU7.TADCOBRB) to the cycle set register (MTU7.TADCORA, MTU7.TADCORB) at the crest and trough of the MTU7.TCNT.	Setting prohibited	Setting prohibited	Setting prohibited

22.2.36 Timer A/D Conversion Start Request Cycle Set Register m (TADCORm) (m = A, B)

Address(es): MTU4.TADCORA 000C 1244h, MTU4.TADCORB 000C 1246h, MTU7.TADCORA 000C 1A44h, MTU7.TADCORB 000C 1A46h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Note: TADCORA and TADCORB must not be accessed in 8 bits; it should be accessed in 16 bits.

Note 1. When the A/D conversion start request delaying function linked with skipping function 1 (for details, refer to section 22.3.9 (5), A/D Conversion Start Request Delaying Function Linked with Interrupt Skipping Function 1) is used, the value of this register should be 0002h to TCDRA setting – 2 in MTU4 and 0002h to TCDRB setting – 2 in MTU7.

Note 2. When interrupt skipping function 2 is used and the difference between the TADCORA value and the TADCORB value is small, the skipping count may not be counted correctly and the A/D conversion start request may not be generated with the expected timing in some cases. The TADCORA and TADCORB values should satisfy the following conditions.

(1) When skipping function 2 is specified with the skipping count set to 0
- The difference between the TADCORA and TADCORB values should be equal to or greater than 4.
- The TADCORA compare interval should be equal to or greater than 4 PCLKC cycles (the TADCORA update value should be the previous value + 4 or greater, or previous value – 4 or smaller).
- The TADCORB compare interval should be equal to or greater than 4 PCLKC cycles (the TADCORB update value should be the previous value + 4 or greater, or previous value – 4 or smaller).

(2) When skipping function 2 is specified with the skipping count set to 1 or greater
- The difference between the TADCORA and TADCORB values should be equal to or greater than 2.
- The TADCORB compare interval should be equal to or greater than 2 PCLKC cycles (the TADCORB update value should be the previous value + 2 or greater, or previous value – 2 or smaller)

TADCORA and TADCORB are 16-bit readable/writable registers that issue a corresponding A/D conversion start request when the MTUn.TCNT (n = 4, 7) count reaches the value in TADCORA or TADCORB.
MTUn.TADCORA and TADCORB are initialized to FFFFh by a reset.

22.2.37 Timer A/D Conversion Start Request Cycle Set Buffer Register m (TADCOBRm) (m = A, B)

Address(es): MTU4.TADCOBRA 000C 1248h, MTU4.TADCOBRB 000C 124Ah, MTU7.TADCOBRA 000C 1A48h, MTU7.TADCOBRB 000C 1A4Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Note: TADCOBRA and TADCOBRB must not be accessed in 8 bits; it should be accessed in 16 bits.

TADCOBRA and TADCOBRB are 16-bit readable/writable registers whose values are transferred to TADCORA and TADCORB, respectively, when the crest or trough of the MTUn.TCNT count is reached.
TADCOBRA and TADCOBRB are initialized to FFFFh by a reset.

22.2.38 Timer Interrupt Skipping Mode Register m (TITMRm) (m = A, B)

Address(es): MTU.TITMRA 000C 123Ah, MTU.TITMRB 000C 1A3Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	TITM
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TITM	Interrupt Skipping Function Select	Selects one of the two types of interrupt skipping functions. 0: Selects interrupt skipping function 1*1 1: Selects interrupt skipping function 2*2	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Setting the TITCR1A or TITCR1B register enables interrupt skipping function 1.
Note 2. Setting the TITCR2A or TITCR2B register enables interrupt skipping function 2.

TITMRA and TITMRB are used to select either of two skipping functions for the TITMRA and TITMRB registers.

22.2.39 Timer Interrupt Skipping Set Register 1m (TITCR1m) (m = A, B)

- MTU.TITCR1A

Address(es): MTU.TITCR1A 000C 1230h

	b7	b6	b5	b4	b3	b2	b1	b0
	T3AEN	T3ACOR[2:0]			T4VEN	T4VCOR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	T4VCOR[2:0]	TCIV4 Interrupt Skipping Count Setting	These bits specify the TCIV4 interrupt skipping count within the range from 0 to 7.*1 For details, refer to Table 22.55.	R/W
b3	T4VEN	T4VEN	0: TCIV4 interrupt skipping disabled 1: TCIV4 interrupt skipping enabled	R/W
b6 to b4	T3ACOR[2:0]	TGIA3 Interrupt Skipping Count Setting	These bits specify the TGIA3 interrupt skipping count within the range from 0 to 7.*1 For details, refer to Table 22.56.	R/W
b7	T3AEN	T3AEN	0: TGIA3 interrupt skipping disabled 1: TGIA3 interrupt skipping enabled	R/W

Note 1. When 0 is specified for the interrupt skipping count, no interrupt skipping will be performed.
Before changing the interrupt skipping count, be sure to set the TITCR1A.T3AEN and TITCR1A.T4VEN bits to 0 to clear the skipping counter (TITCNT1A).

- MTU.TITCR1B

Address(es): MTU.TITCR1B 000C 1A30h

	b7	b6	b5	b4	b3	b2	b1	b0
	T6AEN	T6ACOR[2:0]			T7VEN	T7VCOR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	T7VCOR[2:0]	TCIV7 Interrupt Skipping Count Setting	These bits specify the TCIV7 interrupt skipping count within the range from 0 to 7.*1 For details, refer to Table 22.57.	R/W
b3	T7VEN	T7VEN	0: TCIV7 interrupt skipping disabled 1: TCIV7 interrupt skipping enabled	R/W
b6 to b4	T6ACOR[2:0]	TGIA6 Interrupt Skipping Count Setting	These bits specify the TGIA6 interrupt skipping count within the range from 0 to 7.*1 For details, refer to Table 22.58.	R/W
b7	T6AEN	T6AEN	0: TGIA6 interrupt skipping disabled 1: TGIA6 interrupt skipping enabled	R/W

Note 1. When 0 is specified for the interrupt skipping count, no interrupt skipping will be performed.
Before changing the interrupt skipping count, be sure to set the TITCR1B.T6AEN and TITCR1B.T7VEN bits to 0 to clear the skipping counter (TITCNT1B).

Registers TITCR1A and TITCR1B enable or disable interrupt skipping and specify the interrupt skipping count. This setting is valid only while the TITMRA.TITM or TITMRB.TITM bit is set to 0; when the TITMRA.TITM (TITMRB.TITM) bit is set to 1, the setting in the TITCR1A (TITCR1B) register is cleared.

Table 22.55 Setting of Interrupt Skipping Count by T4VCOR[2:0] Bits

Bit 2	Bit 1	Bit 0	
T4VCOR[2]	T4VCOR[1]	T4VCOR[0]	Description
0	0	0	Does not skip TCIV4 interrupts.
0	0	1	Sets the TCIV4 interrupt skipping count to 1.
0	1	0	Sets the TCIV4 interrupt skipping count to 2.
0	1	1	Sets the TCIV4 interrupt skipping count to 3.
1	0	0	Sets the TCIV4 interrupt skipping count to 4.
1	0	1	Sets the TCIV4 interrupt skipping count to 5.
1	1	0	Sets the TCIV4 interrupt skipping count to 6.
1	1	1	Sets the TCIV4 interrupt skipping count to 7.

Table 22.56 Setting of Interrupt Skipping Count by T3ACOR[2:0] Bits

Bit 6	Bit 5	Bit 4	
T3ACOR[2]	T3ACOR[1]	T3ACOR[0]	Description
0	0	0	Does not skip TGIA3 interrupts.
0	0	1	Sets the TGIA3 interrupt skipping count to 1.
0	1	0	Sets the TGIA3 interrupt skipping count to 2.
0	1	1	Sets the TGIA3 interrupt skipping count to 3.
1	0	0	Sets the TGIA3 interrupt skipping count to 4.
1	0	1	Sets the TGIA3 interrupt skipping count to 5.
1	1	0	Sets the TGIA3 interrupt skipping count to 6.
1	1	1	Sets the TGIA3 interrupt skipping count to 7.

Table 22.57 Setting of Interrupt Skipping Count by T7VCOR[2:0] Bits

Bit 2	Bit 1	Bit 0	
T7VCOR[2]	T7VCOR[1]	T7VCOR[0]	Description
0	0	0	Does not skip TCIV7 interrupts.
0	0	1	Sets the TCIV7 interrupt skipping count to 1.
0	1	0	Sets the TCIV7 interrupt skipping count to 2.
0	1	1	Sets the TCIV7 interrupt skipping count to 3.
1	0	0	Sets the TCIV7 interrupt skipping count to 4.
1	0	1	Sets the TCIV7 interrupt skipping count to 5.
1	1	0	Sets the TCIV7 interrupt skipping count to 6.
1	1	1	Sets the TCIV7 interrupt skipping count to 7.

Table 22.58 Setting of Interrupt Skipping Count by T6ACOR[2:0] Bits

Bit 6	Bit 5	Bit 4	
T6ACOR[2]	T6ACOR[1]	T6ACOR[0]	Description
0	0	0	Does not skip TGIA6 interrupts.
0	0	1	Sets the TGIA6 interrupt skipping count to 1.
0	1	0	Sets the TGIA6 interrupt skipping count to 2.
0	1	1	Sets the TGIA6 interrupt skipping count to 3.
1	0	0	Sets the TGIA6 interrupt skipping count to 4.
1	0	1	Sets the TGIA6 interrupt skipping count to 5.
1	1	0	Sets the TGIA6 interrupt skipping count to 6.
1	1	1	Sets the TGIA6 interrupt skipping count to 7.

22.2.40 Timer Interrupt Skipping Counter 1m (TITCNT1m) (m = A, B)

- MTU.TITCNT1A

Address(es): MTU.TITCNT1A 000C 1231h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	T3ACNT[2:0]			—	T4VCNT[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	T4VCNT[2:0]	TCIV4 Interrupt Counter	While the T4VEN bit in TITCR1A is set to 1, the count in these bits is incremented every time a TCIV4 interrupt occurs.	R
b3	—	Reserved	This bit is read as 0.	R
b6 to b4	T3ACNT[2:0]	TGIA3 Interrupt Counter	While the T3AEN bit in TITCR1A is set to 1, the count in these bits is incremented every time a TGIA3 interrupt occurs.	R
b7	—	Reserved	This bit is read as 0.	R

Note: To clear the TITCNT1A, set the TITCR1A.T3AEN and TITCR1A.T4VEN bits to 0.

TITCNT1A and TITCNT1B are 8-bit readable/writable counters. TITCNT1A and TITCNT1B retain their values even after stopping the count operation of MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT).

T4VCNT[2:0] Bits (TCIV4 Interrupt Counter)

[Clearing conditions]

- When the TITM bit in TITMRA is 1
- When the T4VEN bit in TITCR1A is set to 0
- When the T4VCOR[2:0] bits in TITCR1A are set to 000b
- When the T4VCNT[2:0] bits in TITCNT1A match the T4VCOR[2:0] bits in TITCR1A

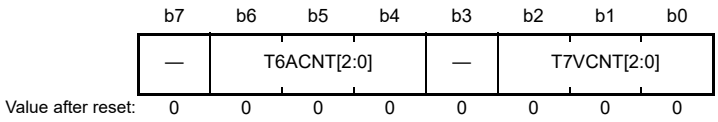
T3ACNT[2:0] Bits (TGIA3 Interrupt Counter)

[Clearing conditions]

- When the TITM bit in TITMRA is 1
- When the T3AEN bit in TITCR1A is set to 0
- When the T3ACOR[2:0] bits in TITCR1A are set to 000b
- When the T3ACNT[2:0] bits in TITCNT1A match the T3ACOR[2:0] bits in TITCR1A

- MTU.TITCNT1B

Address(es): MTU.TITCNT1B 000C 1A31h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	T7VCNT[2:0]	TCIV7 Interrupt Counter	While the T7VEN bit in TITCR1B is set to 1, the count in these bits is incremented every time a TCIV7 interrupt occurs.	R
b3	—	Reserved	This bit is read as 0.	R
b6 to b4	T6ACNT[2:0]	TGIA6 Interrupt Counter	While the T6AEN bit in TITCR1B is set to 1, the count in these bits is incremented every time a TGIA6 interrupt occurs.	R
b7	—	Reserved	This bit is read as 0.	R

Note: To clear the TITCNT1B, set the TITCR1B.T6AEN and TITCR1B.T7VEN bits to 0.

T7VCNT[2:0] Bits (TCIV7 Interrupt Counter)

[Clearing conditions]

- When the TITM bit in TITMRB is 1
- When the T7VEN bit in TITCR1B is set to 0
- When the T7VCOR[2:0] bits in TITCR1B are set to 000b
- When the T7VCNT[2:0] bits in TITCNT1B match the T7VCOR[2:0] bits in TITCR1B

T6ACNT[2:0] Bits (TGIA6 Interrupt Counter)

[Clearing conditions]

- When the TITM bit in TITMRB is 1
- When the T6AEN bit in TITCR1B is set to 0
- When the T6ACOR[2:0] bits in TITCR1B are set to 000b
- When the T6ACNT[2:0] bits in TITCNT1B match the T6ACOR[2:0] bits in TITCR1B

22.2.41 Timer Interrupt Skipping Set Register 2m (TITCR2m) (m = A, B)

- MTU.TITCR2A

Address(es): MTU.TITCR2A 000C 123Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TRG4COR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	TRG4COR[2:0]	TRG4AN/TRG4BN Interrupt Skipping Count Setting	These bits specify the TRG4AN/TRG4BN interrupt skipping count within the range from 0 to 7. For details, refer to Table 22.59.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

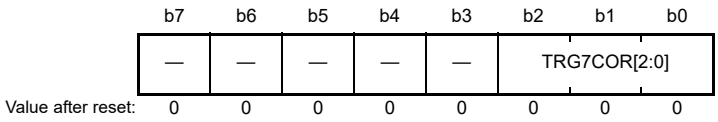
TITCR2A and TITCR2B specify the interrupt skipping count for TRG4AN and TRG4BN (TRG7AN and TRG7BN). This setting is valid only while TITMRA or TITMRB is set to 1.

Table 22.59 Setting of Interrupt Skipping Count by TRG4COR[2:0] Bits

Bit 2	Bit 1	Bit 0	
TRG4COR[2]	TRG4COR[1]	TRG4COR[0]	Description
0	0	0	Does not skip TRG4AN and TRG4BN interrupts.
0	0	1	Sets the TRG4AN and TRG4BN interrupt skipping count to 1.
0	1	0	Sets the TRG4AN and TRG4BN interrupt skipping count to 2.
0	1	1	Sets the TRG4AN and TRG4BN interrupt skipping count to 3.
1	0	0	Sets the TRG4AN and TRG4BN interrupt skipping count to 4.
1	0	1	Sets the TRG4AN and TRG4BN interrupt skipping count to 5.
1	1	0	Sets the TRG4AN and TRG4BN interrupt skipping count to 6.
1	1	1	Sets the TRG4AN and TRG4BN interrupt skipping count to 7.

- MTU.TITCR2B

Address(es): MTU.TITCR2B 000C 1A3Bh



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	TRG7COR[2:0]	TRG7AN/TRG7BN Interrupt Skipping Count Setting	These bits specify the TRG7AN/TRG7BN interrupt skipping count within the range from 0 to 7. For details, refer to Table 22.60.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Table 22.60 Setting of Interrupt Skipping Count by TRG7COR[2:0] Bits

Bit 2	Bit 1	Bit 0	
TRG7COR[2]	TRG7COR[1]	TRG7COR[0]	Description
0	0	0	Does not skip TRG7AN and TRG7BN interrupts.
0	0	1	Sets the TRG7AN and TRG7BN interrupt skipping count to 1.
0	1	0	Sets the TRG7AN and TRG7BN interrupt skipping count to 2.
0	1	1	Sets the TRG7AN and TRG7BN interrupt skipping count to 3.
1	0	0	Sets the TRG7AN and TRG7BN interrupt skipping count to 4.
1	0	1	Sets the TRG7AN and TRG7BN interrupt skipping count to 5.
1	1	0	Sets the TRG7AN and TRG7BN interrupt skipping count to 6.
1	1	1	Sets the TRG7AN and TRG7BN interrupt skipping count to 7.

22.2.42 Timer Interrupt Skipping Counter 2m (TITCNT2m) (m = A, B)

- MTU.TITCNT2A

Address(es): MTU.TITCNT2A 000C 123Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TRG4CNT[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	TRG4CNT[2:0]	TRG4AN/TRG4BN Interrupt Counter	These bits start counting from the value set in TRG4COR[2:0] and the count decrements every time TRG4AN or TRG4BN is generated. When the count reaches 0 and is reloaded, the TRG4AN and TRG4BN interrupts become valid.	R
b7 to b3	—	Reserved	These bits are read as 0.	R

TITCNT2A and TITCNT2B start counting from the values set in the TRG4COR[2:0] and TRG7COR[2:0] bits and the count decrements every time TRG4AN or TRG4BN (TITCNT2A) is generated or TRG7AN or TRG7BN (TITCNT2B) is generated. When the count reaches 0 and is reloaded, the TRG4AN and TRG4BN interrupts or the TRG7AN and TRG7BN interrupts become valid.

TRG4CNT[2:0] Bits (TRG4AN/TRG4BN Interrupt Counter)

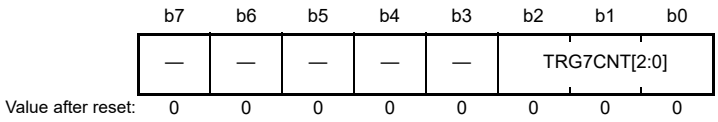
These bits start counting from the value set in the TRG4COR[2:0] bits and the count decrements every time a TRG4AN or TRG4BN interrupt is generated. When the count reaches 0 and is reloaded, the TRG4AN and TRG4BN interrupts become valid.

[Clearing conditions]

- When the TITM bit in TITMRA is 0
- When the TRG4COR[2:0] bits in TITCR2A are set to 000b
- When the count of TRG4AN and TRG4BN occurrence matches the TRG4COR[2:0] value in TITCR2A

- MTU.TITCNT2B

Address(es): MTU.TITCNT2B 000C 1A3Ch



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	TRG7CNT[2:0]	TRG7AN/TRG7BN Interrupt Counter	These bits start counting from the value set in TRG7COR[2:0] and the count decrements every time TRG7AN or TRG7BN is generated. When the count reaches 0 and is reloaded, the TRG7AN and TRG7BN interrupts become valid.	R
b7 to b3	—	Reserved	These bits are read as 0.	R

TRG7CNT[2:0] Bits (TRG7AN/TRG7BN Interrupt Counter)

These bits start counting from the value set in the TRG7COR[2:0] bits and the count decrements every time a TRG7AN or TRG7BN interrupt is generated. When the count reaches 0 and is reloaded, the TRG7AN and TRG7BN interrupts become valid.

[Clearing conditions]

- When the TITM bit in TITMRB is 0
- When the TRG7COR[2:0] bits in TITCR2B are set to 000b
- When the count of TRG7AN and TRG7BN occurrence matches the TRG7COR[2:0] value in TITCR2B

22.2.43 A/D Conversion Start Request Select Register 0 (TADSTRGR0)

Address(es): MTU.TADSTRGR0 000C 1D30h

	b7	b6	b5	b4	b3	b2	b1	b0
	TADSMEN0	—	—	TADSTRS0[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	TADSTRS0[4:0]	A/D Conversion Start Request Select for ADSM0 Pin Output Frame Synchronization Signal Generation	These bits select the A/D conversion start request for generating the frame synchronization signal to be output from the ADSM0 pin. Refer to Table 22.61 for the relationship between the A/D conversion start request and settings. Settings other than those listed in Table 22.61 are prohibited.	R/W
b6 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	TADSMEN0	ADSM0 Pin Output Enable	0: ADSM0 pin output disabled 1: ADSM0 pin output enabled	R/W

The TADSTRGR0 register selects the A/D conversion start request for generating the A/D conversion start request frame synchronization signal to be output from the ADSM0 pin.

22.2.44 A/D Conversion Start Request Select Register 1 (TADSTRGR1)

Address(es): MTU.TADSTRGR1 000C 1D32h

	b7	b6	b5	b4	b3	b2	b1	b0
	TADSMEN1	—	—	TADSTRS1[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	TADSTRS1[4:0]	A/D Conversion Start Request Select for ADSM1 Pin Output Frame Synchronization Signal Generation	These bits select the A/D conversion start request for generating the frame synchronization signal to be output from the ADSM1 pin. Refer to Table 22.61 for the relationship between the A/D conversion start request and settings. Settings other than those listed in Table 22.61 are prohibited.	R/W
b6 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	TADSMEN1	ADSM1 Pin Output Enable	0: ADSM1 pin output disabled 1: ADSM1 pin output enabled	R/W

The TADSTRGR1 register selects the A/D conversion start request for generating the A/D conversion start request frame synchronization signal to be output from the ADSM1 pin.

Table 22.61 Settings of A/D Conversion Start Request for Generating Frame Synchronization Signal (n = 0, 1)

TADSTRSn[4:0]					Source	Descriptions
[4]	[3]	[2]	[1]	[0]		
0	0	0	0	0	—	Source not selected
0	0	0	0	1	TRGA0N	Compare match/input capture in MTU0.TGRA
0	0	0	1	0	TRGA1N	Compare match/input capture in MTU1.TGRA
0	0	0	1	1	TRGA2N	Compare match/input capture in MTU2.TGRA
0	0	1	0	0	TRGA3N	Compare match/input capture in MTU3.TGRA
0	0	1	0	1	TRGA4N	Compare match/input capture in MTU4.TGRA or MTU4.TCNT underflow (trough) in complementary PWM mode
0	0	1	1	0	TRGA6N	Compare match/input capture in MTU6.TGRA
0	0	1	1	1	TRGA7N	Compare match/input capture in MTU7.TGRA or MTU7.TCNT underflow (trough) in complementary PWM mode
0	1	0	0	0	TRG0N	Compare match in MTU0.TGRE
0	1	0	0	1	TRG4AN	Compare match between MTU4.TADCORA and MTU4.TCNT
0	1	0	1	0	TRG4BN	Compare match between MTU4.TADCORB and MTU4.TCNT
0	1	1	0	0	TRG4ABN	Compare match between MTU4.TADCORA and MTU4.TCNT or compare match between MTU4.TADCORB and MTU4.TCNT (Interrupt skipping function 2 used)
0	1	1	0	1	TRG7AN	Compare match between MTU7.TADCORA and MTU7.TCNT
0	1	1	1	0	TRG7BN	Compare match between MTU7.TADCORB and MTU7.TCNT
1	0	0	0	0	TRG7ABN	Compare match between MTU7.TADCORA and MTU7.TCNT or compare match between MTU7.TADCORB and MTU7.TCNT (Interrupt skipping function 2 used)
1	0	0	0	1	TRGA9N	Compare match/input capture in MTU9.TGRA
1	0	0	1	0	TRG9N	Compare match in MTU9.TGRE
1	0	0	1	1	TRG9AEN	Compare match/input capture in MTU9.TGRA or compare match in MTU9.TGRE
1	0	1	0	0	TRG0AEN	Compare match/input capture in MTU0.TGRA or compare match in MTU0.TGRE
1	0	1	0	1	TRGA09N	Compare match/input capture in MTU0.TGRA or compare match/input capture in MTU9.TGRA
1	0	1	1	0	TRG09N	Compare match in MTU0.TGRE or compare match in MTU9.TGRE

22.3 Operation

22.3.1 Basic Functions

Each channel has TCNT and TGR register. TCNT performs up-counting, and is also capable of free-running operation, periodic counting, and external event counting.
Each TGR register can be used as an input capture register or an output compare register.

(1) Counter Operation

When one of bits CST0 to CST4 and CST9 in the TSTRA register, bits CST6 and CST7 in the TSTRB register, and bits CSTU5, CSTV5, and CSTW5 in the MTU5.TSTR register is set to 1, TCNT for the corresponding channel begins counting. TCNT can operate as a free-running counter, periodic counter, for example.

(a) Example of Count Operation Setting Procedure

Figure 22.5 shows an example of the count operation setting procedure.

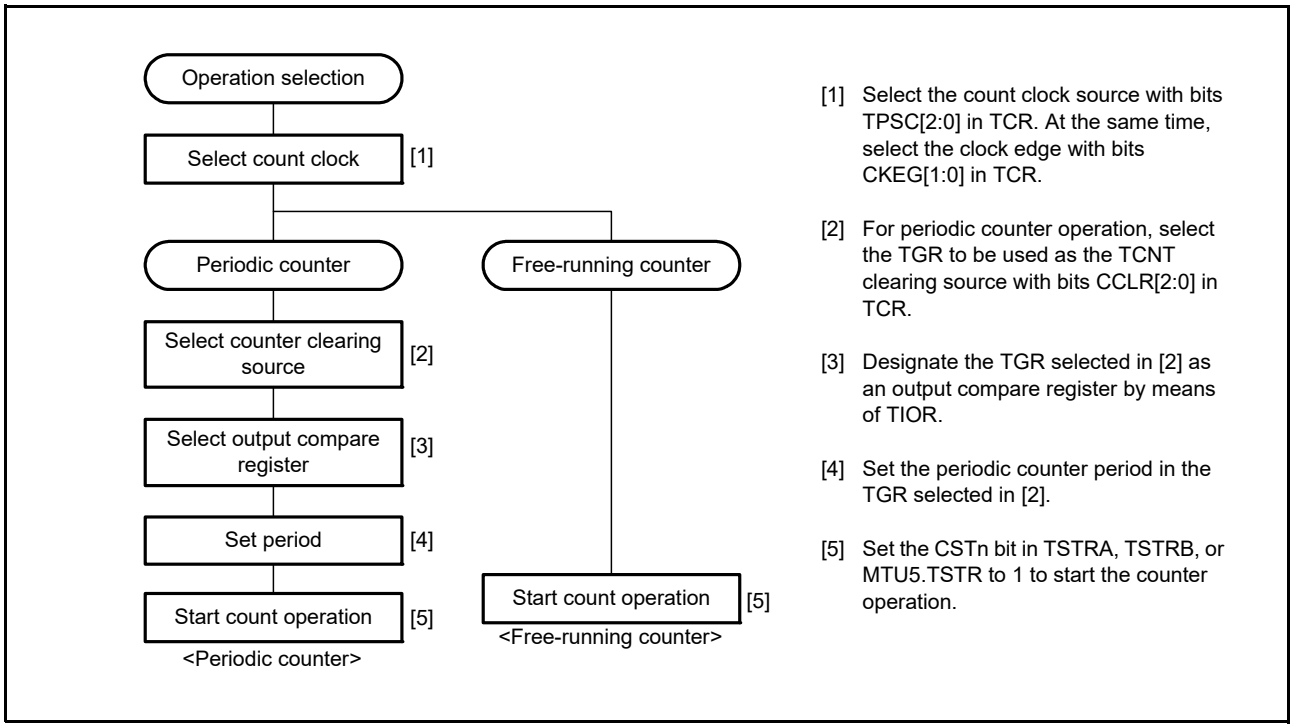


Figure 22.5 Example of Count Operation Setting Procedure

(b) Free-Running Count Operation and Periodic Count Operation

Immediately after a reset, the TCNT counters are all designated as free-running counters. When the CSTn bit in TSTRA, TSTRB, or MTU5.TSTR is set to 1, the corresponding TCNT counter starts up-count operation as a free-running counter. When TCNT overflows (from FFFFh to 0000h), an interrupt request is issued to the CPU if the corresponding TIER.TCIEV bit is 1. After an overflow, TCNT starts counting up again from 0000h.
Figure 22.6 illustrates free-running counter operation.

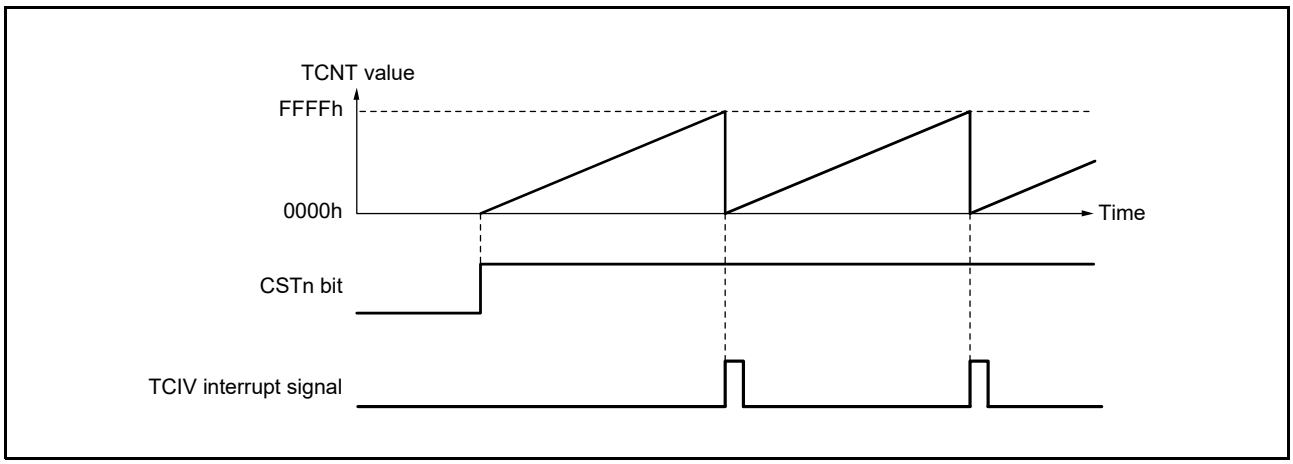


Figure 22.6 Free-Running Counter Operation

When compare match is selected as the TCNT clearing source, TCNT for the relevant channel performs periodic count operation. TGR for setting the period is designated as an output compare register, and counter clearing by compare match is selected by means of bits CCLR[2:0] in TCR. After the settings have been made, TCNT starts up-count operation as a periodic counter when the CSTn bit in TSTRA, TSTRB or MTU5.TSTR is set to 1. When the count matches the value in TGR, TCNT becomes 0000h.
If the value of the corresponding TIER.TGIE bit is 1 at this point, an interrupt request is issued to the CPU. After a compare match, TCNT starts counting up again from 0000h.
Figure 22.7 illustrates periodic counter operation.

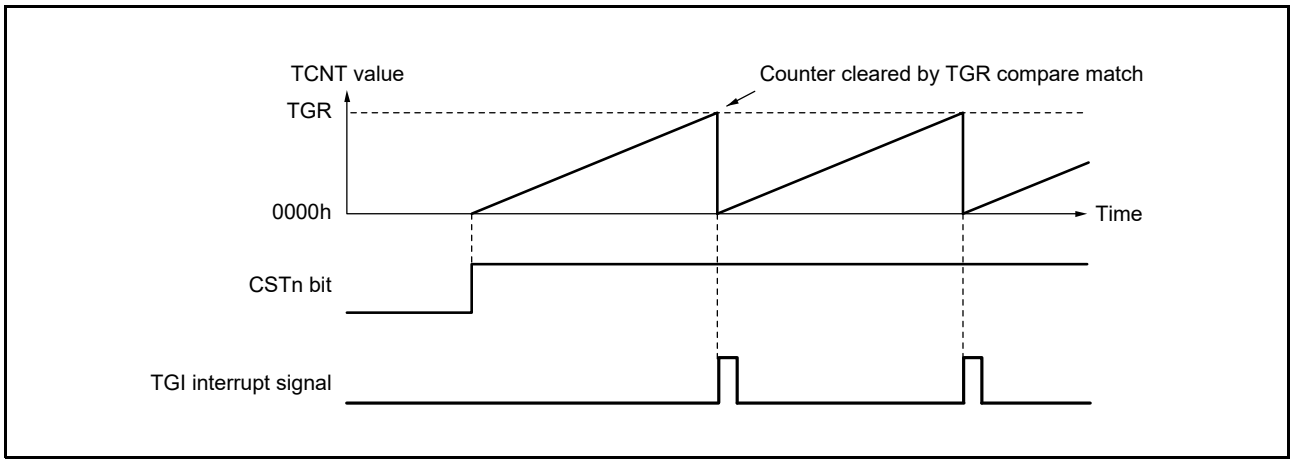


Figure 22.7 Periodic Counter Operation

(2) Waveform Output by Compare Match

Upon compare match, low, high, or toggle output from the corresponding pin can be performed.

(a) Example of Procedure for Setting Waveform Output by Compare Match

Figure 22.8 shows an example of the procedure for setting waveform output by compare match

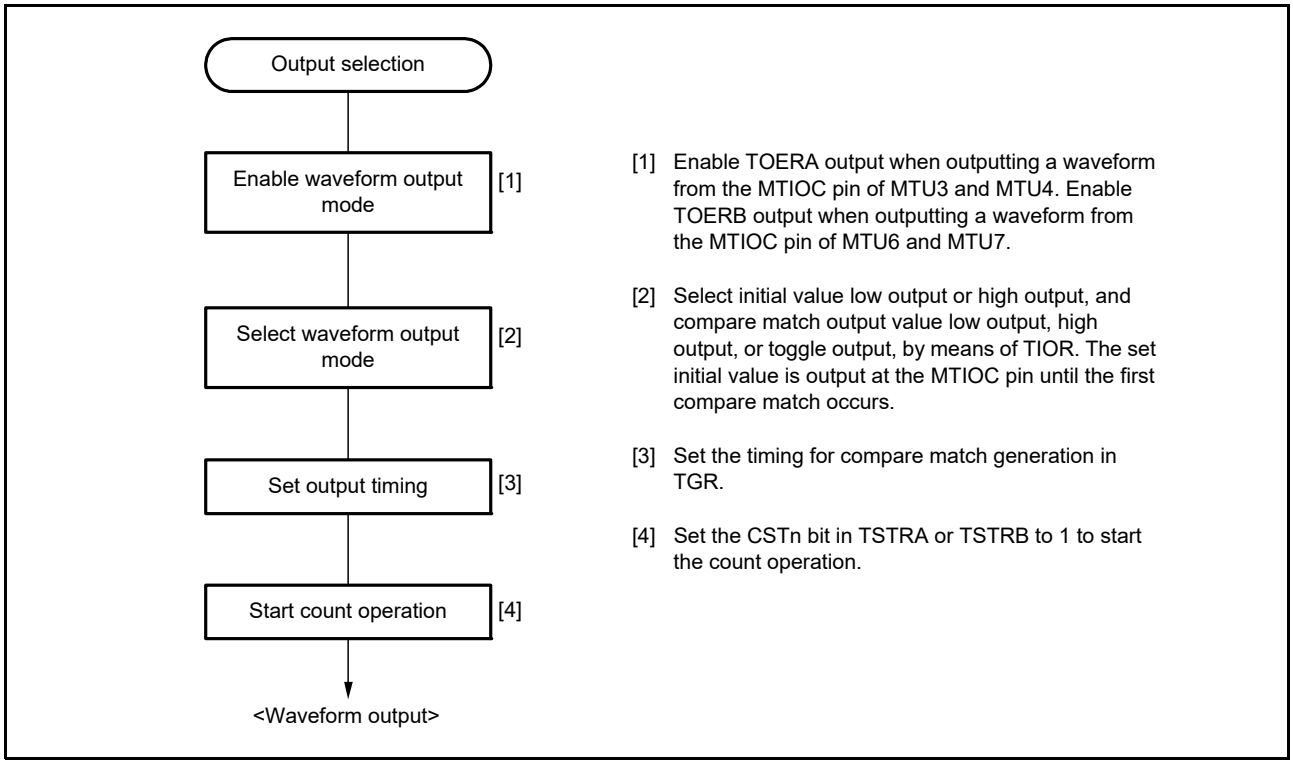


Figure 22.8 Example of Procedure for Setting Waveform Output by Compare Match

(b) Examples of Waveform Output Operation

Figure 22.9 shows an example of low output and high output.
In this example, TCNT has been designated as a free-running counter, and settings have been made so that high is output by compare match A and low is output by compare match B. When the pin level is the same as the specified level, the pin level does not change.

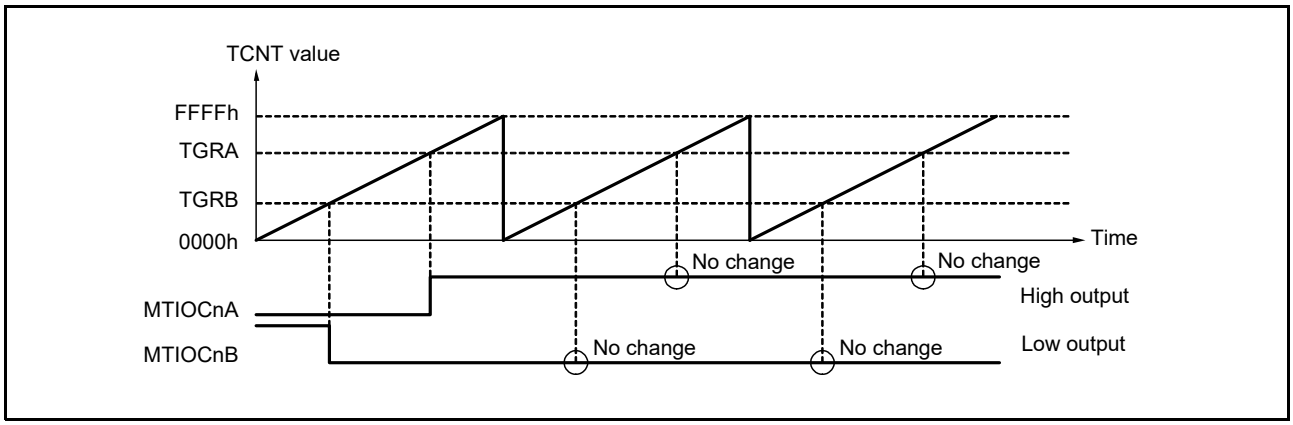


Figure 22.9 Example of low output and high output Operation (n = 0 to 4, 6, 7, 9)

Figure 22.10 shows an example of toggle output.
In this example, TCNT has been designated as a periodic counter (with counter clearing on compare match B), and settings have been made so that the output is toggled by both compare match A and compare match B.

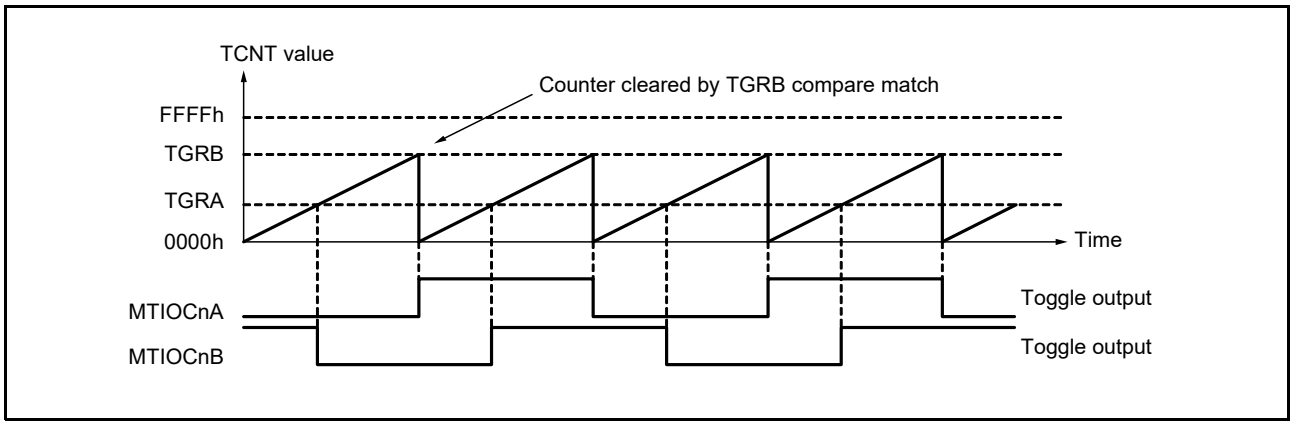


Figure 22.10 Example of Toggle Output Operation (n = 0 to 4, 6, 7, 9)

(3) Input Capture Function

The TCNT value can be transferred to TGR on detection of the MTIOCnm pin (n = 0 to 4, 6, 7, 9; m = A to D) or MTIC5m pin (m = U, V, W) input edge.
The rising edge, falling edge, or both edges can be selected as the detection edge. For MTU0, MTU1, and MTU9, another channel's count clock or compare match signal can also be specified as the input capture source.

Note: When another channel's count clock is used as the input capture input for MTU0, MTU1, and MTU9, PCLKC/1 should not be selected as the count clock used for input capture input. Input capture will not be generated if PCLKC/1 is selected.

(a) Example of Input Capture Operation Setting Procedure

Figure 22.11 shows an example of the input capture operation setting procedure.

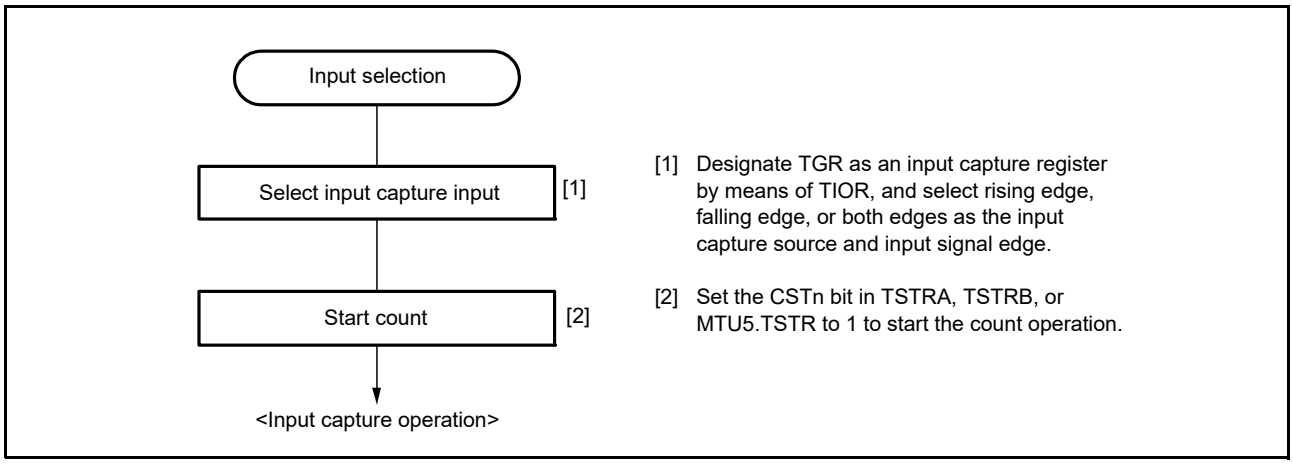


Figure 22.11 Example of Input Capture Operation Setting Procedure

(b) Example of Input Capture Operation

Figure 22.12 shows an example of input capture operation.
In this example, both rising and falling edges have been selected as the MTIOCnA pin input capture input edge, the falling edge has been selected as the MTIOCnB pin input capture input edge, and counter clearing by TGRB input capture has been designated for TCNT. (n = 0 to 4, 6, 7, 9)

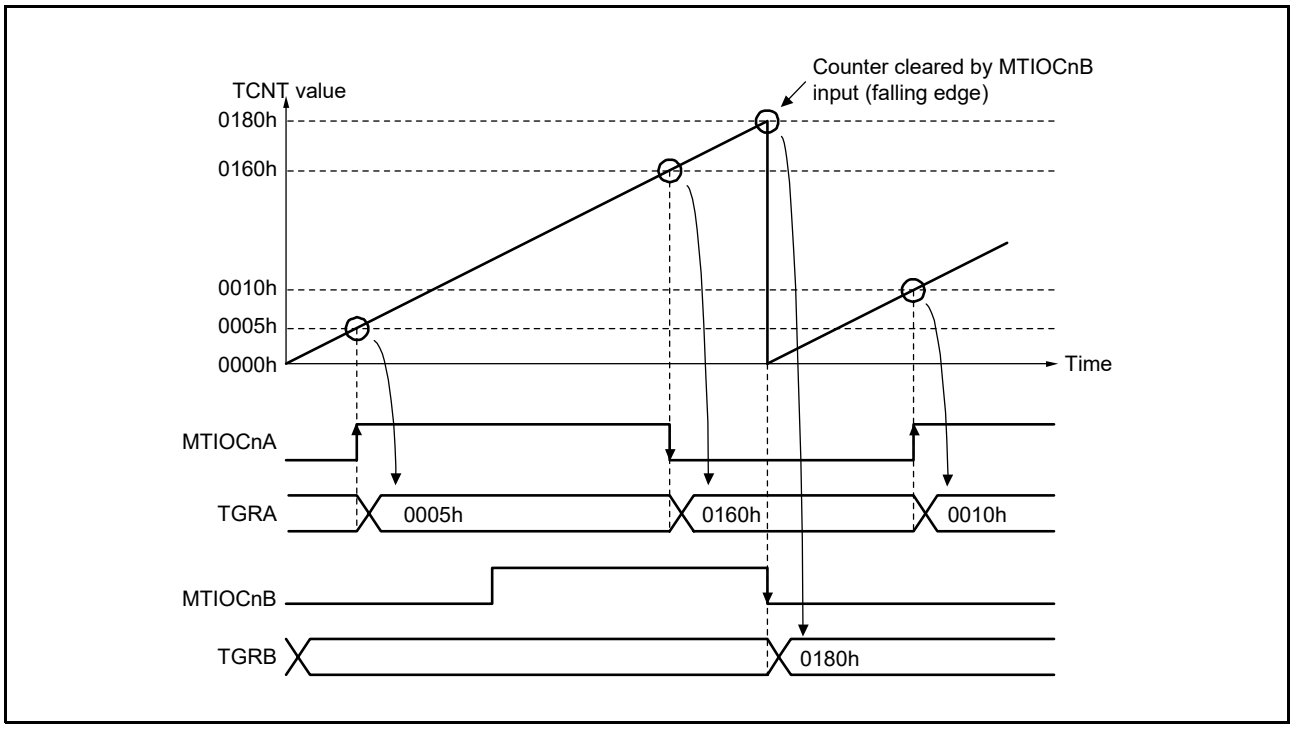


Figure 22.12 Example of Input Capture Operation (n = 0 to 4, 6, 7, 9)

22.3.2 Synchronous Operation

In synchronous operation, the values in multiple TCNT counters can be modified simultaneously (synchronous setting). In addition, multiple TCNT counters can be cleared simultaneously (synchronous clearing) by making the appropriate setting in TCR.
Synchronous operation can increase the number of TGR registers assigned to the same time base.
MTU0 to MTU4, MTU6, MTU7, and MTU9 can all be designated for synchronous operation. MTU5 cannot be used for synchronous operation.

(1) Example of Synchronous Operation Setting Procedure

Figure 22.13 shows an example of the synchronous operation setting procedure.

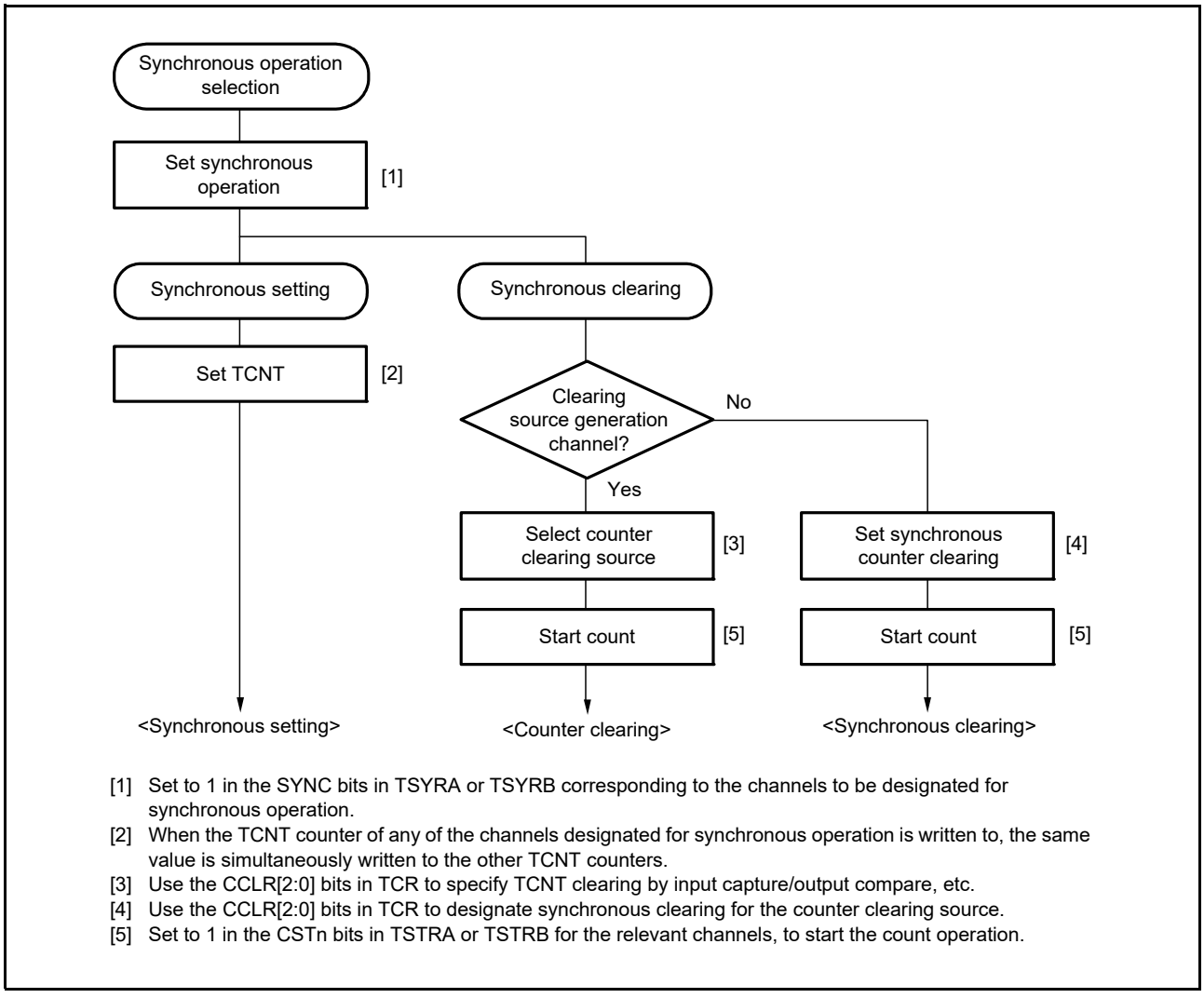


Figure 22.13 Example of Synchronous Operation Setting Procedure

(2) Example of Synchronous Operation

Figure 22.14 shows an example of synchronous operation.
In this example, synchronous operation and PWM mode 1 have been designated for MTU0 to MTU 2, MTU0.TGRB compare match has been set as the counter clearing source in MTU0, and synchronous clearing has been set for the counter clearing source in MTU1 and MTU2.
Three-phase PWM waveforms are output from pins MTIOC0A, MTIOC1A, and MTIOC2A. At this time, synchronous setting and synchronous clearing by MTU0.TGRB compare match are performed for the TCNT counters in MTU0 to MTU2, and the data set in MTU0.TGRB is used as the PWM period.
For details of PWM modes, refer to section 22.3.5, PWM Modes.

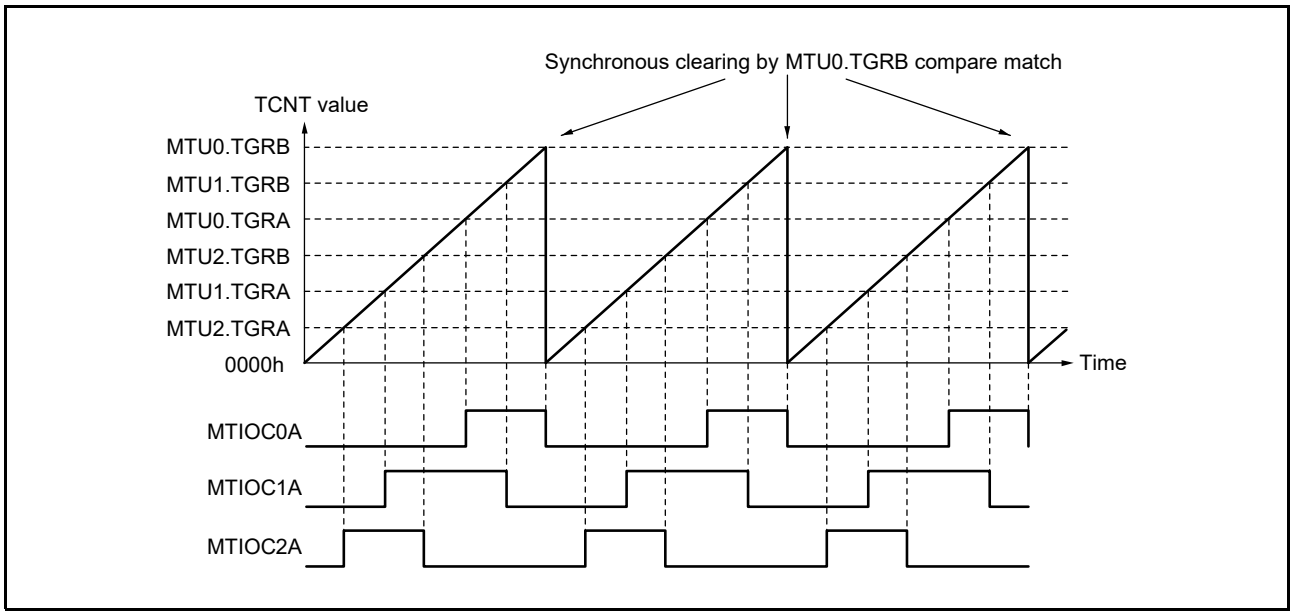


Figure 22.14 Example of Synchronous Operation

22.3.3 Buffer Operation

Buffer operation, provided for MTU0, MTU3, MTU4, MTU6, MTU7, and MTU9, enables TGRC and TGRD to be used as buffer registers. In MTU0 and MTU9, TGRF can also be used as a buffer register.
Buffer operation differs depending on whether TGR has been designated as an input capture register or as a compare match register.

Note: MTU0.TGRE and MTU9.TGRE cannot be designated as an input capture register and can only operate as a compare match register.

Table 22.62 shows the register combinations used in buffer operation.

Table 22.62 Register Combinations in Buffer Operation

Channel	Timer General Register	Buffer Register
MTU0	TGRA	TGRC
	TGRB	TGRD
	TGRE	TGRF
MTU3	TGRA	TGRC
	TGRB	TGRD
MTU4	TGRA	TGRC
	TGRB	TGRD
MTU6	TGRA	TGRC
	TGRB	TGRD
MTU7	TGRA	TGRC
	TGRB	TGRD
MTU9	TGRA	TGRC
	TGRB	TGRD
	TGRE	TGRF

- When TGR is an output compare register

When a compare match occurs, the value in the buffer register for the corresponding channel is transferred to the timer general register.
This operation is illustrated in Figure 22.15.

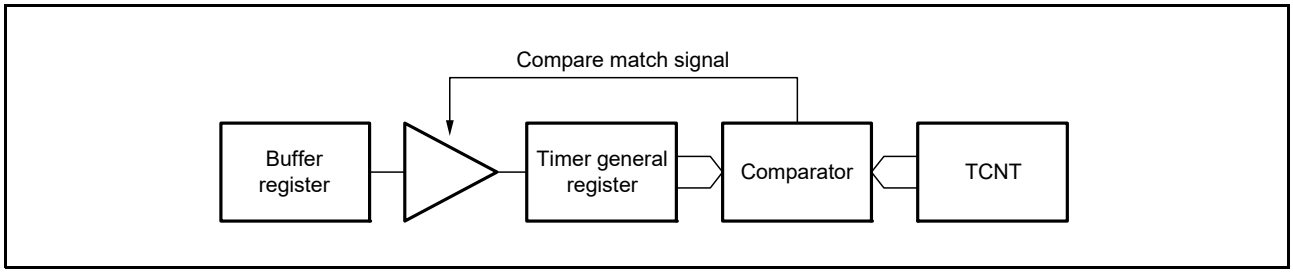


Figure 22.15 Compare Match Buffer Operation

- When TGR is an input capture register

When an input capture occurs, the value in TCNT is transferred to TGR and the value previously held in TGR is transferred to the buffer register.
This operation is illustrated in Figure 22.16.

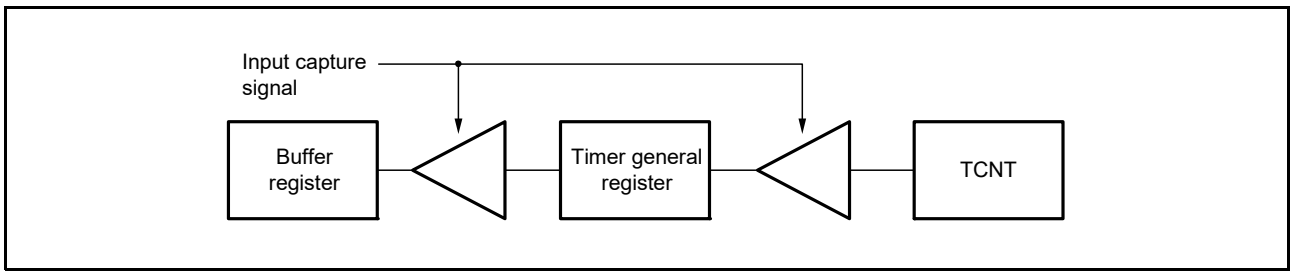


Figure 22.16 Input Capture Buffer Operation

(1) Example of Buffer Operation Setting Procedure

Figure 22.17 shows an example of the buffer operation setting procedure.

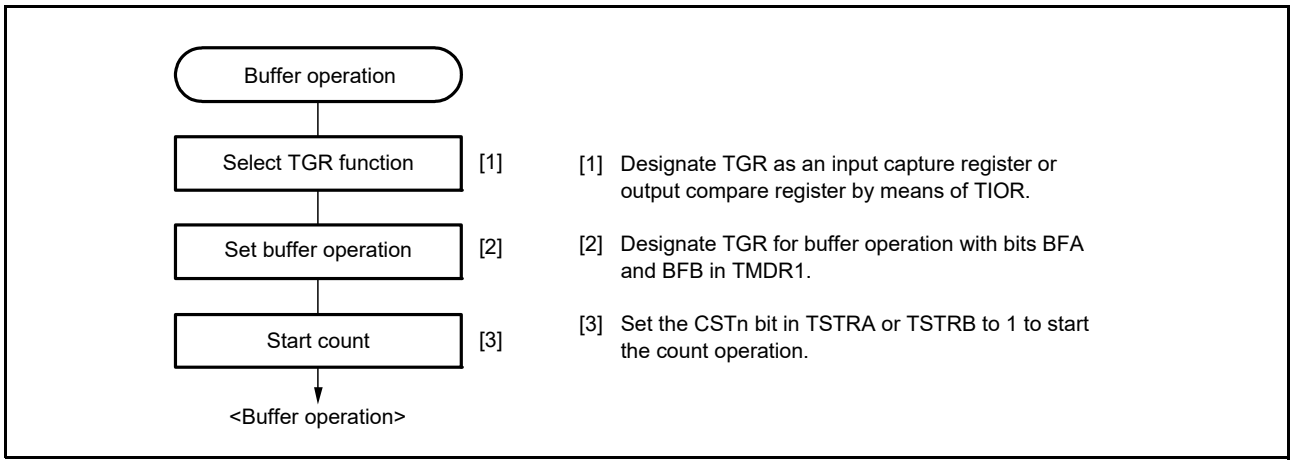


Figure 22.17 Example of Buffer Operation Setting Procedure

(2) Examples of Buffer Operation

(a) When TGR is an Output Compare Register

Figure 22.18 shows an operation example in which PWM mode 1 has been designated for MTU0, and buffer operation has been designated for TGRA and TGRC. The settings used in this example are TCNT clearing by compare match B, high output at compare match A, and low output at compare match B. In this example, the TTSA bit in TBTM is set to 0. As buffer operation has been set, when compare match A occurs, the output changes and the value in buffer register TGRC is simultaneously transferred to timer general register TGRA. This operation is repeated each time compare match A occurs.
For details of PWM modes, refer to section 22.3.5, PWM Modes.

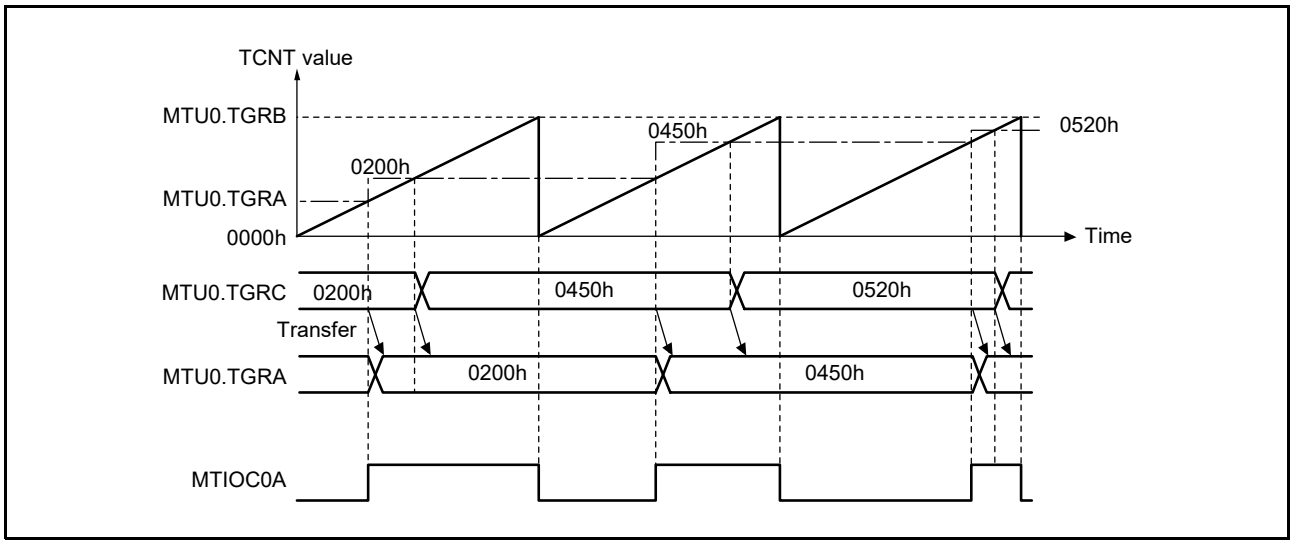


Figure 22.18 Example of Buffer Operation (1)

(b) When TGR is an Input Capture Register

Figure 22.19 shows an operation example in which TGRA has been designated as an input capture register, and buffer operation has been designated for TGRA and TGRC.

Counter clearing by TGRA input capture has been set for TCNT, and both rising and falling edges have been selected as the MTIOCnA pin input capture input edge. (n = 0 to 4, 6, 7, 9)

As buffer operation has been set, when the TCNT value is transferred to TGRA upon occurrence of input capture A, the value previously stored in TGRA is simultaneously transferred to TGRC.

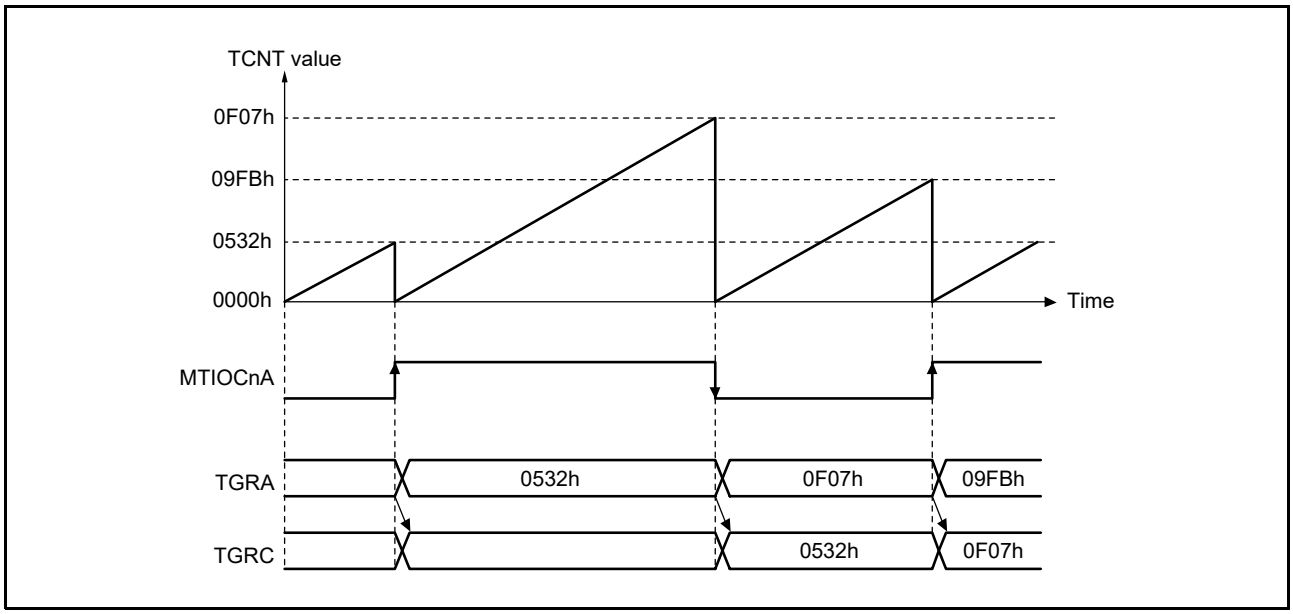


Figure 22.19 Example of Buffer Operation (2) (n = 0 to 4, 6, 7, 9)

(3) Selecting Timing for Transfer from Buffer Registers to Timer General Registers in Buffer Operation

The timing for transfer from buffer registers to timer general registers can be selected in PWM mode 1 or 2 for MTU0 and MTU9 or in PWM mode 1 for MTU3, MTU4, MTU6, and MTU7 by setting the buffer operation transfer mode registers (MTUn.TBTM (n = 0, 3, 4, 6, 7, 9)). Either compare match (value after reset) or TCNT clearing can be selected for the transfer timing. TCNT clearing as transfer timing is one of the following cases.

- When TCNT overflows (FFFFh to 0000h)
- When 0000h is written to TCNT during counting
- When TCNT becomes 0000h under the condition specified in the CCLR[2:0] bits in TCR

Note: TBTM must be modified only while TCNT stops.

Figure 22.20 shows an operation example in which PWM mode 1 is designated for MTU0 and buffer operation is designated for MTU0.TGRA and MTU0.TGRC. The settings used in this example are MTU0.TCNT clearing by compare match B, high output at compare match A, and low output at compare match B. The TTSA bit in MTU0.TBTM is set to 1.

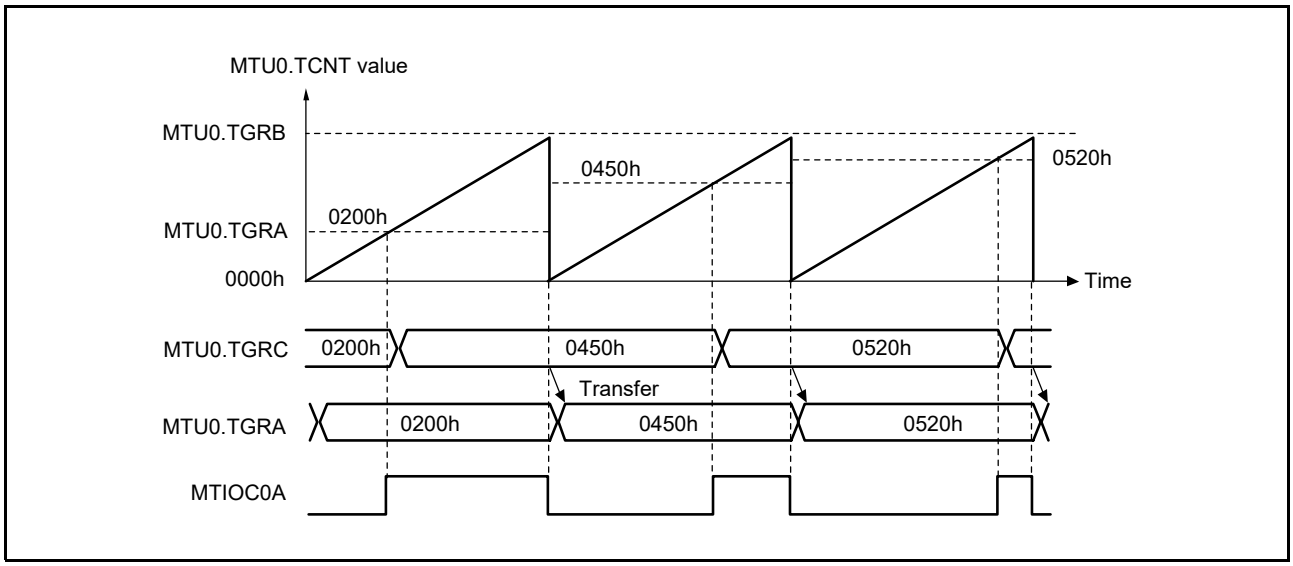


Figure 22.20 Example of Buffer Operation When MTU0.TCNT Clearing is Selected for MTU0.TGRC to MTU0.TGRA Transfer Timing

22.3.4 Cascaded Operation

In cascaded operation, two 16-bit counters in different channels are used together as a 32-bit counter.
There are two functions for connecting MTU1 and MTU2 to use as a 32-bit counter: cascade connection to be set when the MTU1.TMDR3.LWA bit is 0, and cascade connection 32-bit phase counting mode to be set when the MTU1.TMDR3.LWA bit is 1. For details on cascade connection 32-bit phase counting mode, refer to section 22.3.6.2, Cascade Connection 32-Bit Phase Counting Mode. This section describes the cascade connection function to be set when the MTU1.TMDR3.LWA bit is 0.
This function operates when the MTU1.TMDR3.LWA bit is set to 0 and the MTU1.TCR.TPSC[2:0] bits are set so that MTU1.TCNT counts at an overflow/underflow of MTU2.TCNT. Underflow occurs only when the MTU2 to which the lower 16 bits allocated is in phase counting mode.
Table 22.63 shows the register combinations used in cascaded operation.

Note: When phase counting mode is set for MTU1, the count clock setting is invalid and the counters operate independently in phase counting mode.

Table 22.63 Cascaded Combinations

Combination	Upper 16 Bits	Lower 16 Bits
MTU1 and MTU2	MTU1.TCNT	MTU2.TCNT

For simultaneous input capture of MTU1.TCNT and MTU2.TCNT during cascaded operation, additional input capture input pins can be specified by the input capture control register (TICCR). The input-capture condition is of edges in the signal produced by taking the logical OR of the input level on the main input pin and the input level on the added input pin. Accordingly, if either is at the high level, a change in the level of the other will not produce an edge for detection. For details, refer to (4), Cascaded Operation Example (c). For input capture in cascade connection, refer to section 22.6.21, Simultaneous Input Capture in MTU1.TCNT and MTU2.TCNT in Cascade Connection.
Table 22.64 shows the TICCR setting and input capture input pins.

Table 22.64 TICCR Setting and Input Capture Input Pins

Target Input Capture	TICCR Setting	Input Capture Input Pin
Input capture from MTU1.TCNT to MTU1.TGRA	I2AE bit = 0 (Initial value)	MTIOC1A
	I2AE bit = 1	MTIOC1A, MTIOC2A
Input capture from MTU1.TCNT to MTU1.TGRB	I2BE bit = 0 (Initial value)	MTIOC1B
	I2BE bit = 1	MTIOC1B, MTIOC2B
Input capture from MTU2.TCNT to MTU2.TGRA	I1AE bit = 0 (Initial value)	MTIOC2A
	I1AE bit = 1	MTIOC2A, MTIOC1A
Input capture from MTU2.TCNT to MTU2.TGRB	I1BE bit = 0 (Initial value)	MTIOC2B
	I1BE bit = 1	MTIOC2B, MTIOC1B

(1) Example of Cascaded Operation Setting Procedure

Figure 22.21 shows an example of the cascaded operation setting procedure.

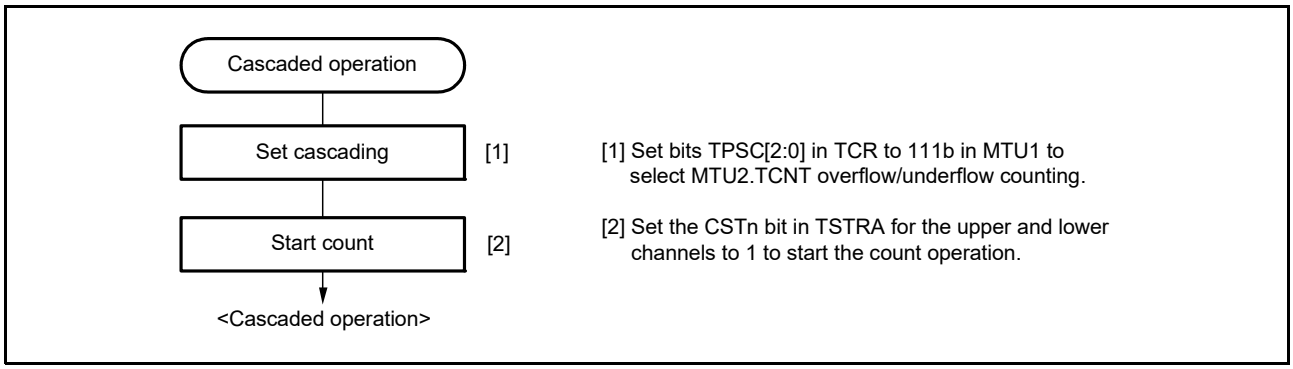


Figure 22.21 Cascaded Operation Setting Procedure

(2) Cascaded Operation Example (a)

Figure 22.22 shows the operation when MTU1.TCNT is set for counting at MTU2.TCNT overflow/underflow and MTU2 is set for phase counting mode 1 while MTU1.TCNT and MTU2.TCNT are cascaded.
MTU1.TCNT is incremented by MTU2.TCNT overflow and decremented by MTU2.TCNT underflow.

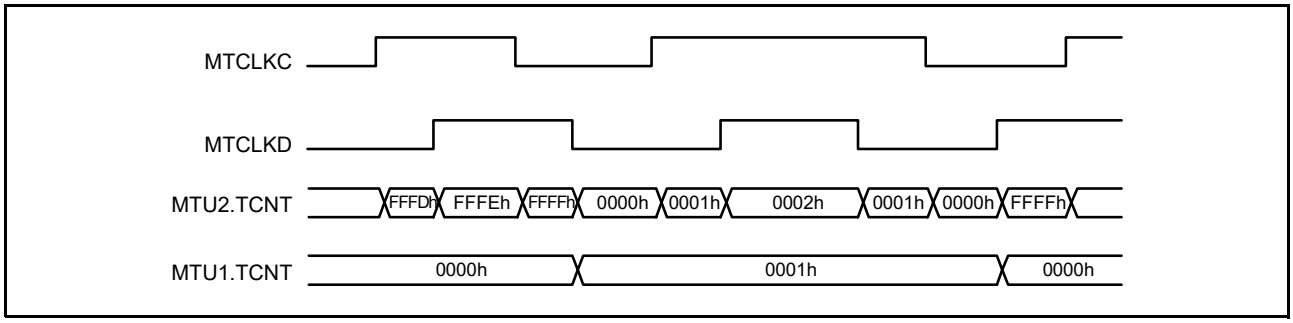


Figure 22.22 Cascaded Operation Example (a)

(3) Cascaded Operation Example (b)

Figure 22.23 illustrates the operation when MTU1.TCNT and MTU2.TCNT have been cascaded and the I2AE bit in TICCR has been set to 1 to include the MTIOC2A pin in the MTU1.TGRA input capture conditions. In this example, the MTU1.TIOR.IOA[3:0] bits have selected the MTIOC1A rising edge for the input capture timing while the MTU2.TIOR.IOA[3:0] bits have selected the MTIOC2A rising edge for the input capture timing.
Under these conditions, the rising edge of both MTIOC1A and MTIOC2A is used for the MTU1.TGRA input capture condition. For the MTU2.TGRA input capture condition, the MTIOC2A rising edge is used.

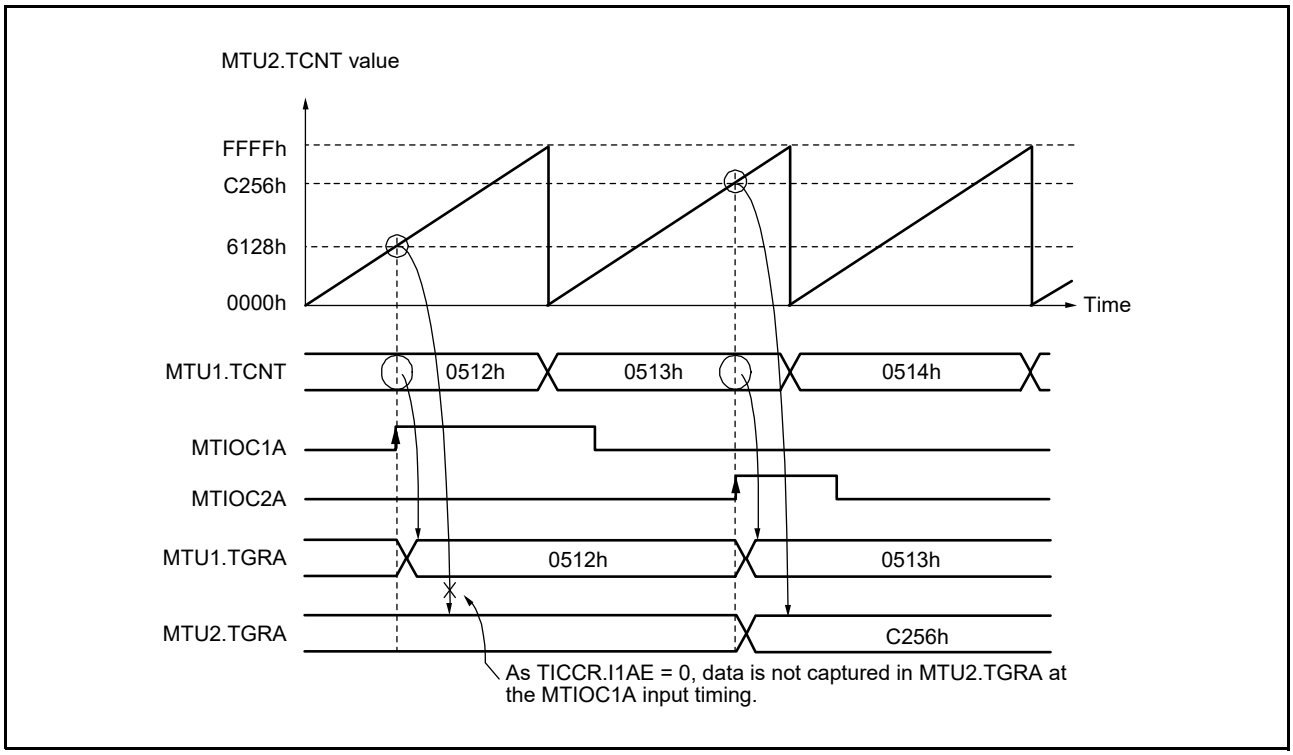
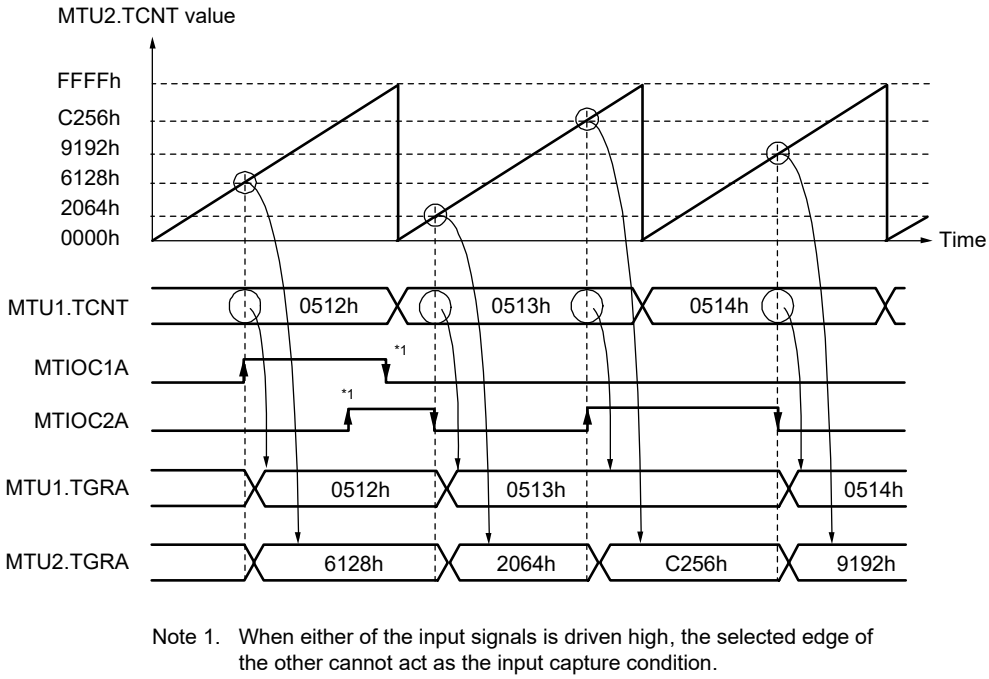


Figure 22.23 Cascaded Operation Example (b)

(4) Cascaded Operation Example (c)

Figure 22.24 illustrates the operation when MTU1.TCNT and MTU2.TCNT have been cascaded and the TICCR.I2AE and I1AE bits have been set to 1 to include the MTIOC2A and MTIOC1A pins in the MTU1.TGRA and MTU2.TGRA input capture conditions, respectively. In this example, the IOA[3:0] bits in both MTU1.TIOR and MTU2.TIOR have selected both the rising and falling edges for the input capture timing. Under these conditions, the ORed result of MTIOC1A and MTIOC2A input is used for the MTU1.TGRA and MTU2.TGRA input capture conditions.



Note 1. When either of the input signals is driven high, the selected edge of the other cannot act as the input capture condition.

Figure 22.24 Cascaded Operation Example (c)

(5) Cascaded Operation Example (d)

Figure 22.25 illustrates the operation when MTU1.TCNT and MTU2.TCNT have been cascaded and the TICCR.I2AE bit has been set to 1 to include the MTIOC2A pin in the MTU1.TGRA input capture conditions. In this example, the IOA[3:0] bits in MTU1.TIOR have selected occurrence of MTU0.TGRA compare match or input capture for the input capture timing while the IOA[3:0] bits in MTU2.TIOR have selected the MTIOC2A rising edge for the input capture timing.

Under these conditions, as MTU1.TIOR has selected occurrence of MTU0.TGRA compare match or input capture for the input capture timing, the MTIOC2A edge is not used for MTU1.TGRA input capture condition although the I2AE bit in TICCR has been set to 1.

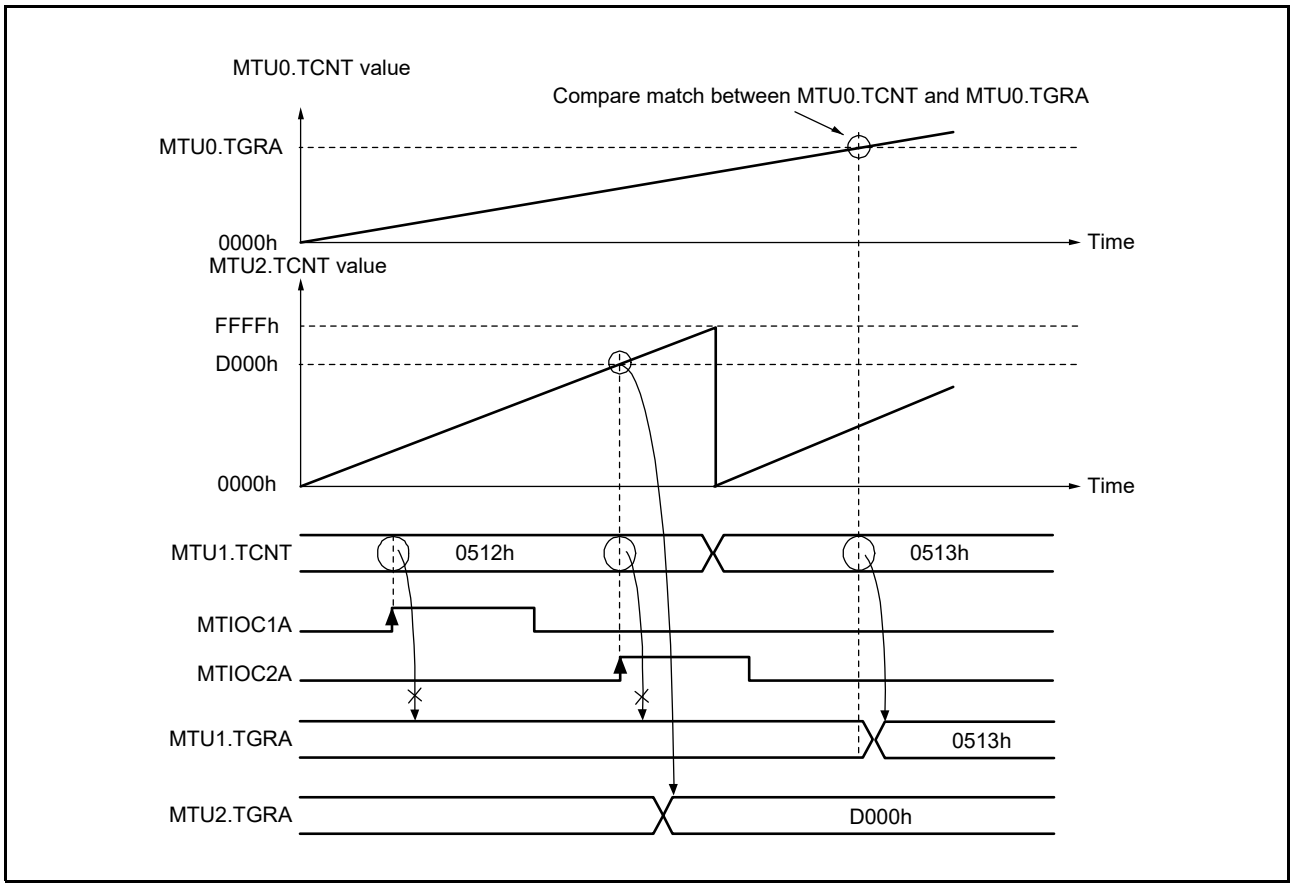


Figure 22.25 Cascaded Operation Example (d)

22.3.5 PWM Modes

PWM modes are provided to output PWM waveforms from the external pins. The output level can be selected as low, high, or toggle output in response to a compare match of each TGR.
PWM waveforms in the range of 0% to 100% duty cycle can be output according to the TGR settings.
By designating TGR compare match as the counter clearing source, the PWM period can be specified in that register.
Every channel except MTU5 can be set to PWM mode independently. Channels set to PWM mode can perform synchronous operation with each other or other channels set to any other mode.
There are two PWM modes as described below.

(a) PWM Mode 1

PWM waveforms are output from the MTIOCnA and MTIOCnC pins by pairing TGRA with TGRB and TGRC with TGRD. The levels specified by the TIOR.IOA[3:0] and IOC[3:0] bits are output from the MTIOCnA and MTIOCnC pins at compare matches A and C, and the level specified by the TIOR.IOB[3:0] and IOD[3:0] bits are output at compare matches B and D (n = 0 to 4, 6, 7, 9). The initial output value is set in TGRA or TGRC. If the values set in paired TGRs are identical, the output value does not change even when a compare match occurs.
In PWM mode 1, PWM waveforms in up to 14 phases can be output.

(b) PWM Mode 2

PWM waveform output is generated using one TGR as the period register and the others as duty registers. The level specified in TIOR is output at compare matches. Upon counter clearing by a period register compare match, the initial value set in TIOR is output from each pin. If the values set in the period and duty registers are identical, the output value does not change even when a compare match occurs.

Up to 12 phases of PWM waveforms can be output by combining synchronous clearing of channels that cannot be set to PWM mode 2 as synchronous operation.

The correspondence between PWM output pins and registers is shown in Table 22.65.

Table 22.65 PWM Output Registers and Output Pins

		Output Pins	
Channel	**Register**	**PWM Mode 1**	**PWM Mode 2**
MTU0	TGRA	MTIOC0A	MTIOC0A
	TGRB		MTIOC0B
	TGRC	MTIOC0C	MTIOC0C
	TGRD		MTIOC0D
MTU1	TGRA	MTIOC1A	MTIOC1A
	TGRB		MTIOC1B
MTU2	TGRA	MTIOC2A	MTIOC2A
	TGRB		MTIOC2B
MTU3	TGRA	MTIOC3A	Setting prohibited
	TGRB		
	TGRC	MTIOC3C	
	TGRD		
MTU4	TGRA	MTIOC4A	
	TGRB		
	TGRC	MTIOC4C	
	TGRD		
MTU6	TGRA	MTIOC6A	
	TGRB		
	TGRC	MTIOC6C	
	TGRD		
MTU7	TGRA	MTIOC7A	
	TGRB		
	TGRC	MTIOC7C	
	TGRD		
MTU9	TGRA	MTIOC9A	MTIOC9A
	TGRB		MTIOC9B
	TGRC	MTIOC9C	MTIOC9C
	TGRD		MTIOC9D

Note: In PWM mode 2, PWM waveform output is not possible for the TGR register in which the PWM period is set.

(1) Example of PWM Mode Setting Procedure

Figure 22.26 shows an example of the PWM mode setting procedure.

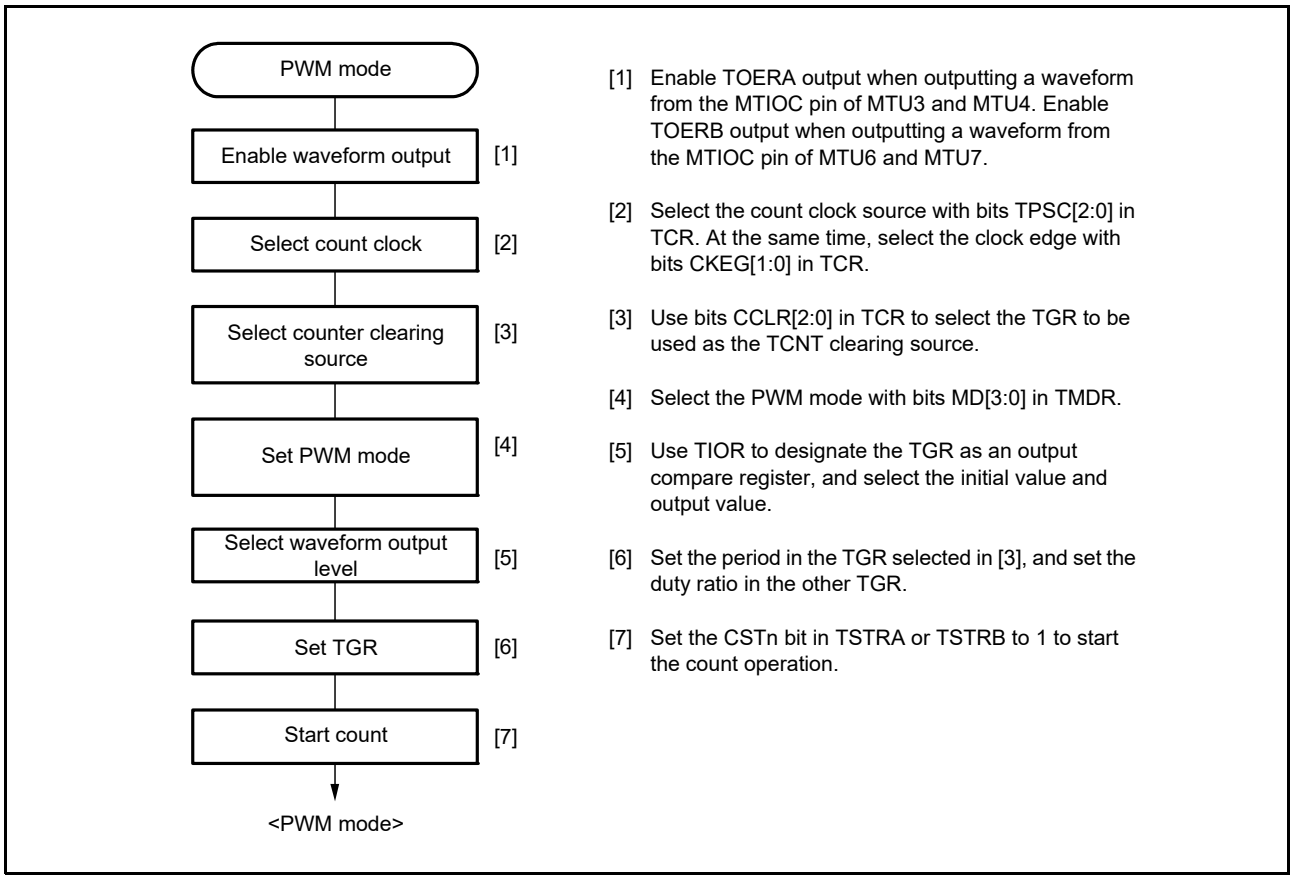


Figure 22.26 Example of PWM Mode Setting Procedure

(2) Examples of PWM Mode Operation

Figure 22.27 shows an example of operation in PWM mode 1.

In this example, TGRA compare match is set as the TCNT clearing source, 0 is set as the initial output value and output value for TGRA, and 1 is set as the output value for TGRB.

In this case, the value set in TGRA is used as the period, and the value set in TGRB is used as the duty ratio.

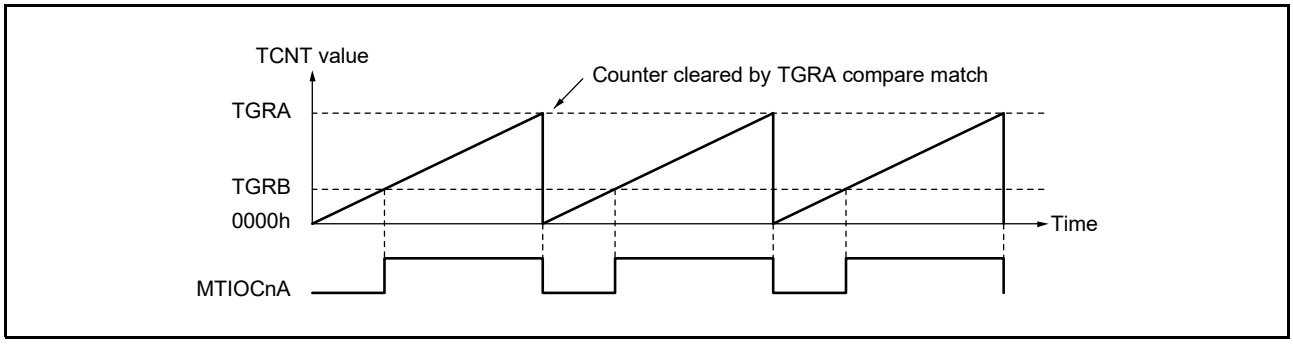


Figure 22.27 Example of PWM Mode 1 Operation (n = 0 to 4, 6, 7, 9)

Figure 22.28 shows an example of operation in PWM mode 2.
In this example, synchronous operation is designated for MTU0 and MTU1, MTU1.TGRB compare match is set as the TCNT clearing source, and low is set as the initial output value and high as the output value for the other TGR registers (MTU0.TGRA to MTU0.TGRD and MTU1.TGRA), outputting 5-phase PWM waveforms.
In this case, the value set in MTU1.TGRB is used as the period, and the values set in the other TGRs are used as the duty ratio.

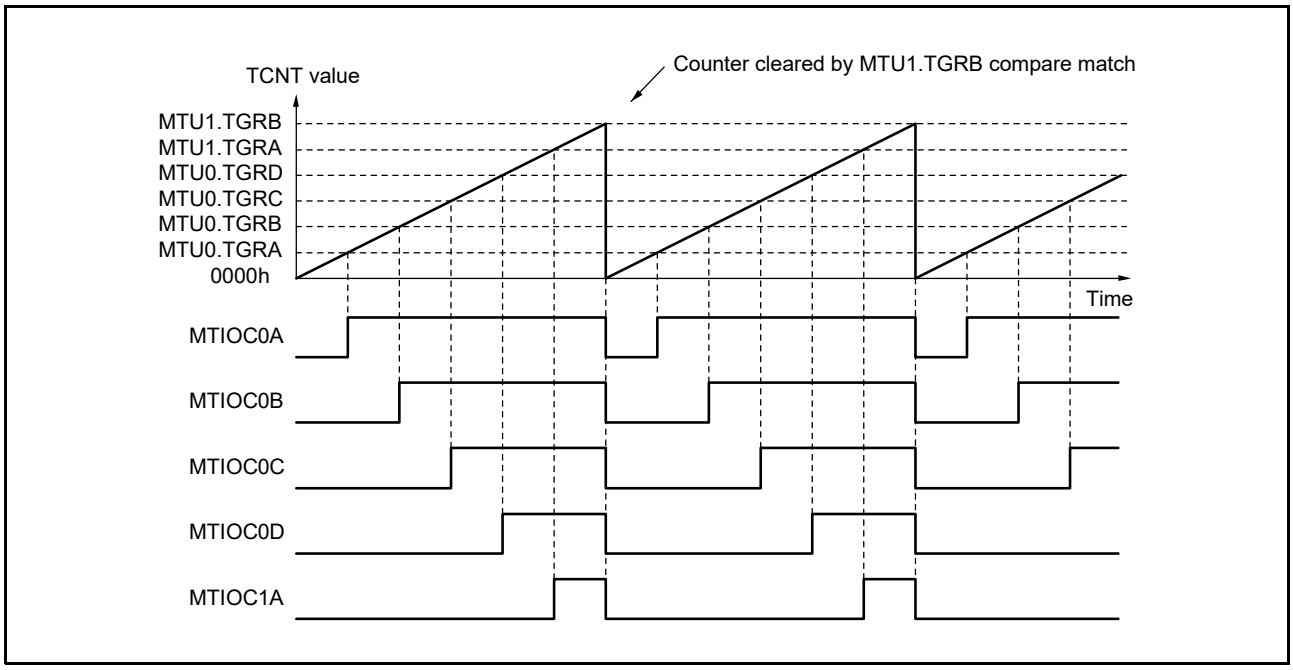


Figure 22.28 Example of PWM Mode 2 Operation

Figure 22.29 shows examples of PWM waveform output with 0% duty and 100% duty in PWM mode 1.
In this example, TGRA compare match is set as the TCNT clearing source, a low level is set as the initial output value and output value for TGRA, and a high level is set as the output value for TGRB.

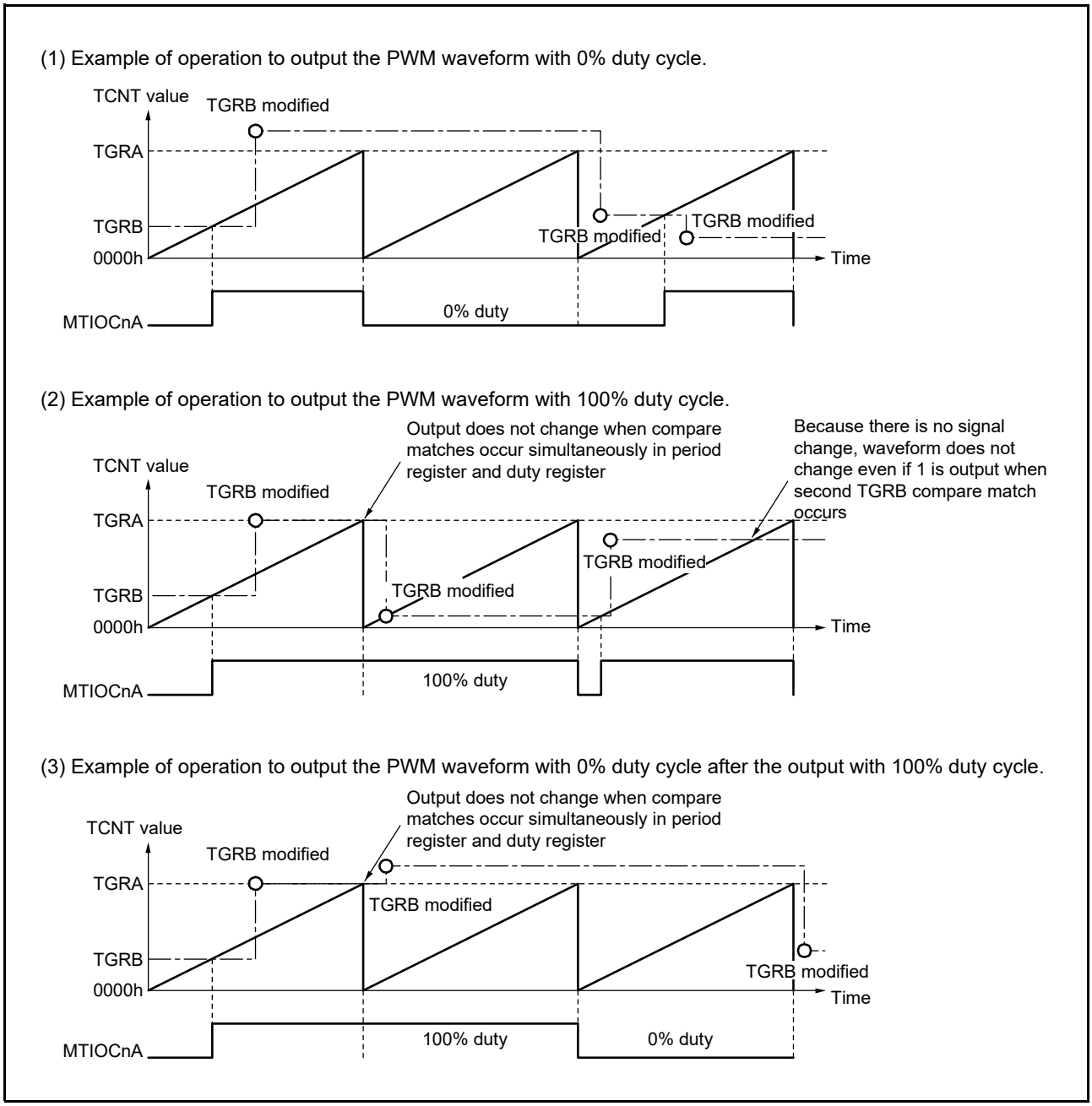


Figure 22.29 Examples of PWM Mode Operation (PWM Waveform Output with 0% Duty and 100% Duty) (n = 0 to 4, 6, 7, 9)

22.3.6 Phase Counting Mode

There are two phase counting modes: 16-bit phase counting mode in which MTU1 and MTU2 operate independently, and cascade connection 32-bit phase counting mode in which MTU1 and MTU2 are cascaded.
In phase counting mode, the phase difference between two external input clocks is detected and the corresponding TCNT is incremented or decremented.
Two external clock input pins for each phase counting mode are not affected by the settings of TCR.TPSC[2:0], TCR2.TPSC2[2:0], and CKEG[1:0]. The two external clock input pins used in 16-bit phase counting mode and cascade connection 32-bit phase counting mode of MTU2 can be selected by MTU1.TMDR3.PHCKSEL. In a phase counting mode other than 16-bit phase counting mode and cascade connection 32-bit phase counting mode of MTU2, MTCLKA and MTCLKB are selected for A-phase and B-phase, respectively. In phase counting mode, the external clock pins MTCLKA, MTCLKB, MTCLKC, and MTCLKD are used for two-phase encoder pulse input.
Table 22.66 lists the external clock input pins to be connected in each phase counting mode.

Table 22.66 Clock Input Pins in Phase Counting Mode

		External Clock Input Pins	
Phase Counting Mode	**TMDR3.PHCKSEL bit**	**A-Phase**	**B-Phase**
MTU1 16-bit phase counting mode	x (Don't care)	MTCLKA	MTCLKB
MTU2 16-bit phase counting mode	0	MTCLKA	MTCLKB
	1 (initial value)	MTCLKC	MTCLKD
Cascade connection 32-bit phase counting mode	0	MTCLKA	MTCLKB
	1 (initial value)	MTCLKC	MTCLKD

22.3.6.1 16-Bit Phase Counting Mode

When the MTU1.TMDR3.LWA is 0, 16-bit phase counting mode can be set individually for MTU1 and MTU2.
In 16-bit phase counting mode, the phase difference between two external input clocks is detected and the 16-bit counter TCNT of the corresponding channel is incremented or decremented.
When 16-bit phase counting mode is specified, an external clock is selected as the count clock and TCNT operates as an up-counter/down-counter regardless of the setting of TCR.TPSC[2:0], TCR2.TPSC2[2:0], and CKEG[1:0]. However, the functions of TCR.CCLR[1:0],TIOR, TIER, and TGR are enabled, and input capture/compare match and interrupt functions can be used.
These external input pins can be used for two-phase encoder pulse input.
When an overflow occurs while TCNT is counting up and the corresponding TIER.TCIEV bit is 1, a TCIV interrupt is generated. When an underflow occurs while TCNT is counting down and the corresponding TIER.TCIEU bit is 1, a TCIU interrupt is generated.
The TSR.TCFD flag is the count direction flag. Read the TCFD flag to check whether TCNT is counting up and down.

(1) Example of 16-Bit Phase Counting Mode Setting Procedure

Figure 22.30 shows an example of the phase counting mode setting procedure.

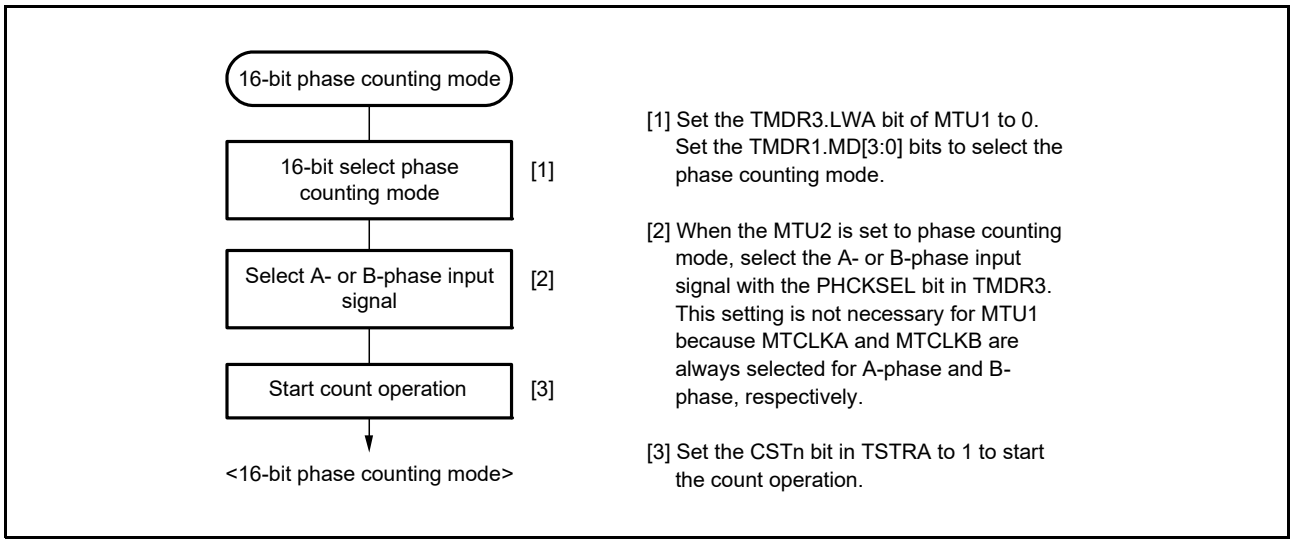


Figure 22.30 Example of 16-Bit Phase Counting Mode Setting Procedure

(2) Examples of 16-Bit Phase Counting Mode Operation

In phase counting mode, TCNT is incremented or decremented according to the phase difference between two external clocks. There are five modes according to the count conditions. Each mode operates under the condition PHCKSEL = 1, which means the phase clock for MTU1 is input from MTCLKA or MTCLKB and that for MTU2 is input from MTCLKC or MTCLKD.

(a) Phase Counting Mode 1

Figure 22.31 shows an example of operation in phase counting mode 1, and Table 22.67 summarizes the TCNT up-counting and down-counting conditions.

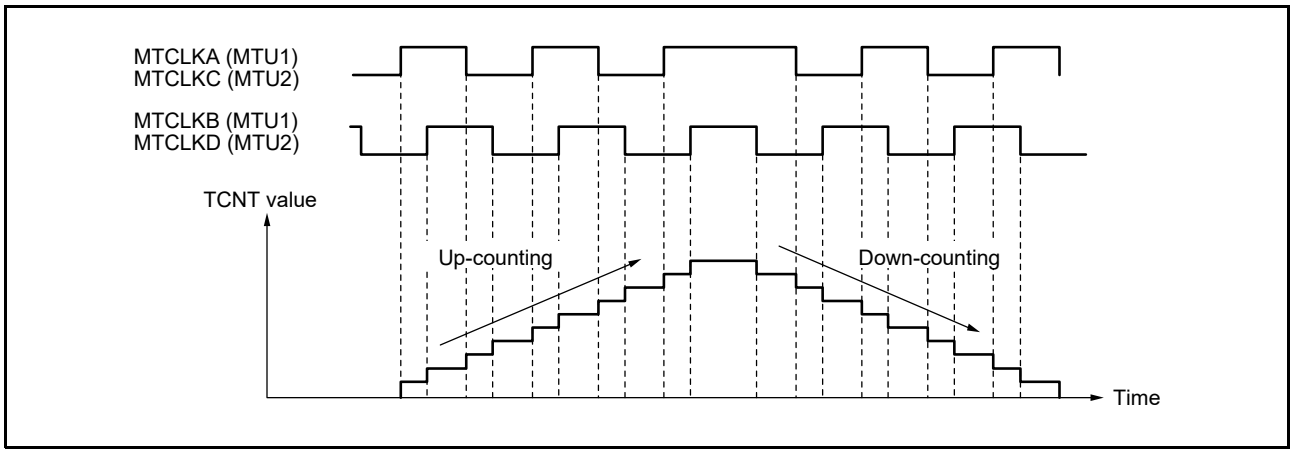


Figure 22.31 Example of Operation in Phase Counting Mode 1

Table 22.67 Up-Counting and Down-Counting Conditions in Phase Counting Mode 1

MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
High	↑	Up-counting
Low	↓	
↑	Low	
↓	High	
High	↓	Down-counting
Low	↑	
↑	High	
↓	Low	

↑ : Rising edge

↓ : Falling edge

(b) Phase Counting Mode 2

Figure 22.32 to Figure 22.34 show the examples of operation in phase counting mode 2 and Table 22.68 summarizes the TCNT up-counting and down-counting conditions.

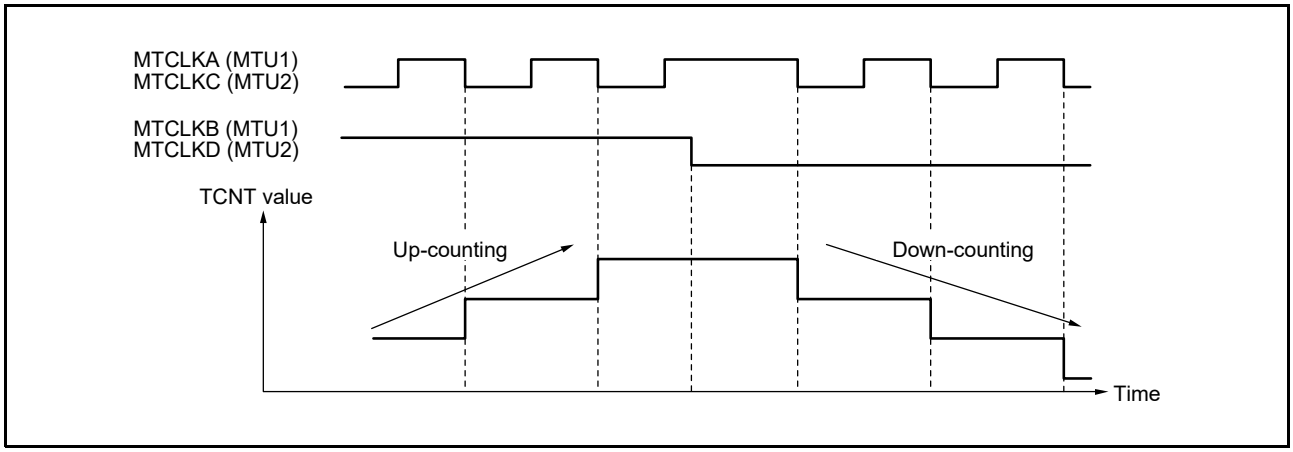


Figure 22.32 Example of Operation in Phase Counting Mode 2 (When MTUn.TCR2.PCB[1:0] = 00b (n = 1, 2))

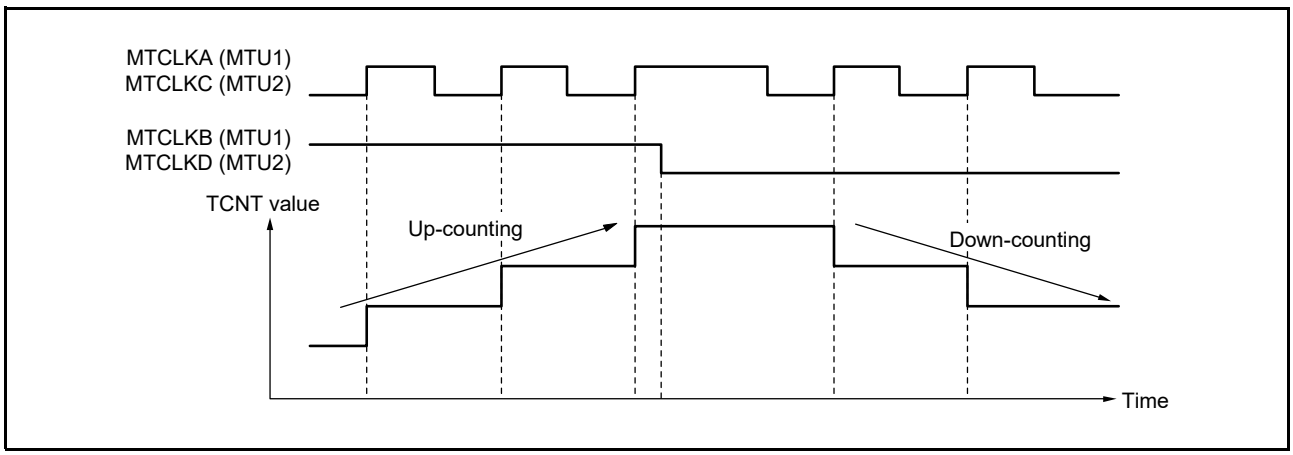


Figure 22.33 Example of Operation in Phase Counting Mode 2 (When MTUn.TCR2.PCB[1:0] = 01b (n = 1, 2))

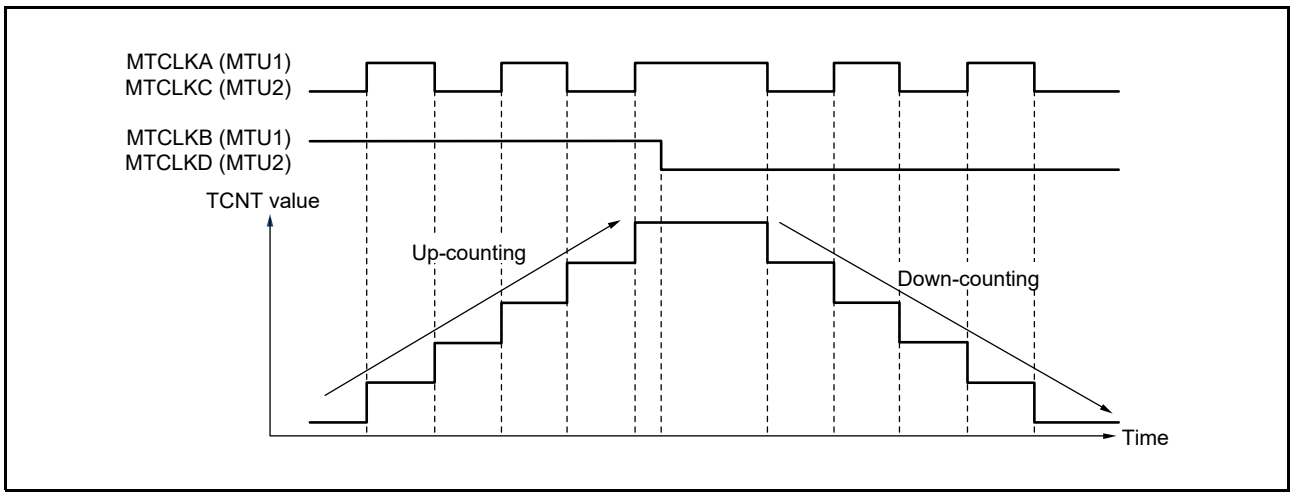


Figure 22.34 Example of Operation in Phase Counting Mode 2 (When MTUn.TCR2.PCB[1:0] = 1xb (n = 1, 2))

Table 22.68 Up-Counting and Down-Counting Conditions in Phase Counting Mode 2

PCB[1:0]	MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
00b	High	↑	Not counted (Don't care)
	Low	↓	
	↑	Low	
	↓	High	Up-counting
	High	↓	Not counted (Don't care)
	Low	↑	
	↑	High	
	↓	Low	Down-counting
01b	High	↑	Not counted (Don't care)
	Low	↓	
	↑	Low	Down-counting
	↓	High	Not counted (Don't care)
	High	↓	
	Low	↑	
	↑	High	Up-counting
	↓	Low	Not counted (Don't care)
1xb	High	↑	Not counted (Don't care)
	Low	↓	
	↑	Low	Down-counting
	↓	High	Up-counting
	High	↓	Not counted (Don't care)
	Low	↑	
	↑	High	Up-counting
	↓	Low	Down-counting

↑ : Rising edge

↓ : Falling edge

(c) Phase Counting Mode 3

Figure 22.35 to Figure 22.37 show the examples of operation in phase counting mode 3 and Table 22.69 summarizes the TCNT up-counting and down-counting conditions.

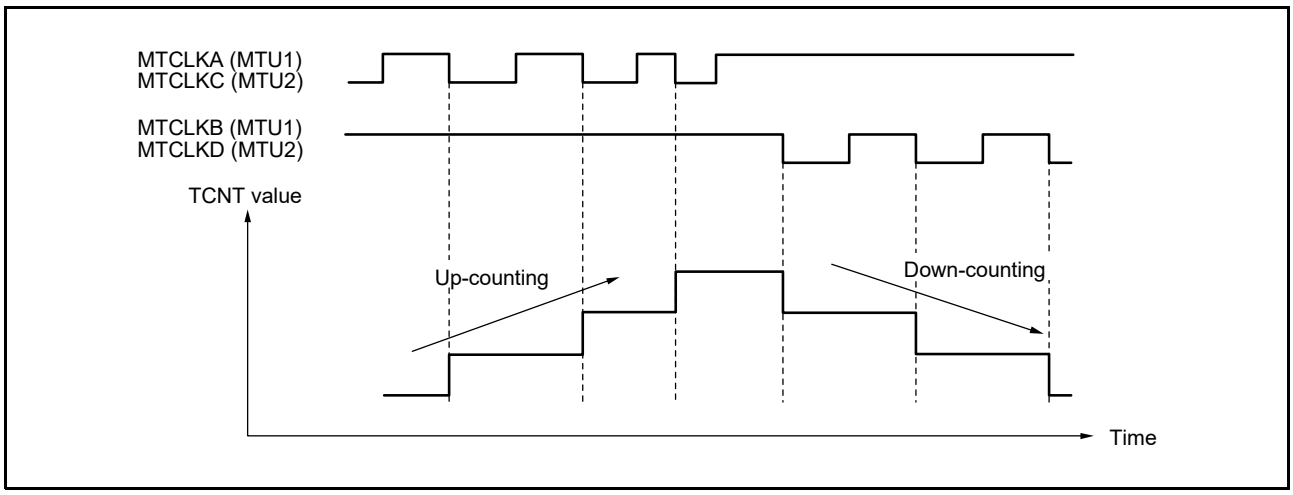


Figure 22.35 Example of Operation in Phase Counting Mode 3 (When MTUn.TCR2.PCB[1:0] = 00b (n = 1, 2))

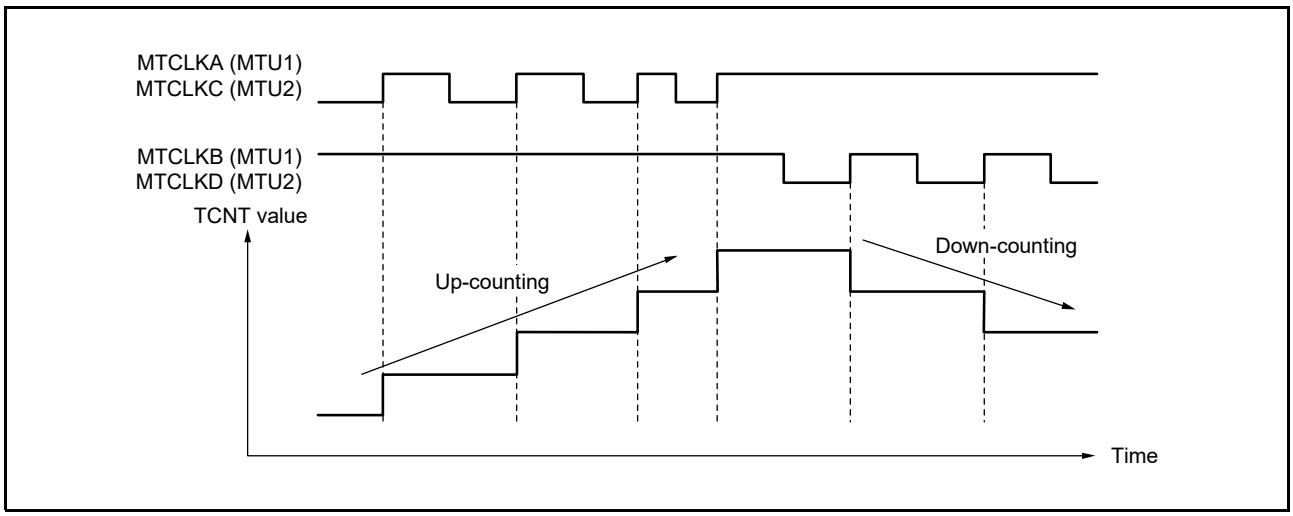


Figure 22.36 Example of Operation in Phase Counting Mode 3 (When MTUn.TCR2.PCB[1:0] = 01b (n = 1, 2))

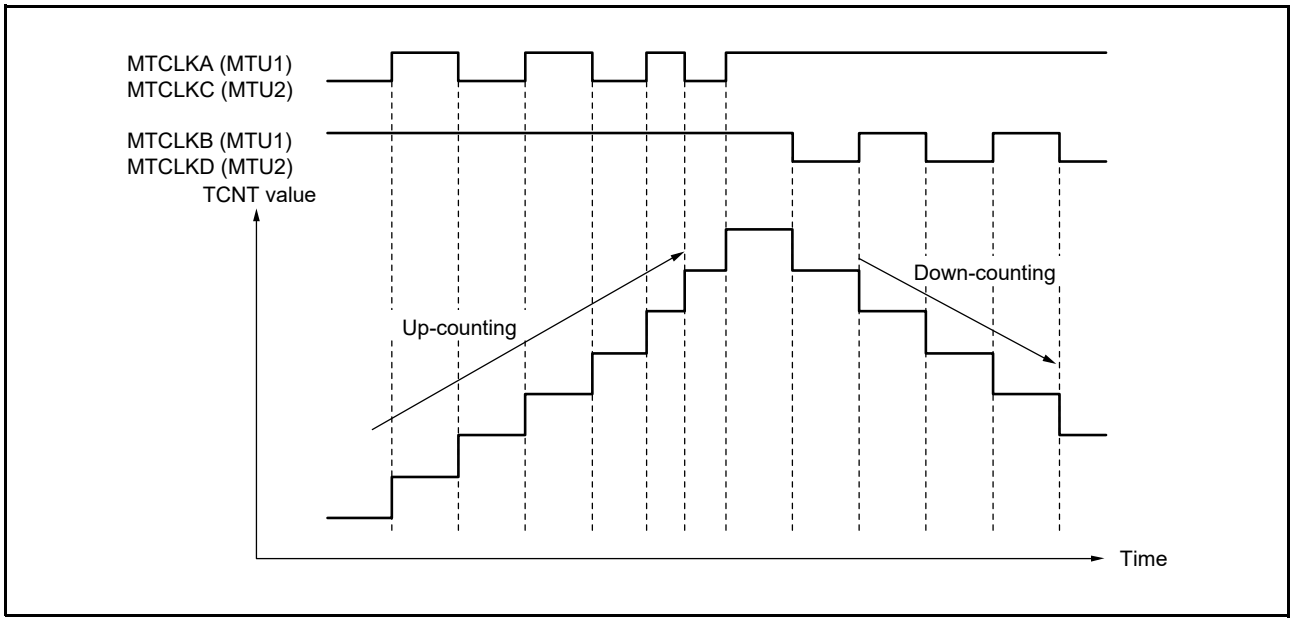


Figure 22.37 Example of Operation in Phase Counting Mode 3 (When MTUn.TCR2.PCB[1:0] = 1xb (n = 1, 2))

Table 22.69 Up-Counting and Down-Counting Conditions in Phase Counting Mode 3

PCB[1:0]	MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
00b	High	↑	Not counted (Don't care)
	Low	↓	
	↑	Low	
	↓	High	Up-counting
	High	↓	Down-counting
	Low	↑	Not counted (Don't care)
	↑	High	
	↓	Low	
01b	High	↑	Down-counting
	Low	↓	Not counted (Don't care)
	↑	Low	
	↓	High	
	High	↓	
	Low	↑	
	↑	High	Up-counting
	↓	Low	Not counted (Don't care)
1xb	High	↑	Down-counting
	Low	↓	Not counted (Don't care)
	↑	Low	
	↓	High	Up-counting
	High	↓	Down-counting
	Low	↑	Not counted (Don't care)
	↑	High	Up-counting
	↓	Low	Not counted (Don't care)

↑ : Rising edge

↓ : Falling edge

(d) Phase Counting Mode 4

Figure 22.38 shows an example of operation in phase counting mode 4, and Table 22.70 summarizes the TCNT up-counting and down-counting conditions.

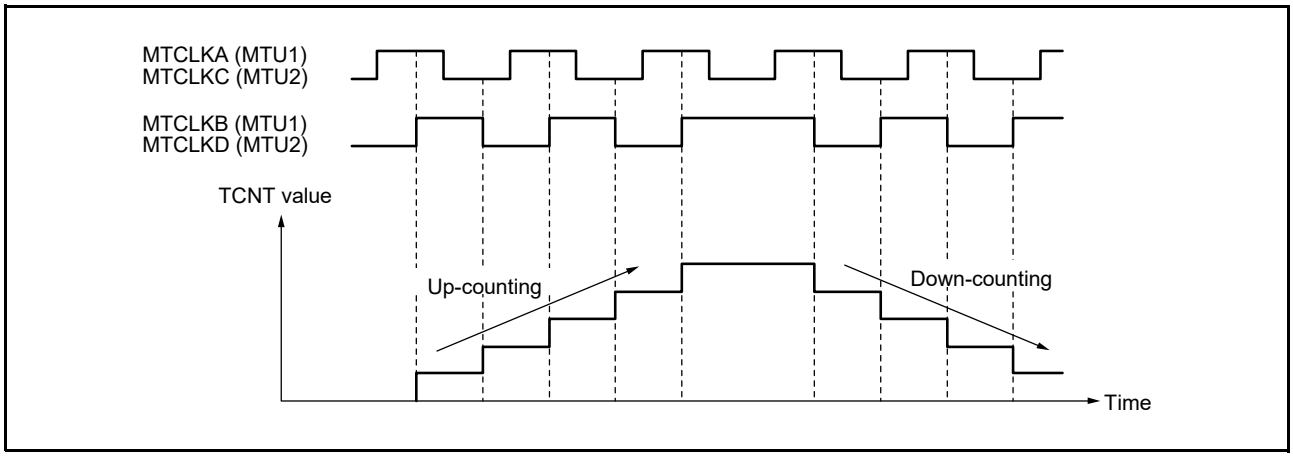


Figure 22.38 Example of Operation in Phase Counting Mode 4

Table 22.70 Up-Counting and Down-Counting Conditions in Phase Counting Mode 4

MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
High	↑	Up-counting
Low	↓	
↑	Low	Not counted (Don't care)
↓	High	
High	↓	Down-counting
Low	↑	
↑	High	Not counted (Don't care)
↓	Low	

↑ : Rising edge

↓ : Falling edge

(e) Phase Counting Mode 5

Figure 22.39 and Figure 22.40 show the examples of operation in phase counting mode 5 and Table 22.71 summarizes the TCNT up-counting and down-counting conditions.

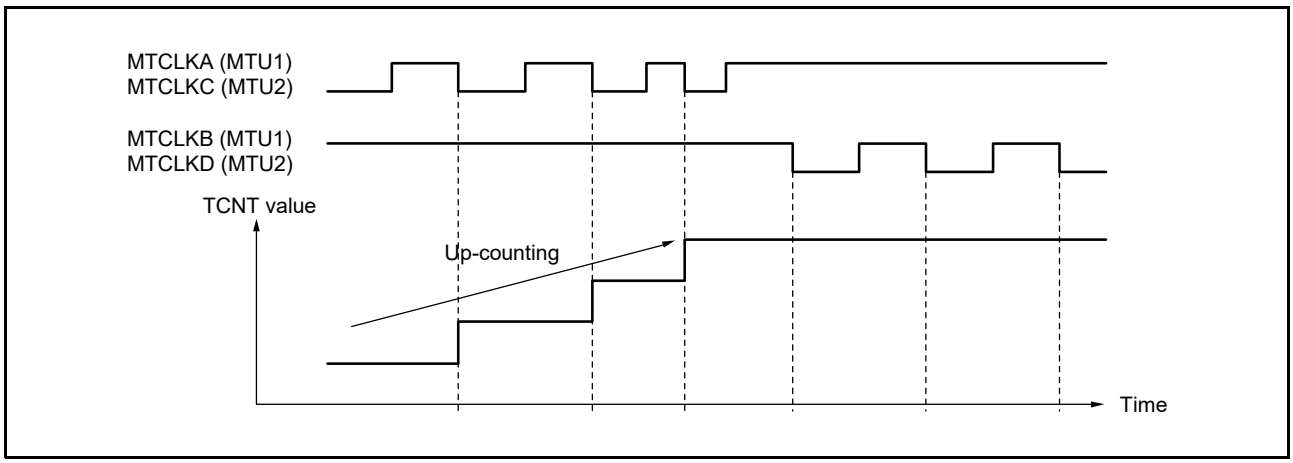


Figure 22.39 Example of Operation in Phase Counting Mode 5 (When MTUn.TCR2.PCB[1:0] = 0xb (n = 1, 2))

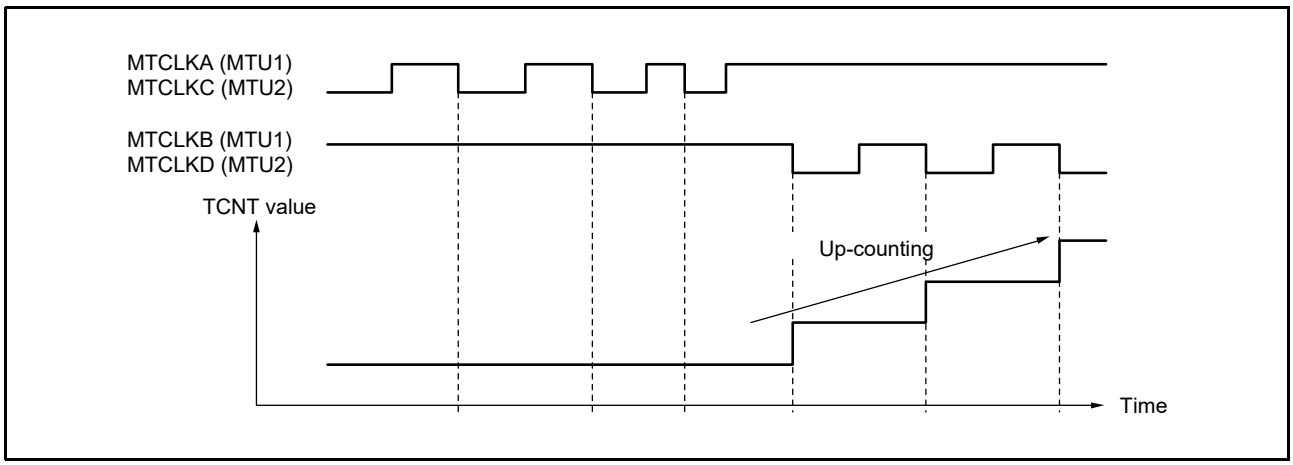


Figure 22.40 Example of Operation in Phase Counting Mode 5 (When MTUn.TCR2.PCB[1:0] = 1xb (n = 1, 2))

Table 22.71 Up-Counting and Down-Counting Conditions in Phase Counting Mode 5

PCB[1:0]	MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
0xb	High	↑	Not counted (Don't care)
	Low	↓	
	↑	Low	
	↓	High	Up-counting
	High	↓	Not counted (Don't care)
	Low	↑	
	↑	High	
	↓	Low	Up-counting
1xb	High	↑	Not counted (Don't care)
	Low	↓	Up-counting
	↑	Low	Not counted (Don't care)
	↓	High	
	High	↓	Up-counting
	Low	↑	Not counted (Don't care)
	↑	High	
	↓	Low	

↑ : Rising edge

↓ : Falling edge

(3) 16-Bit Phase Counting Mode Application Example

Figure 22.41 shows an example in which MTU1 is in phase counting mode, and MTU1 is coupled with MTU0 to input 2-phase encoder pulses of a servo motor in order to detect position or speed.
MTU1 is set to phase counting mode 1, and the encoder pulse A-phase and B-phase are input to MTCLKA and MTCLKB.
In MTU0, MTU0.TGRC compare match is specified as the TCNT clearing source and MTU0.TGRA and MTU0.TGRC are used for the compare match function and are set with the speed control period and position control period.
MTU0.TGRB is used for input capture, with MTU0.TGRB and MTU0.TGRD operating in buffer mode. The MTU1 count clock is designated as the MTU0.TGRB input capture source, and the widths of 2-phase encoder 4-multiplication pulses are detected.
MTU1.TGRA and MTU1.TGRB for MTU1 are designated for the input capture function and MTU0.TGRA and MTU0.TGRC compare matches in MTU0 are selected as the input capture sources to store the up/down-counter values for the control periods.
This procedure enables the accurate detection of position and speed.

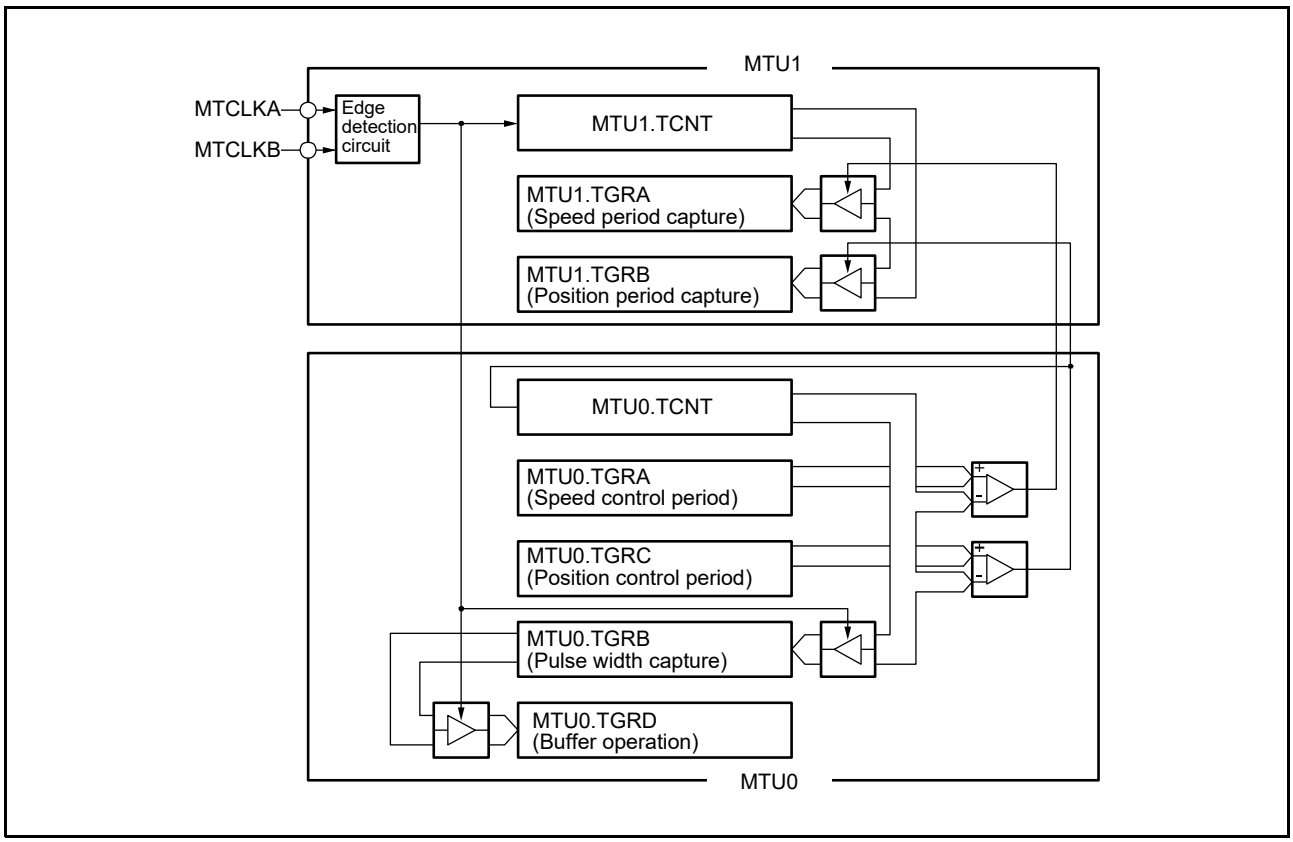


Figure 22.41 16-Bit Phase Counting Mode Application Example

22.3.6.2 Cascade Connection 32-Bit Phase Counting Mode

When MTU1 is set to phase counting mode by setting MTU1.TMDR3.LWA = 1, MTU1 and MTU2 are connected to operate in cascade connection 32-bit phase counting mode. When this mode is used, the TCR, TCR2, TIOR, TIER, TGR, and TSR registers are controlled by MTU1 and the settings of MTU2 are disabled. Refer to Figure 22.42 for the procedure for setting cascade connection 32-bit phase counting mode.
Refer to section 22.3.4, Cascaded Operation, for details on the cascade connection function for connecting MTU1 and MTU2 in a mode other than cascade connection 32-bit phase counting mode.

(1) Example of Setting Cascade Connection 32-Bit Phase Counting Mode

Figure 22.42 shows an example of the procedure for setting cascade connection 32-bit phase counting mode.

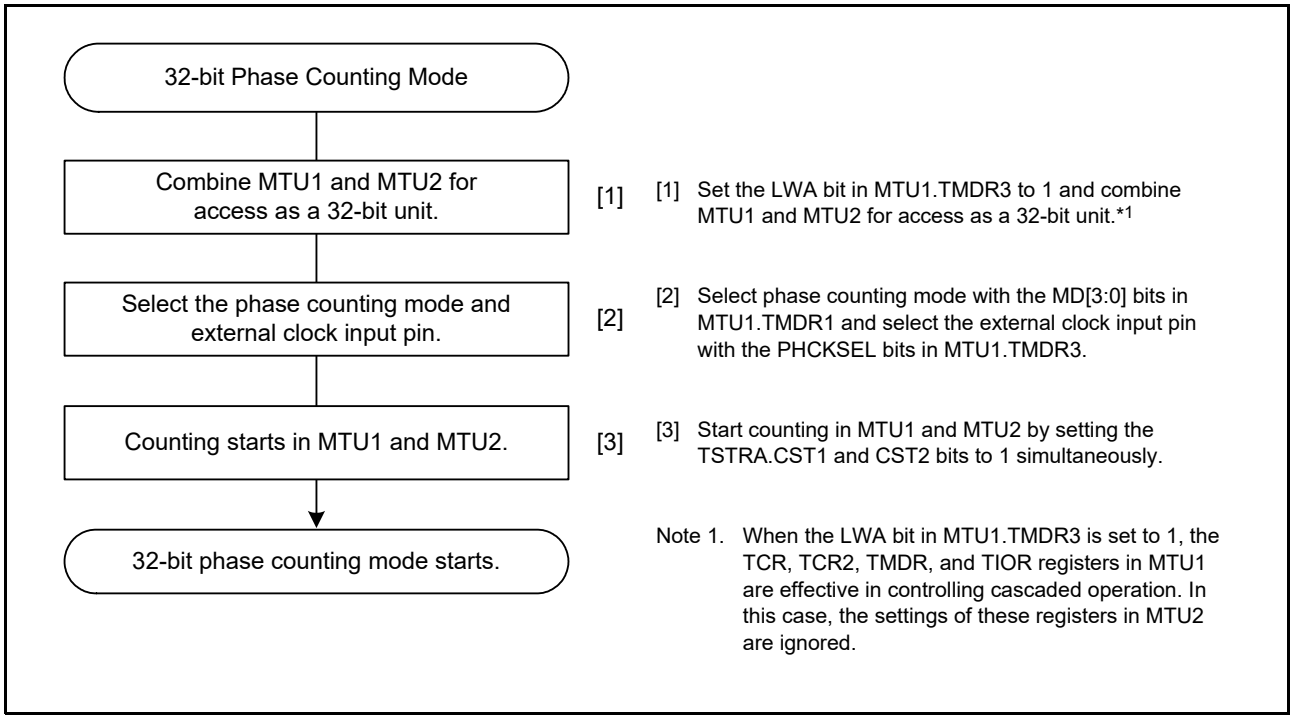


Figure 22.42 Procedure for Setting Cascade Connection 32-Bit Phase Counting Mode

22.3.7 Reset-Synchronized PWM Mode

In the reset-synchronized PWM mode, six phases of positive and negative PWM waveforms (12 phases in total) that share a common wave transition point can be output by combining MTU3 and MTU4 and MTU6 and MTU7.
When set for reset-synchronized PWM mode, the MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D, MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, and MTIOC7D pins function as PWM output pins and timer counters 3 and 6 (MTU3.TCNT and MTU6.TCNT) functions as an up-counter.
Table 22.72 shows the PWM output pins used. Table 22.73 shows the settings of the registers.

Table 22.72 Output Pins for Reset-Synchronized PWM Mode

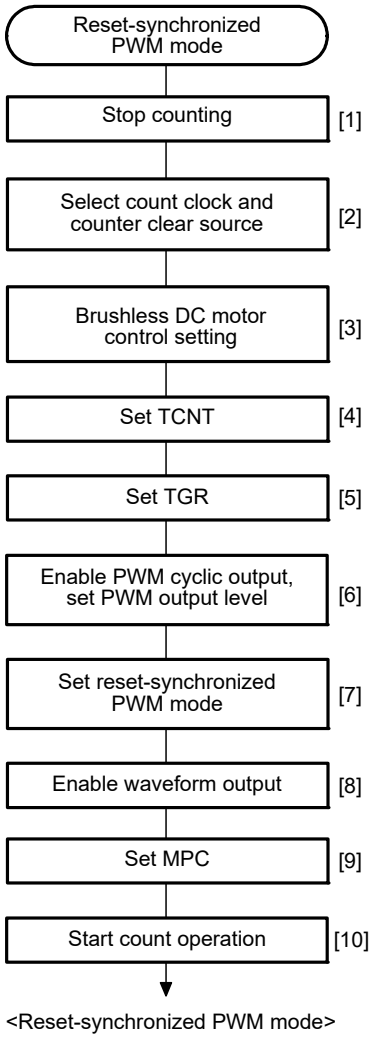
Channel	Output Pin	Description
MTU3	MTIOC3A	Toggle output synchronized with PWM period (or I/O port)
	MTIOC3B	PWM output pin 1
	MTIOC3D	PWM output pin 1' (negative-phase waveform of PWM output 1)
MTU4	MTIOC4A	PWM output pin 2
	MTIOC4C	PWM output pin 2' (negative-phase waveform of PWM output 2)
	MTIOC4B	PWM output pin 3
	MTIOC4D	PWM output pin 3' (negative-phase waveform of PWM output 3)
MTU6	MTIOC6A	Toggle output synchronized with PWM period (or I/O port)
	MTIOC6B	PWM output pin 4
	MTIOC6D	PWM output pin 4' (negative-phase waveform of PWM output 4)
MTU7	MTIOC7A	PWM output pin 5
	MTIOC7C	PWM output pin 5' (negative-phase waveform of PWM output 5)
	MTIOC7B	PWM output pin 6
	MTIOC7D	PWM output pin 6' (negative-phase waveform of PWM output 6)

Table 22.73 Register Settings for Reset-Synchronized PWM Mode

Register	Setting
MTU3.TCNT	Initial setting (0000h)
MTU4.TCNT	Initial setting (0000h)
MTU3.TGRA	Set the count period for MTU3.TCNT
MTU3.TGRB	Set the transition point of the PWM waveform to be output from the MTIOC3B and MTIOC3D pins
MTU4.TGRA	Set the transition point of the PWM waveform to be output from the MTIOC4A and MTIOC4C pins
MTU4.TGRB	Set the transition point of the PWM waveform to be output from the MTIOC4B and MTIOC4D pins
MTU6.TCNT	Initial setting (0000h)
MTU7.TCNT	Initial setting (0000h)
MTU6.TGRA	Set the count period for MTU6.TCNT
MTU6.TGRB	Set the transition point of the PWM waveform to be output from the MTIOC6B and MTIOC6D pins
MTU7.TGRA	Set the transition point of the PWM waveform to be output from the MTIOC7A and MTIOC7C pins
MTU7.TGRB	Set the transition point of the PWM waveform to be output from the MTIOC7B and MTIOC7D pins

(1) Example of Procedure for Setting Reset-Synchronized PWM Mode

Figure 22.43 shows an example of procedure for setting the reset-synchronized PWM mode.



[1] Set the TSTRA.CST3 (TSTRB.CST6) and TSTRA.CST4 (TSTRB.CST7) bits to 0 to stop the TCNT count operation. Specify the reset-synchronized PWM mode while MTU3.TCNT (MTU6.TCNT) and MTU4.TCNT (MTU7.TCNT) are stopped.

[2] Set the TCR.TPSC[2:0] and TCR.CKEG[1:0] bits of MTU3 (MTU6) to select the count clock source and clock edge for MTU3 (MTU6). Set the TCR.CCLR[2:0] bits to select TGRA compare match as the counter clear source.

[3] To control a brushless DC motor, set the TGCRA.BDC (TGCRB.BDC) bit and set the feedback signal input source and output chopping or gate signal direct output.

[4] Set MTU3.TCNT (MTU6.TCNT) and MTU4.TCNT (MTU7.TCNT) to 0000h.

[5] MTU3.TGRA (MTU6.TGRA) is the period register. Set the PWM period in MTU3.TGRA (MTU6.TGRA).
Set the PWM output transition timing in MTU3.TGRB (MTU6.TGRB), MTU4.TGRA (MTU7.TGRA), and MTU4.TGRB (MTU7.TGRB).
Note that the setting values should be within the range of compare match with MTU3.TCNT (MTU6.TCNT).

X ≤ MTU3.TGRA (X: Set value)

[6] Set the PSYE bit in the TOCR1A (TOCR1B) register to enable or disable toggle output synchronized with the PWM period, and set the OLSP and OLSN bits to select the PWM output level.
When specifying the PWM output level using the TOCR2A (TOCR2B) and TOLBRA (TOLBRB) in buffer operation, refer to Timer Output Level Buffer Registers (TOLBRA and TOLBRB).

[7] Set the MD[3:0] bits in MTU3.TMDR1 (MTU6.TMDR1) register to 1000b to select the reset-synchronized PWM mode. Specify the buffer operation (BFA = 1) of the MTU3.TGRA (MTU6.TGRA) register and the buffer operation (BFB = 1) of the MTU3.TGRB (MTU6.TGRB) register when necessary. Set the MTU4.TMDR1 (MTU7.TMDR1) register to the initial value (normal operation; BFA = 0 and BFB = 0).

[8] Set the TOERA (TOERB) register to enable or disable output from the PWM output pin. Then, set the MTU3.TIORH (MTU6.TIORH), MTU3.TIORL (MTU6.TIORL), MTU4.TIORH (MTU7.TIORH), and MTU4.TIORL (MTU7.TIORL) registers to 00h.

[9] Set the MPU and the port mode register (PMR) of the I/O ports.

[10] Set the TSTRA.CST3 (TSTRB.CST6) bit to 1 to start count operation.

Note: The output waveform starts to toggle at the point of MTU3.TCNT (MTU6.TCNT) = MTU3.TGRA (MTU6.TGRA) = X by setting X = MTU3.TGRA (MTU6.TGRA), i.e., period = duty ratio.

Figure 22.43 Procedure for Selecting Reset-Synchronized PWM Mode

(2) Example of Reset-Synchronized PWM Mode Operation

Figure 22.44 shows an example of operation in the reset-synchronized PWM mode.
MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT) operate as up-counters. The counters are cleared when a compare match occurs between MTU3.TCNT (MTU6.TCNT) and MTU3.TGRA (MTU6.TGRA), and then begin incrementing from 0000h. The output from the PWM pins toggles every time a compare match occurs in MTU3.TGRB (MTU6.TGRB), MTU4.TGRA (MTU7.TGRA), and MTU4.TGRB (MTU7.TGRB) and the counters are cleared.

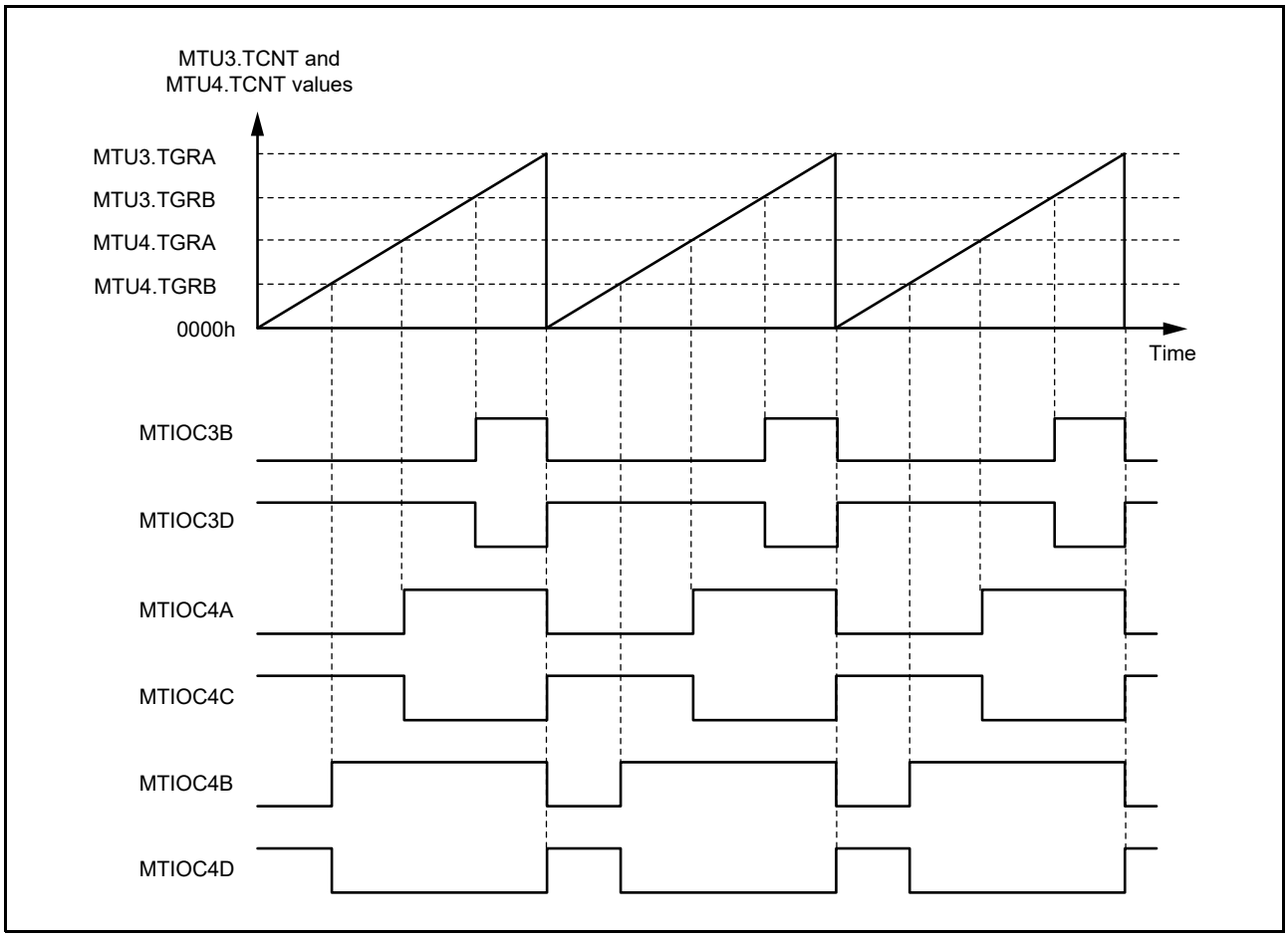


Figure 22.44 Example of Reset-Synchronized PWM Mode Operation (When OLSN = 1 and OLSP = 1 in MTU3.TOCR1 and MTU4.TOCR1)

22.3.8 Complementary PWM Mode

In complementary PWM mode, dead time can be set for PWM waveforms to be output. The dead time is the period during which the upper and lower arm transistors are set to the inactive level in order to prevent short-circuiting of the arms.

Six positive-phase and six negative-phase PWM waveforms (12 phases in total) with dead time can be output by combining MTU3/ MTU4 and MTU6/MTU7. PWM waveforms without dead time can also be output.

In complementary PWM mode, MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, MTIOC4D, MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7B, MTIOC7C, and MTIOC7D pins function as PWM output pins, and the MTIOC3A and MTIOC6A pins can be set for toggle output synchronized with the PWM period.

MTU3.TCNT, MTU4.TCNT, MTU6.TCNT, and MTU7.TCNT function as up/down-counters.

Table 22.74 shows the PWM output pins used. Table 22.75 shows the settings of the registers used.

A function to directly cut off the PWM output by using an external signal is supported as a port function.

Table 22.74 Output Pins for Complementary PWM Mode

Channel	Output Pin	Description
MTU3	MTIOC3A	Toggle output synchronized with PWM period (or I/O port)
	MTIOC3B	PWM output pin 1
	MTIOC3C	I/O port*1
	MTIOC3D	PWM output pin 1' (negative-phase waveform output of PWM output 1)
MTU4	MTIOC4A	PWM output pin 2
	MTIOC4C	PWM output pin 2' (negative-phase waveform output of PWM output 1)
	MTIOC4B	PWM output pin 3
	MTIOC4D	PWM output pin 3' (negative-phase waveform output of PWM output 1)
MTU6	MTIOC6A	Toggle output synchronized with PWM period (or I/O port)
	MTIOC6B	PWM output pin 4
	MTIOC6C	I/O port*1
	MTIOC6D	PWM output pin 4' (negative-phase waveform output of PWM output 4)
MTU7	MTIOC7A	PWM output pin 5
	MTIOC7C	PWM output pin 5' (negative-phase waveform output of PWM output 5)
	MTIOC7B	PWM output pin 6
	MTIOC7D	PWM output pin 6' (negative-phase waveform output of PWM output 6)

Note 1. Avoid setting the MTIOC3C and MTIOC6C pins as timer I/O pins in complementary PWM mode.

Table 22.75 Register Settings for Complementary PWM Mode (1/2)

Channel	Counter/ Register	Description	Read/Write from CPU
MTU3	TCNT	Starts up-counting from the value set in the dead time register	Maskable by TRWERA setting*1
	TGRA	Set MTU3.TCNT upper limit value (1/2 carrier period + dead time)	Maskable by TRWERA setting*1
	TGRB	PWM output 1 compare register	Maskable by TRWERA setting*1
	TGRC	MTU3.TGRA buffer register	Readable/writable
	TGRD	PWM output 1/MTU3.TGRB buffer register	Readable/writable
	TGRE	MTU3.TGRB buffer register B (when double buffer function is used)	Readable/writable
MTU4	TCNT	Starts up-counting after being initialized to 0000h	Maskable by TRWERA setting*1
	TGRA	PWM output 2 compare register	Maskable by TRWERA setting*1
	TGRB	PWM output 3 compare register	Maskable by TRWERA setting*1
	TGRC	PWM output 2/MTU4.TGRA buffer register	Readable/writable
	TGRD	PWM output 3/MTU4.TGRB buffer register	Readable/writable
	TGRE	MTU4.TGRA buffer register B (when double buffer function is used)	Readable/writable
	TGRF	MTU4.TGRB buffer register B (when double buffer function is used)	Readable/writable
MTU6	TCNT	Starts up-counting from the value set in the dead time register	Maskable by TRWERB setting*2
	TGRA	Set MTU6.TCNT upper limit value (1/2 carrier period + dead time)	Maskable by TRWERB setting*2
	TGRB	PWM output 4 compare register	Maskable by TRWERB setting*2
	TGRC	MTU6.TGRA buffer register	Readable/writable
	TGRD	PWM output 4/MTU6.TGRB buffer register	Readable/writable
	TGRE	MTU6.TGRB buffer register B (when double buffer function is used)	Readable/writable
MTU7	TCNT	Starts up-counting after being initialized to 0000h	Maskable by TRWERB setting*2
	TGRA	PWM output 5 compare register	Maskable by TRWERB setting*2
	TGRB	PWM output 6 compare register	Maskable by TRWERB setting*2
	TGRC	PWM output 5/MTU7.TGRA buffer register	Readable/writable
	TGRD	PWM output 6/MTU7.TGRB buffer register	Readable/writable
	TGRE	MTU7.TGRA buffer register B (when double buffer function is used)	Readable/writable
	TGRF	MTU7.TGRB buffer register B (when double buffer function is used)	Readable/writable

Note 1. Access can be enabled or disabled according to the setting in TRWERA (timer read/write enable register A).
Note 2. Access can be enabled or disabled according to the setting in TRWERB (timer read/write enable register B).

Table 22.76 Register Settings for Complementary PWM Mode (2/2)

Channel	Counter/ Register	Description	Read/Write from CPU
Timer dead time data register A (TDDRA)		Set MTU4.TCNT and MTU3.TCNT offset value (dead time value)	Maskable by TRWERA setting*1
Timer dead time data register B (TDDRB)		Set MTU7.TCNT and MTU6.TCNT offset value (dead time value)	Maskable by TRWERB setting *2
Timer period data register A (TCDRA)		Set MTU4.TCNT upper limit value (1/2 carrier period)	Maskable by TRWERA setting*1
Timer period data register B (TCDRB)		Set MTU7.TCNT upper limit value (1/2 carrier period)	Maskable by TRWERB setting*2
Timer period buffer register A (TCBRA)		TCDRA buffer register	Readable/writable
Timer period buffer register B (TCBRB)		TCDRB buffer register	Readable/writable
Subcounter A (TCNTSA)		Subcounter A for dead time generation	Read-only
Subcounter B (TCNTSB)		Subcounter B for dead time generation	Read-only
Temporary register 1A (TEMP1A)		PWM output 1/MTU3.TGRB temporary register A	Not readable/writable
Temporary register 1B (TEMP1B)		PWM output 1/MTU3.TGRB temporary register B (when double buffer function is used)	Not readable/writable
Temporary register 2A (TEMP2A)		PWM output 2/MTU4.TGRA temporary register A	Not readable/writable
Temporary register 2B (TEMP2B)		PWM output 2/MTU4.TGRA temporary register B (when double buffer function is used)	Not readable/writable
Temporary register 3A (TEMP3A)		PWM output 3/MTU4.TGRB temporary register A	Not readable/writable
Temporary register 3B (TEMP3B)		PWM output 3/MTU4.TGRB temporary register B (when double buffer function is used)	Not readable/writable
Temporary register 4A (TEMP4A)		PWM output 4/MTU6.TGRB temporary register A	Not readable/writable
Temporary register 4B (TEMP4B)		PWM output 4/MTU6.TGRB temporary register B (when double buffer function is used)	Not readable/writable
Temporary register 5A (TEMP5A)		PWM output 5/MTU7.TGRA temporary register A	Not readable/writable
Temporary register 5B (TEMP5B)		PWM output 5/MTU7.TGRA temporary register B (when double buffer function is used)	Not readable/writable
Temporary register 6A (TEMP6A)		PWM output 6/MTU7.TGRB temporary register A	Not readable/writable
Temporary register 6B (TEMP6B)		PWM output 6/MTU7.TGRB temporary register B (when double buffer function is used)	Not readable/writable

Note 1. Access can be enabled or disabled according to the setting in TRWERA (timer read/write enable register A).
Note 2. Access can be enabled or disabled according to the setting in TRWERB (timer read/write enable register B).

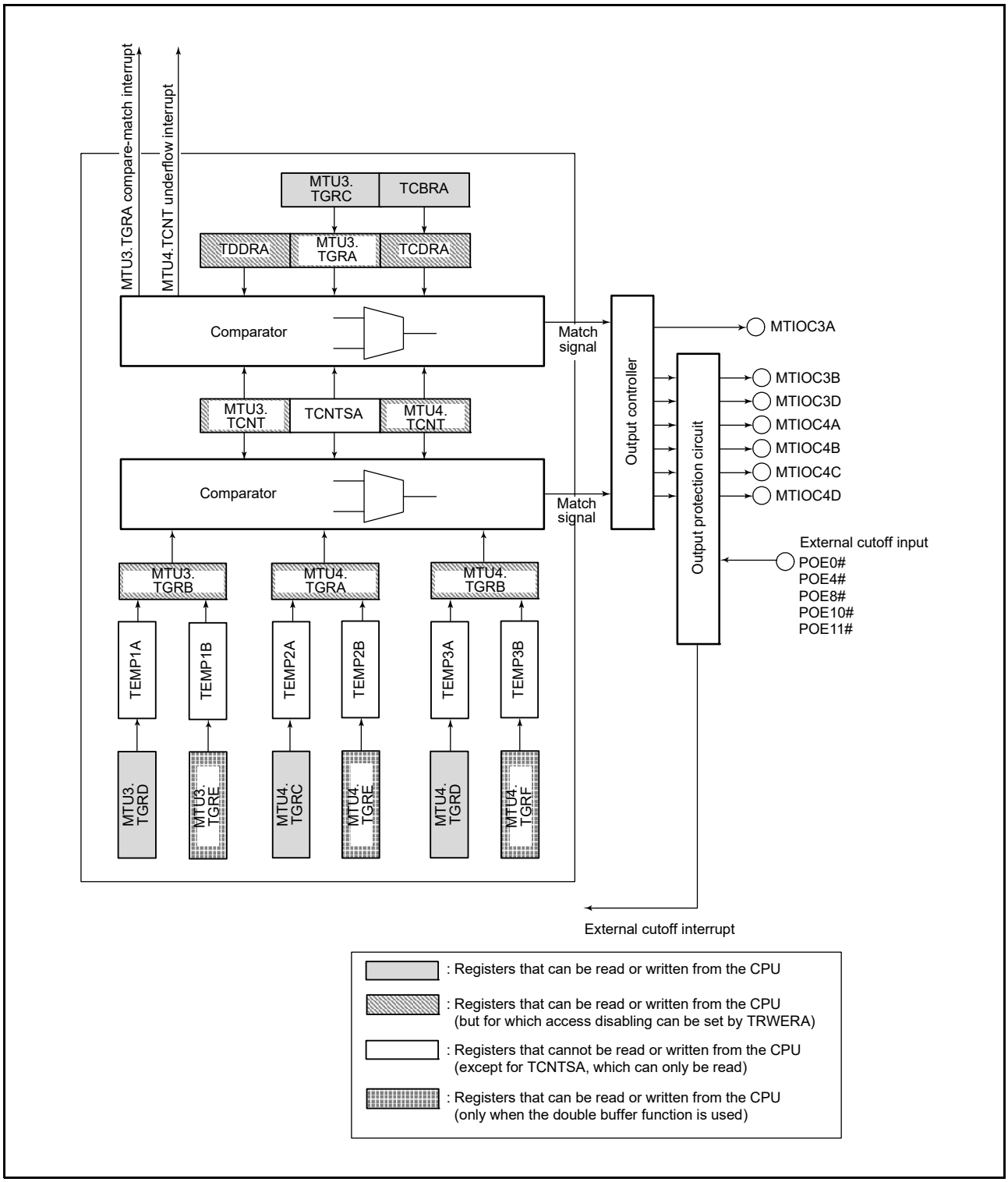


Figure 22.45 Block Diagram of MTU3 and MTU4 in Complementary PWM Mode

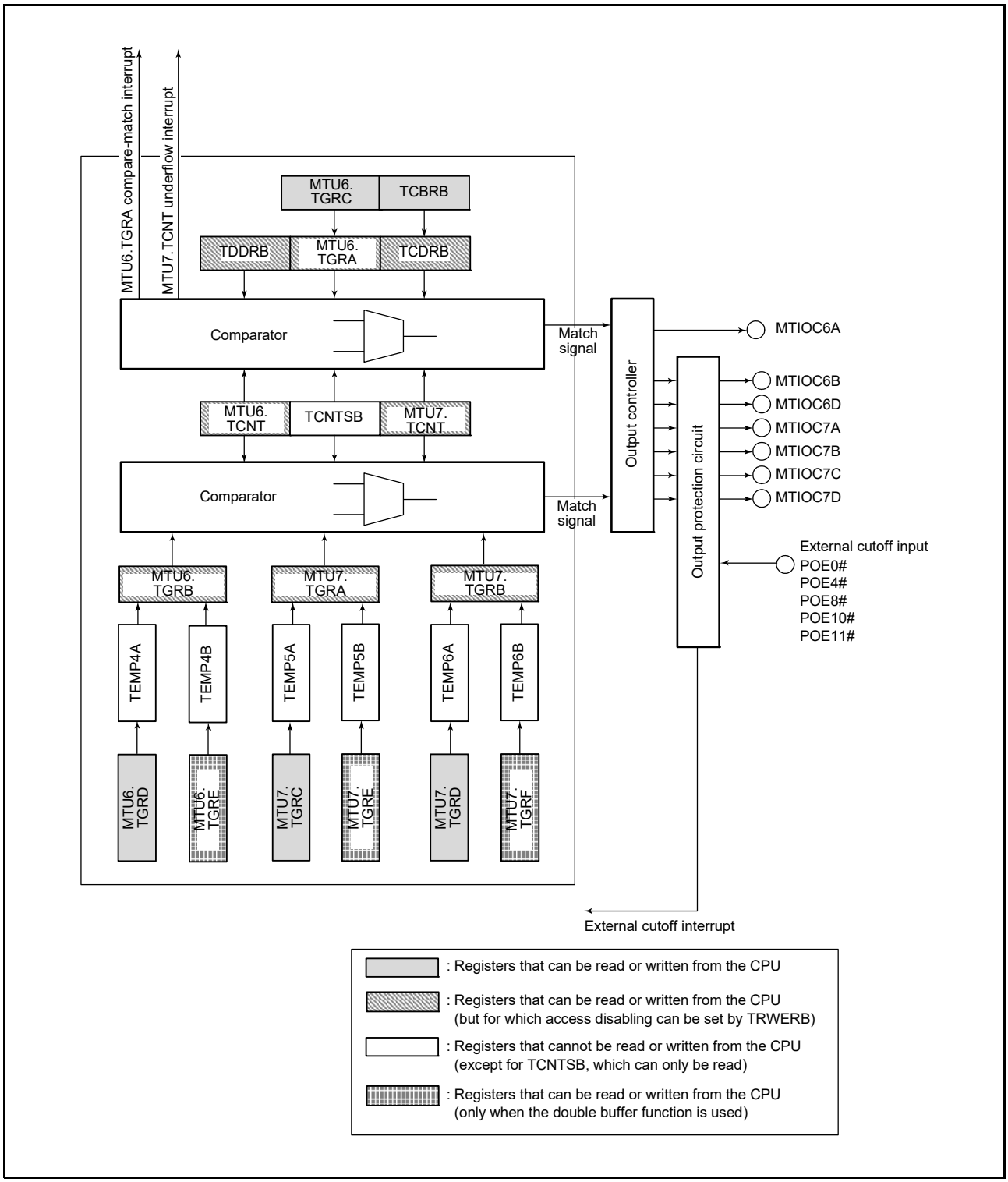
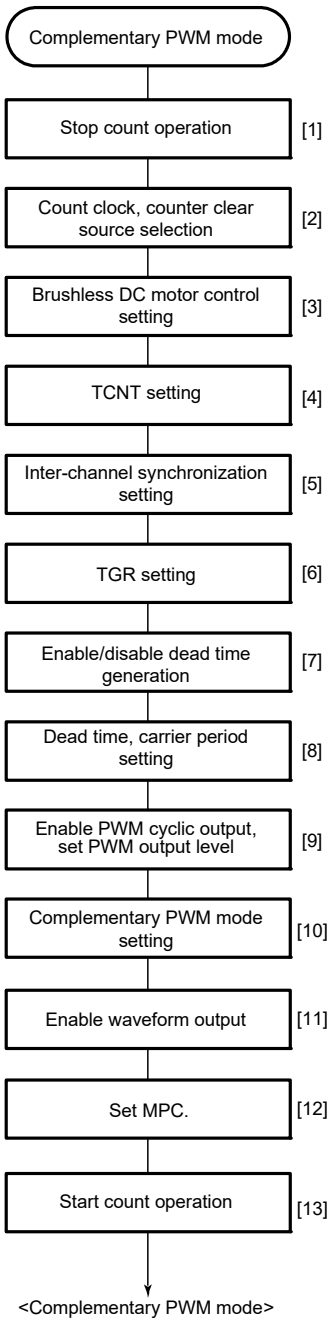


Figure 22.46 Block Diagram of MTU6 and MTU7 in Complementary PWM Mode

(1) Example of Complementary PWM Mode Setting Procedure

Figure 22.47 shows an example of the complementary PWM mode setting procedure.



[1] Set the TSTRA.CST3 (TSTRB.CST6) and TSTRA.CST4 (TSTRB.CST7) bits to 0 to stop the TCNT count operation.
Specify complementary PWM mode while MTU3.TCNT (MTU6.TCNT) and MTU4.TCNT (MTU7.TCNT) are stopped.

[2] Set the TCR.TPSC[2:0] and TCR.CKEG[1:0] bits of MTU3 and MTU4 (MTU6 and MTU7) to select the same count clock source and clock edge for MTU3 and MTU4 (MTU6 and MTU7).
Set the TCR.CCLR[2:0] bits only when restarting by synchronous clearing with another channel during complementary PWM operation.

[3] To control a brushless DC motor, set the TGCRA.BDC (TGCRB.BDC) bit and set the feedback signal input source and output chopping or gate signal direct output.

[4] Set the dead time in MTU3.TCNT (MTU6.TCNT). Set MTU4.TCNT (MTU7.TCNT) to 0000h.

[5] Make this setting only when restarting by synchronous clearing with another channel during complementary PWM mode operation. In this case, set the TSYRA register (TSYRB) to synchronize the channel generating the synchronous clear with MTU3 and MTU4 (MTU6 and MTU7).

[6] When MTU3 and MTU4 are used, set the PWM output duty ratio in the duty registers (MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB) and buffer registers (MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD).
Be sure to set the same value in each pair of the duty registers (MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD) and buffer registers (MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB).
Only when the double buffer function is used, set the buffer registers (MTU3.TGRE, MTU4.TGRE, and MTU4.TGRF).
When MTU6 and MTU7 are used, set the value to the following registers:
Duty registers (MTU6.TGRB, MTU7.TGRA, and MTU7.TGRB)
Buffer registers (MTU6.TGRD, MTU7.TGRC, and MTU7.TGRD)
Buffer registers when the double buffer function is used (MTU6.TGRE, MTU7.TGRE, and MTU7.TGRF)

[7] This setting is necessary only when no dead time should be generated. Make appropriate settings in the TDERA (TDERB) register so that no dead time generated.

[8] Set the dead time in the TDDRA (TDDRB) register, 1/2 the PWM period in the TCDRA (TCDRB) register and TCBRA (TCBRB) register, and 1/2 the PWM period plus the dead time in MTU3.TGRA (MTU6.TGRA) and MTU3.TGRC (MTU6.TGRC).
When no dead time generation is selected, set 1 in the TDDRA (TDDRB) register and 1/2 the PWM period + 1 in MTU3.TGRA (MTU6.TGRA) and MTU3.TGRC (MTU6.TGRC).

[9] Set the PSYE bit in the TOCR1A (TOCR1B) register to enable or disable toggle output synchronized with the PWM period, and set the OLSP and OLSN bits to select the PWM output level.
When specifying the PWM output level using the TOCR2A (TOCR2B) and TOLBRA (TOLBRA) in buffer operation, refer to Timer Output Level Buffer Registers (TOLBRA and TOLBRB).

[10] Select complementary PWM mode in MTU3.TMDR1 (MTU6.TMDR1). Specify the buffer operation (BFA = 1) of MTU3.TGRA (MTU6.TGRA) and the buffer operation (BFB = 1) of MTU3.TGRB (MTU6.TGRB).
Only when the double buffer function is used, set the TMDR2A (TMDR2B) register to enable the double buffer function.

[11] Set the TOERA (TOERB) register to enable or disable output from the PWM output pin. Then, set the MTU3.TIORH (MTU6.TIORH), MTU3.TIORL (MTU6.TIORL), MTU4.TIORH (MTU7.TIORH), and MTU4.TIORL (MTU7.TIORL) to 00h.

[12] Set the MPU and the port mode register (PMR) of the I/O ports.

[13] Set the TSTRA.CST3 (TSTRB.CST6) and TSTRA.CST4 (TSTRB.CST7) bits to 1 simultaneously to start count operation.

Figure 22.47 Example of Complementary PWM Mode Setting Procedure

(2) Outline of Complementary PWM Mode Operation

In complementary PWM mode, six phases (three positive and three negative) PWM waveforms can be output. Figure 22.48 illustrates counter operation in complementary PWM mode (MTU3 and MTU4), and Figure 22.49 shows an example of operation in complementary PWM mode.

(a) Counter Operation

In complementary PWM mode, three counters—MTU3.TCNT, MTU4.TCNT, and TCNTSA (MTU6.TCNT, MTU7.TCNT, and TCNTSB) —in each unit perform up-/down-count operations.
MTU3.TCNT (MTU6.TCNT) is automatically initialized to the value set in TDDRA (TDDRB) when complementary PWM mode is selected and the CST3 bit in TSTRA (TSTRB) is 0. When the CST3 bit is set to 1, MTU3.TCNT (MTU6.TCNT) counts up to the value set in MTU3.TGRA (MTU6.TGRA), then switches to down-counting when it matches MTU3.TGRA (MTU6.TGRA). When the MTU4.TCNT (MTU7.TCNT) value matches 0000h, MTU3.TCNT (MTU6.TCNT) switches to up-counting, and the operation is repeated in this way.
MTU4.TCNT (MTU7.TCNT) should be initialized to 0000h after a reset. When the CST4 bit is set to 1, MTU4.TCNT (MTU7.TCNT) counts up in synchronization with MTU3.TCNT (MTU6.TCNT), and switches to down-counting when MTU3.TCNT (MTU6.TCNT) matches MTU3.TGRA (MTU6.TGRA). On reaching 0000h, MTU4.TCNT (MTU7.TCNT) switches to up-counting, and the operation is repeated in this way. TCNTSA (TCNTSB) is a read-only counter. It does not need to be initialized after a reset.
In counting up by MTU3.TCNT and MTU4.TCNT (or MTU6.TCNT and MTU7.TCNT), MTU3.TCNT (or MTU6.TCNT) starts counting up when it matches TCDRA (or TCDRB) and switches to counting down when it matches MTU3.TGRA (or MTU6.TGRA). Furthermore, when MTU4.TCNT (or MTU7.TCNT) matches TDDRA (or TDDRB), TCNTSA (or TCNTSB) is set to the value in MTU3.TGRA (or MTU6.TGRA) and counting is stopped.
When MTU4.TCNT (MTU7.TCNT) matches TDDRA (TDDRB) during down-counting of MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT), TCNTSA (TCNTSB) starts up-counting, and when MTU4.TCNT (MTU7.TCNT) matches 0000h, the operation switches to down-counting. When MTU3.TCNT (MTU6.TCNT) matches TCDRA (TCDRB), TCNTSA (TCNTSB) becomes 0000h and stops counting.
TCNTSA (TCNTSB) is compared with the compare register and temporary register, in which the PWM duty is specified, only during the count operation.

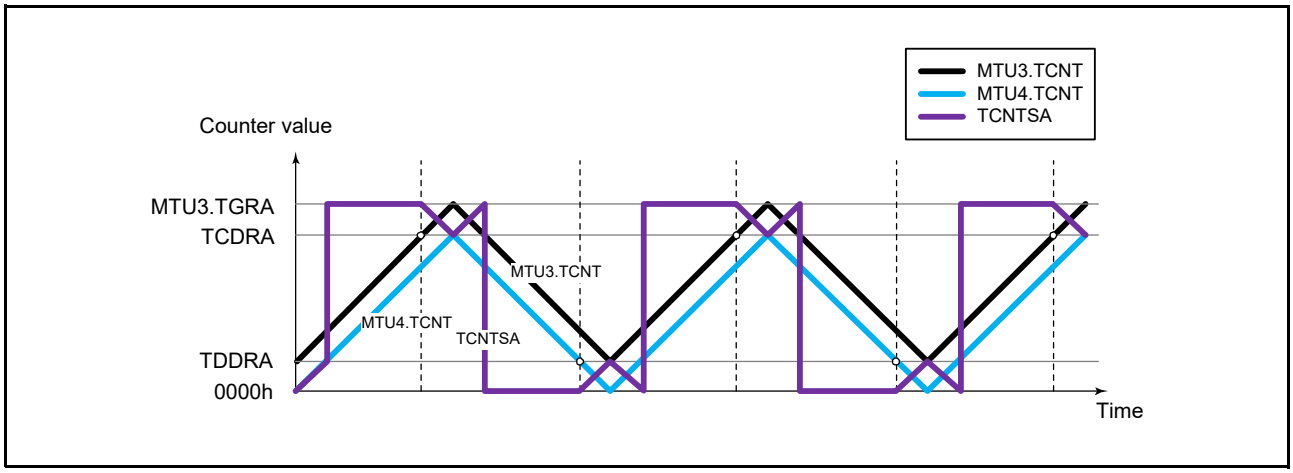


Figure 22.48 Count Operation in Complementary PWM Mode (MTU3 and MTU4)

(b) Register Operation

In complementary PWM mode, nine registers (compare registers, buffer registers, and temporary registers) are used to control the duty ratio for the PWM output. Figure 22.49 shows an example of operation in complementary PWM mode (MTU3 and MTU4).

MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB (MTU6.TGRB, MTU7.TGRA, and MTU7.TGRB) are constantly compared with the counters to generate PWM waveforms. When these registers match the counter, the value set in the OLSN and OLSP bits in the timer output control registers (TOCR1A and TOCR1B) is output from the PWM output pin.

MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD (MTU6.TGRD, MTU7.TGRC, and MTU7.TGRD) are buffer registers for these compare registers.

When the double buffer function is used, MTU3.TGRE, MTU4.TGRE, and MTU4.TGRF (MTU6.TGRE, MTU7.TGRE, and MTU7.TGRF) are also used as buffer registers B. For details of double buffer operation, refer to section 22.3.8 (2) (s), Double Buffer Function in Complementary PWM Mode.

Data in a compare register can be changed by writing new data to the corresponding buffer register. The buffer registers can be read or written at any time.

When modifying data in a buffer register, write to MTU4.TGRD (MTU7.TGRD) last and enable data transfer from the buffer register to a temporary register. At this time, transfer from the TCBRA (TCBRB) register and MTU3.TGRC (MTU6.TGRC) register, which operate as buffer registers for the timer period registers, to temporary registers is also enabled. Data is transferred to all five temporary registers at the same time.

When transfer is enabled in the Ta interval, data written to a buffer register is transferred to the temporary register. The data is not transferred to the temporary register in the Tb1 and Tb2 intervals. Data enabled for transfer in this interval is transferred to the temporary register at the end of this interval.

The value transferred to a temporary register is transferred to the compare register at the end of the Tb1 interval (when matches MTU3.TGRA (MTU6.TGRA) while TCNTSA (TCNTSB) is counting up), or at the end of the Tb2 interval (when matches 0000h while TCNTSA (TCNTSB) is counting down). The timing for transfer from the temporary register to the compare register can be selected with bits MD[3:0] in the timer mode register 1 (TMDR1). Figure 22.49 shows an example in which the trough is selected for the transfer timing.

In the Tb interval in which data is not transferred to the temporary register (Tb1 in Figure 22.49), the temporary register has the same function as the compare register and is compared with the counter. In this interval, therefore, there are two compare match registers for one output phase; the compare register contains the pre-change data and the temporary register contains new data. In this interval, three counters MTU3.TCNT, MTU4.TCNT, and TCNTSA (MTU6.TCNT, MTU7.TCNT, and TCNTSB) and two registers (compare register and temporary register) are compared, and PWM output is controlled accordingly.

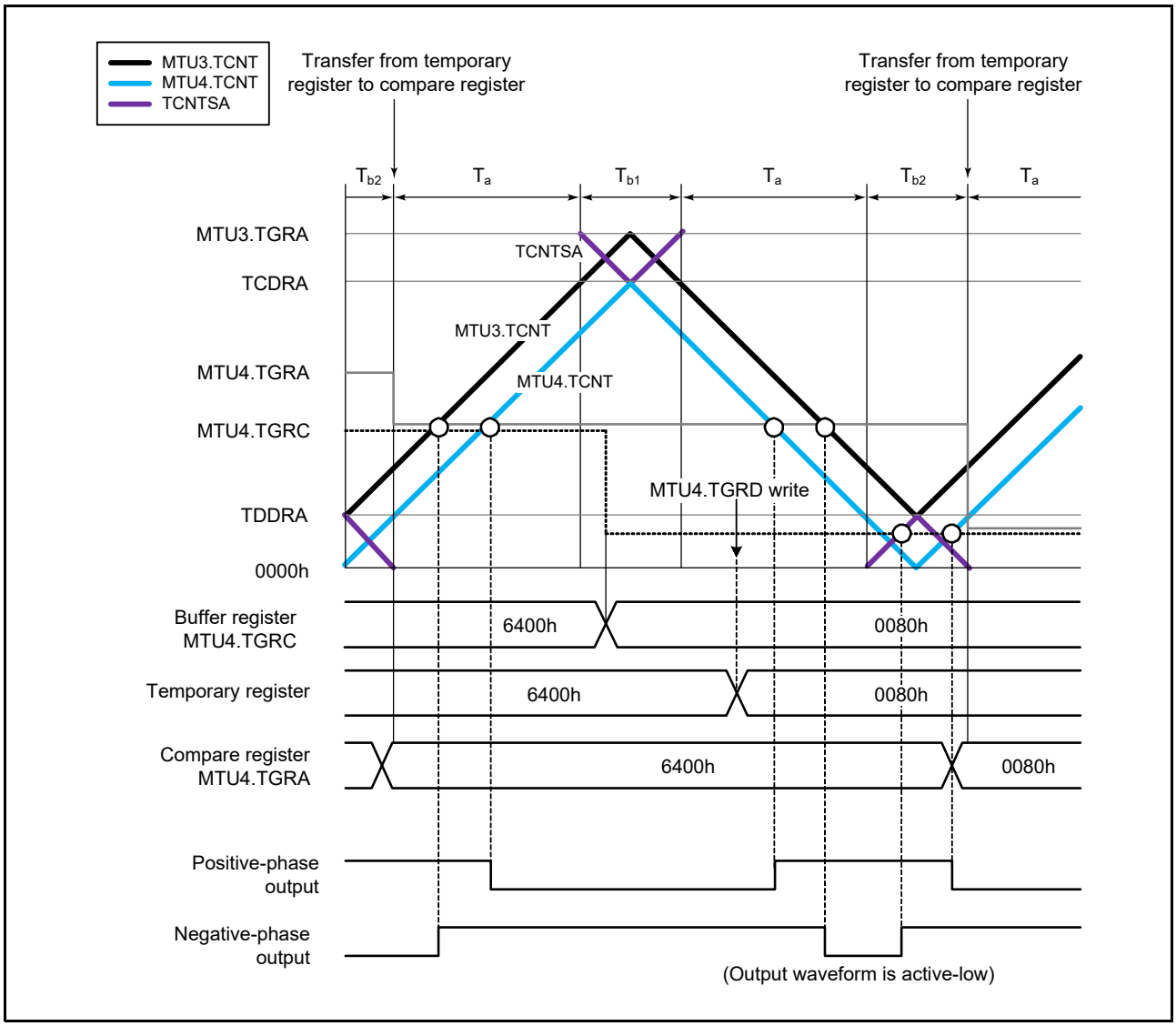


Figure 22.49 Example of Operation in Complementary PWM Mode (MTU3 and MTU4)

(c) Initial Setting

In complementary PWM mode, there are nine registers that require initial setting. In addition, there is a register that specifies whether to generate dead time (it should be used only when dead time generation should be disabled).
Before setting complementary PWM mode with MTU3.TMDR1.MD[3:0] (MTU6.TMDR1.MD[3:0]) bits, initial values should be set in the following registers.
The TOCR1A, TOCR2A, TOCR1B, and TOCR2B registers are used to set the PWM output level. MTU3.TGRC (MTU6.TGRC) operates as the buffer register for MTU3.TGRA (MTU6.TGRA), and should be set with 1/2 the PWM period + dead time Td. The timer period buffer register (TCBRA or TCBRB) operates as the buffer register for the timer period data register (TCDRA or TCDRB), and should be set with 1/2 the PWM period. Set dead time Td in the timer dead time data register (TDDRA or TDDRB).
When dead time is not needed, the TDER bit in the timer dead time enable register (TDERA or TDERB) should be set to 0, MTU3.TGRC and MTU3.TGRA (MTU6.TGRC and MTU6.TGRA) should be set to 1/2 the PWM carrier period + 1, and TDDRA (TDDRB) should be set to 1.
Set the respective initial PWM duty values in three buffer registers A (MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD (MTU6.TGRD, MTU7.TGRC, and MTU7.TGRD)).
Set three buffer registers B (MTU3.TGRE, MTU4.TGRE, and MTU4.TGRF (MTU6.TGRE, MTU7.TGRE, and MTU7.TGRF)) only when the double buffer function is used.
The values set in the five buffer registers excluding TDDRA (TDDRB) are transferred to the corresponding compare registers as soon as complementary PWM mode is set.
Set MTU4.TCNT (MTU7.TCNT) to 0000h before setting complementary PWM mode.

Table 22.77 Registers and Counters Requiring Initial Setting

Register and Counter	Setting
TOCR1A, TOCR2A, TOCR1B, TOCR2B	PWM output level
MTU3.TGRC MTU6.TGRC	1/2 PWM period + dead time Td (1/2 PWM period + 1 when dead time generation is disabled by TDERA or TDERB)
TDDRA, TDDRB	Dead time Td (1 when dead time generation is disabled by TDERA or TDERB)
TCBRA, TCBRB	1/2 PWM period
MTU3.TGRD, MTU4.TGRC, MTU4.TGRD MTU6.TGRD, MTU7.TGRC, MTU7.TGRD	Initial PWM duty ratio value for each phase
MTU3.TGRE, MTU4.TGRE, MTU4.TGRF MTU6.TGRE, MTU7.TGRE, MTU7.TGRF	Initial PWM duty ratio value for each phase (only when double buffer function is used)
MTU4.TCNT MTU7.TCNT	0000h

Note: The value set in MTU3.TGRC (MTU6.TGRC) should be the sum of 1/2 the PWM period set in TCBRA (TCBRB) and dead time Td set in TDDRA (TDDRB). When dead time generation is disabled by TDERA (TDERB), TGRC should be set to 1/2 the PWM period + 1.

(d) PWM Output Level Setting

In complementary PWM mode, the PWM output level is set with bits OLSN and OLSP in timer output control register 1 (TOCR1A or TOCR1B) or bits OLS1P to OLS3P and OLS1N to OLS3N in timer output control register 2 (TOCR2A or TOCR2B).
The output level can be set for each of the three positive phases and three negative phases of 6-phase output.
Complementary PWM mode should be cleared before setting or changing output levels.

(e) Dead Time Setting

In complementary PWM mode, dead time can be set for PWM output.
The dead time is set in the timer dead time data register (TDDRA or TDDRB). The value set in TDDRA (TDDRB) is used as the MTU3.TCNT (MTU6.TCNT) counter start value and creates a non-overlapping interval between MTU3.TCNT (MTU6.TCNT) and MTU4.TCNT (MTU7.TCNT). Complementary PWM mode should be cleared before changing the contents of TDDRA (TDDRB).

(f) Dead Time Suppressing

Dead time generation is suppressed by setting the TDER bit in the timer dead time enable register (TDERA or TDERB) to 0. TDERA (TDERB) can be set to 0 only when 0 is written to it after reading TDER = 1.
MTU3.TGRA and MTU3.TGRC (MTU6.TGRA and MTU6.TGRC) should be set to 1/2 PWM period + 1 and the timer dead time data register (TDDRA or TDDRB) should be set to 1.
By the above settings, PWM waveforms without dead time can be obtained. Figure 22.50 shows an example of operation without dead time (MTU3 and MTU4).

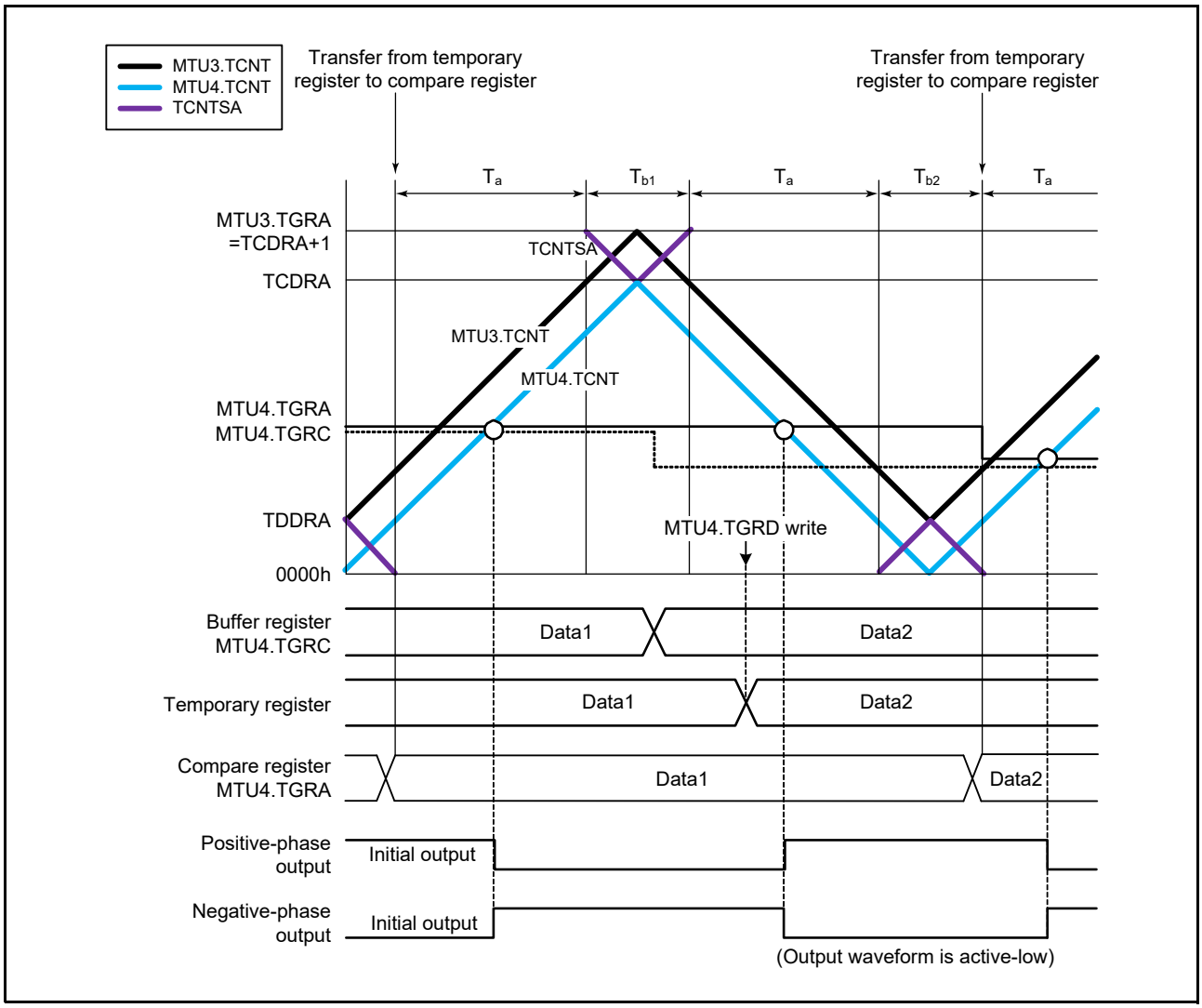


Figure 22.50 Example of Operation without Dead Time (MTU3 and MTU4)

(g) PWM Period Setting

In complementary PWM mode, the PWM period is set in two registers—MTU3.TGRA (MTU6.TGRA), in which the MTU3.TCNT (MTU6.TCNT) upper limit value is set, and TCDRA (TCDRB), in which the MTU4.TCNT (MTU7.TCNT) upper limit value is set. The settings should be made so as to achieve the following relationship between these two registers:

With dead time: MTU3.TGRA (MTU6.TGRA) setting = TCDRA (TCDRB) setting + TDDRA (TDDRB) setting
Without dead time: MTU3.TGRA (MTU6.TGRA) setting = TCDRA (TCDRB) setting + 1

In addition, the settings should be made so as to achieve the following relationship between the TCDRA (TCDRB) register and the TDDRA (TDDRB) register:

TCDRA (TCDRB) setting > TDDRA (TDDRB) setting × 2 + 2

The MTU3.TGRA and TCDRA (MTU6.TGRA and TCDRB) settings are made by setting values in buffer registers MTU3.TGRC and TCBRA (MTU6.TGRC and TCBRB). When data is written to MTU4.TGRD (MTU7.TGRD) to enable transfers, the values set in MTU3.TGRC and TCBRA (MTU6.TGRC and TCBRB) are transferred simultaneously to the MTU3.TGRA and TCDRA (MTU6.TGRA and TCDRB) with the transfer timing selected with the MTU3.TMDR1.MD[3:0] (MTU6.TMDR1.MD[3:0]) bits.
The new PWM period is reflected from the next cycle when data is updated at the crest, or from the current cycle when updated in the trough. Figure 22.51 illustrates the operation when the PWM period is updated at the crest.
Refer to the following section, (h), Register Data Updating, for the method of updating the data in each buffer register.

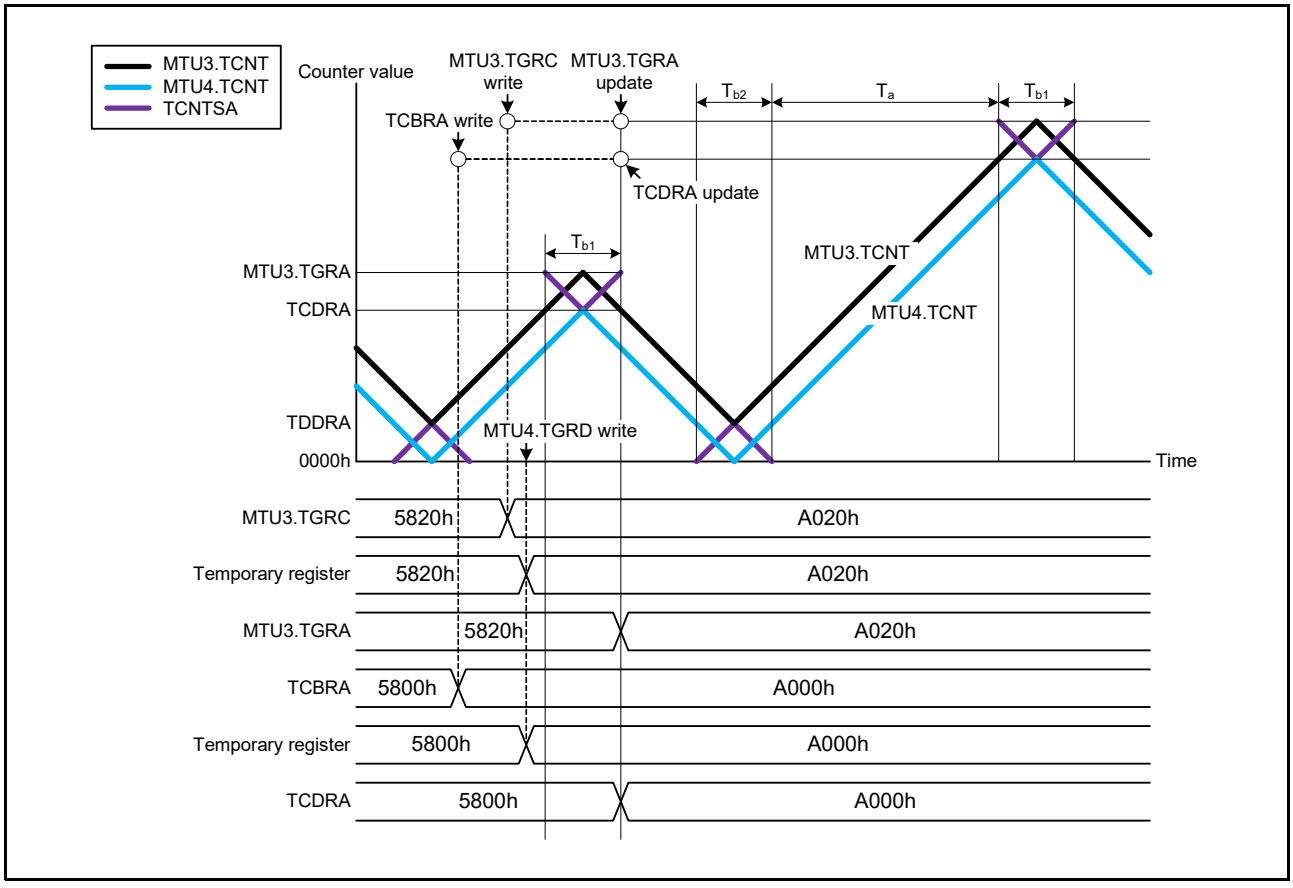


Figure 22.51 Example of PWM Period Updating (MTU3 and MTU4)

(h) Register Data Updating

The buffer registers are used to update the data in five compare registers for the PWM duty and PWM period in complementary PWM mode. The update data can be written to the buffer register at any time.

There is a temporary register between each of these registers and its buffer register. While subcounter TCNTSA (TCNTSB) is not counting, if buffer register data is updated, the temporary register value also changes. Data is not transferred from buffer registers to temporary registers while TCNTSA (TCNTSB) is counting; in this case, the value written to a buffer register is transferred after TCNTSA (TCNTSB) halts.

The temporary register value is transferred to the compare register at the data update timing set with MTU3.TMDR1.MD[3:0] (MTU6.TMDR1.MD[3:0]) bits. Figure 22.52 shows an example of data updating in complementary PWM mode (MTU3 and MTU4). This example shows the mode in which data is updated at both the counter crest and trough.

When updating buffer register data, be sure to write to MTU4.TGRD (MTU7.TGRD) at the end of the update. Data is transferred from buffer registers to the temporary registers simultaneously for all five registers after the write to MTU4.TGRD (MTU7.TGRD).

Even when not updating all five registers or when not updating the MTU4.TGRD (MTU7.TGRD) data, be sure to write to MTU4.TGRD (MTU7.TGRD) after writing data to the registers to be updated. In this case, the data written to MTU4.TGRD (MTU7.TGRD) should be the same as the data prior to the write operation.

Refer to section 22.3.8 (2) (s), Double Buffer Function in Complementary PWM Mode, for data updating when the double buffer function is used.

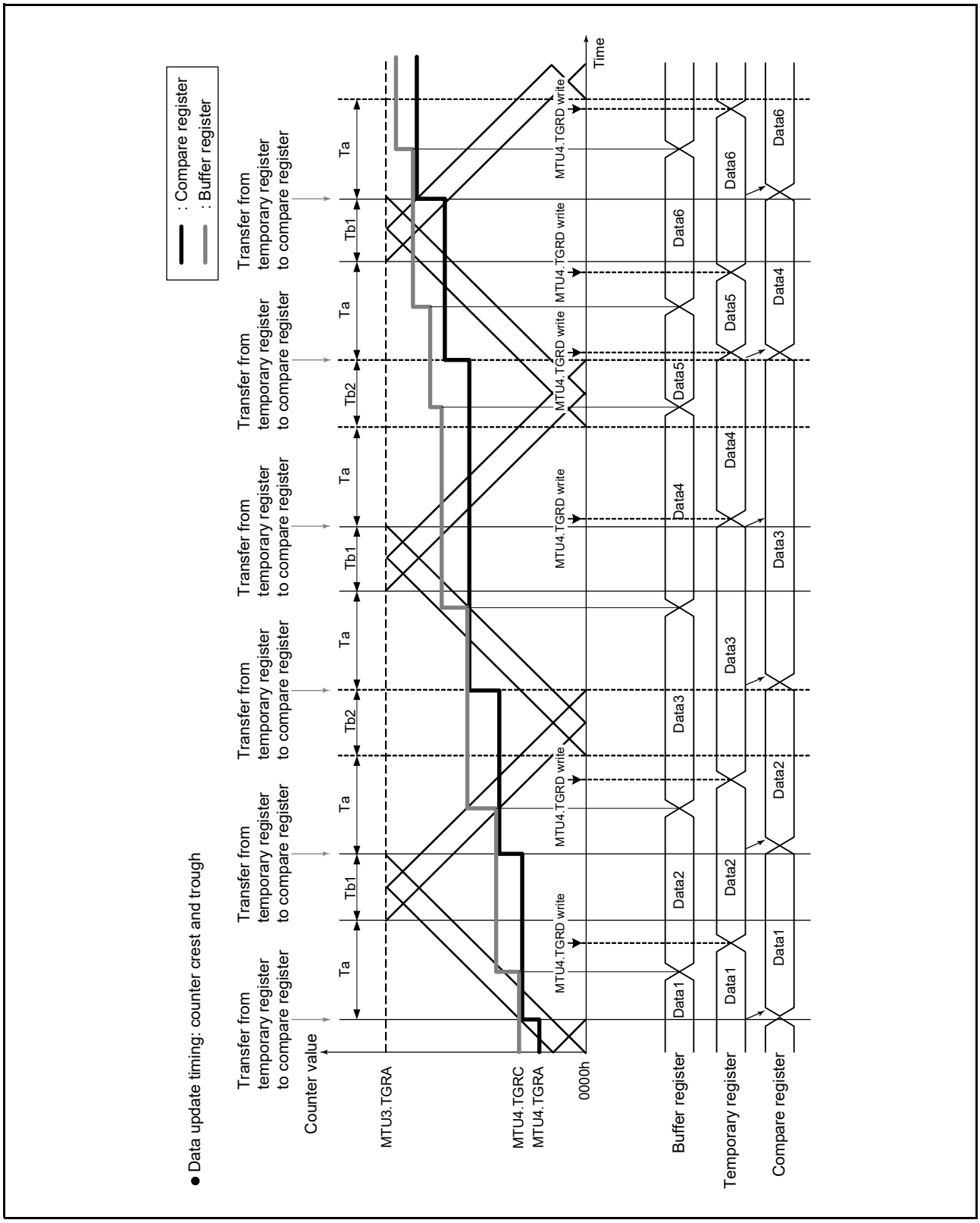


Figure 22.52 Example of Data Updating in Complementary PWM Mode (MTU3 and MTU4)

(i) Initial Output in Complementary PWM Mode

In complementary PWM mode, the initial output is determined by the setting of the OLSN and OLSP bits in the TOCR1A (TOCR1B) register or the OLS1N to OLS3N and OLS1P to OLS3P bits in the TOCR2A register (TOCR2B). This initial output is the non-active level of the PWM output and continues from when complementary PWM mode is set with the MTU3.TMDR1 (MTU6.TMDR1) until MTU4.TCNT (MTU7.TCNT) exceeds the value set in the TDDRA (TDDRB) register. Figure 22.53 shows an example of the initial output in complementary PWM mode.
An example of the waveform when the initial PWM duty ratio value is smaller than the TDDRA (TDDRB) value is shown in Figure 22.54.

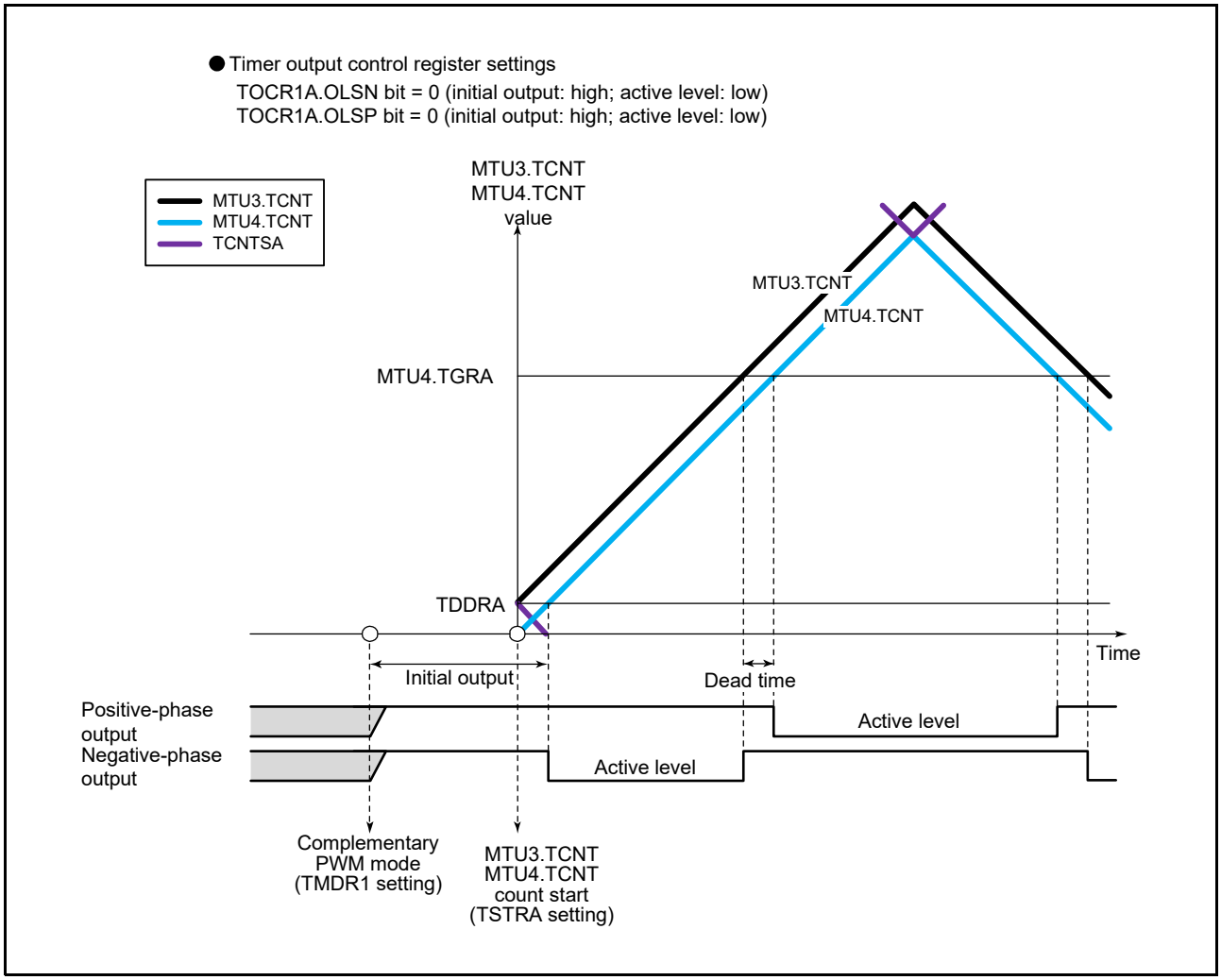


Figure 22.53 Example of Initial Output in Complementary PWM Mode (MTU3 and MTU4) (1))

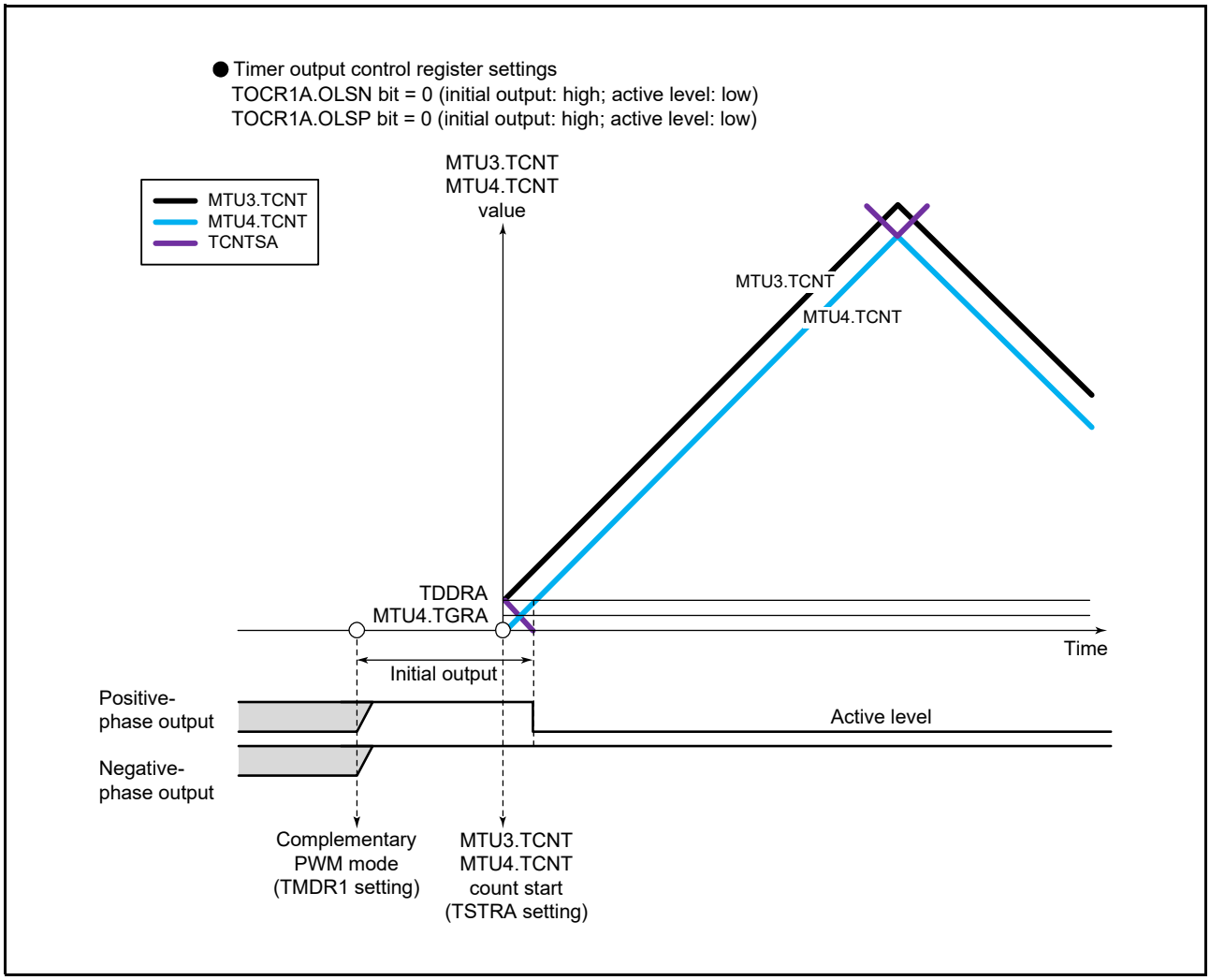


Figure 22.54 Example of Initial Output in Complementary PWM Mode (MTU3 and MTU4) (2)

(j) Method for Generating PWM Output in Complementary PWM Mode

In complementary PWM mode, six phases (three positive and three negative) PWM waveforms can be output. Dead time can be set for PWM waveforms to be output.

A PWM waveform is generated by output of the level selected in the timer output control register in the event of a compare match between a counter and a compare register. While TCNTSA (TCNTSB) is counting, the compare register and temporary register values are simultaneously compared to create consecutive PWM output from 0 to 100% duty ratio. The relative timing of turn-on and turn-off compare match occurrence may vary, but the compare match that turns off each phase takes precedence to secure the dead time and ensure that the positive-phase and negative-phase turn-on times do not overlap. Figure 22.55 to Figure 22.57 show examples of waveform generation in complementary PWM mode.

The positive-phase and negative-phase turn-off timing is generated by a compare match with the counter indicated by a solid line, and the turn-on timing is generated by a compare match with the counter indicated by a dotted line, which operates with a delay equal to the dead time behind the counter indicated by a solid line. In the T1 period, compare match a that turns off the negative phase has the highest priority, and compare matches before a are ignored. In the T2 period, compare match c that turns off the positive phase has the highest priority, and compare matches before c are ignored.

In most cases, compare matches occur in the order a → b → c → d (or c → d → a' → b') as shown in Figure 22.55.

If compare matches deviate from the a → b → c → d order, since the time for which the negative phase is off is shorter than twice the dead time, the positive phase is not turned on. If compare matches deviate from the c → d → a' → b' order, since the time for which the positive phase is off is shorter than twice the dead time, the negative phase is not turned on.

As shown in Figure 22.56, if compare match c follows compare match a before compare match b, compare match b is ignored and the negative phase is turned on by compare match d. This is because turning off the positive phase has priority due to the occurrence of compare match c (positive-phase off timing) before compare match b (positive-phase on timing) (consequently, the waveform does not change because the positive phase goes from off to off).

Similarly, in the example in Figure 22.57, turning off the negative phase has priority due to the occurrence of compare match a' (negative-phase off timing) before compare match d (negative-phase on timing). As a result, the negative phase is not turned on.

Thus, in complementary PWM mode, compare matches at turn-off timings take precedence, and turn-on timing compare matches that occur before a turn-off timing compare match are ignored.

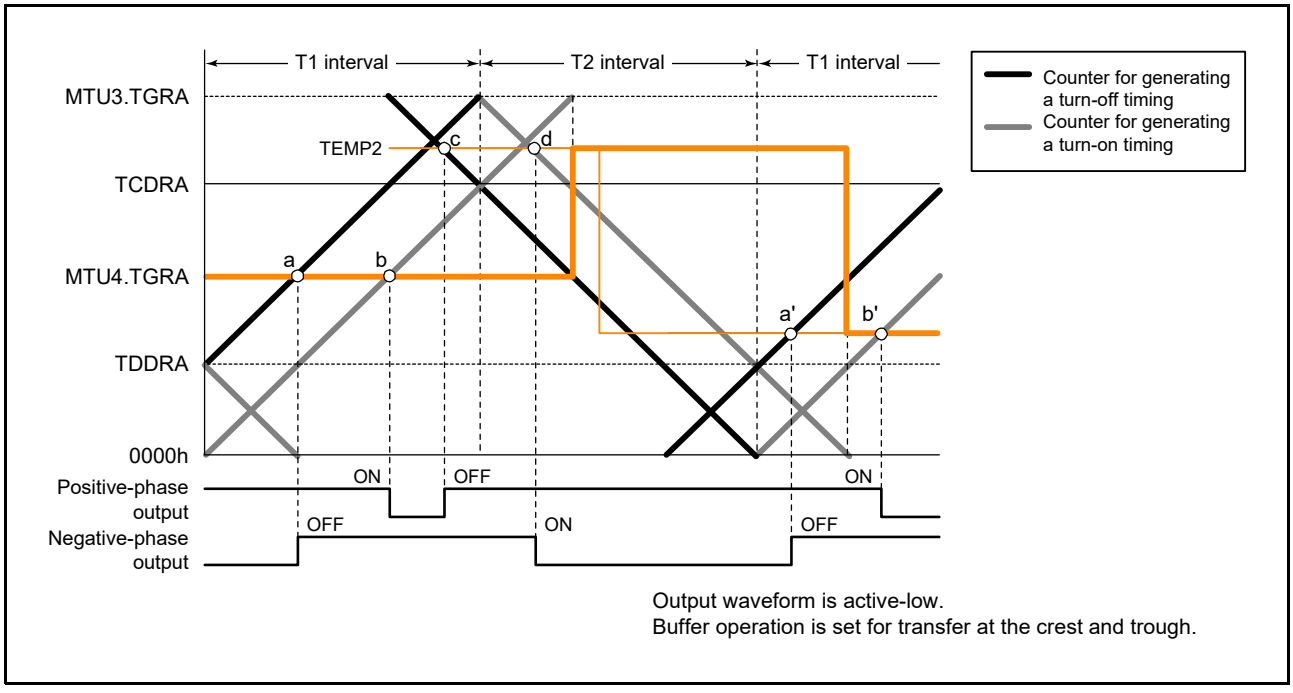


Figure 22.55 Example of Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (1)

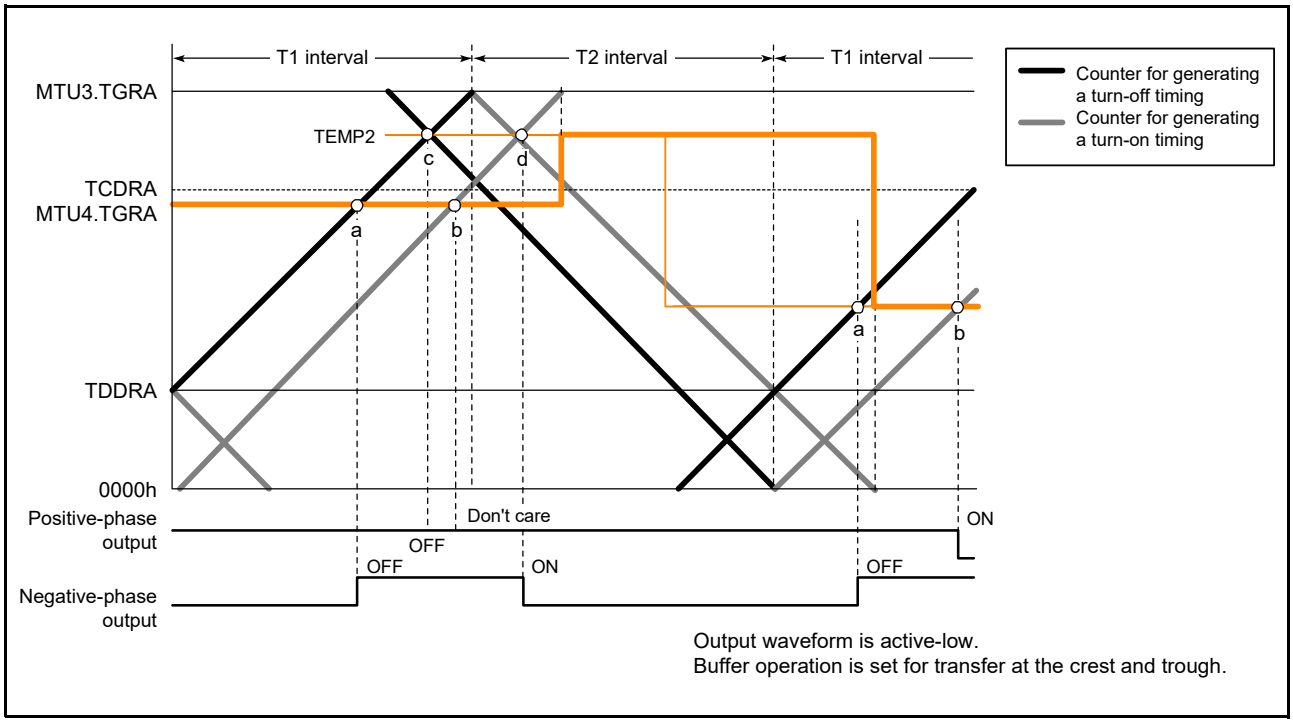


Figure 22.56 Example of Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (2)

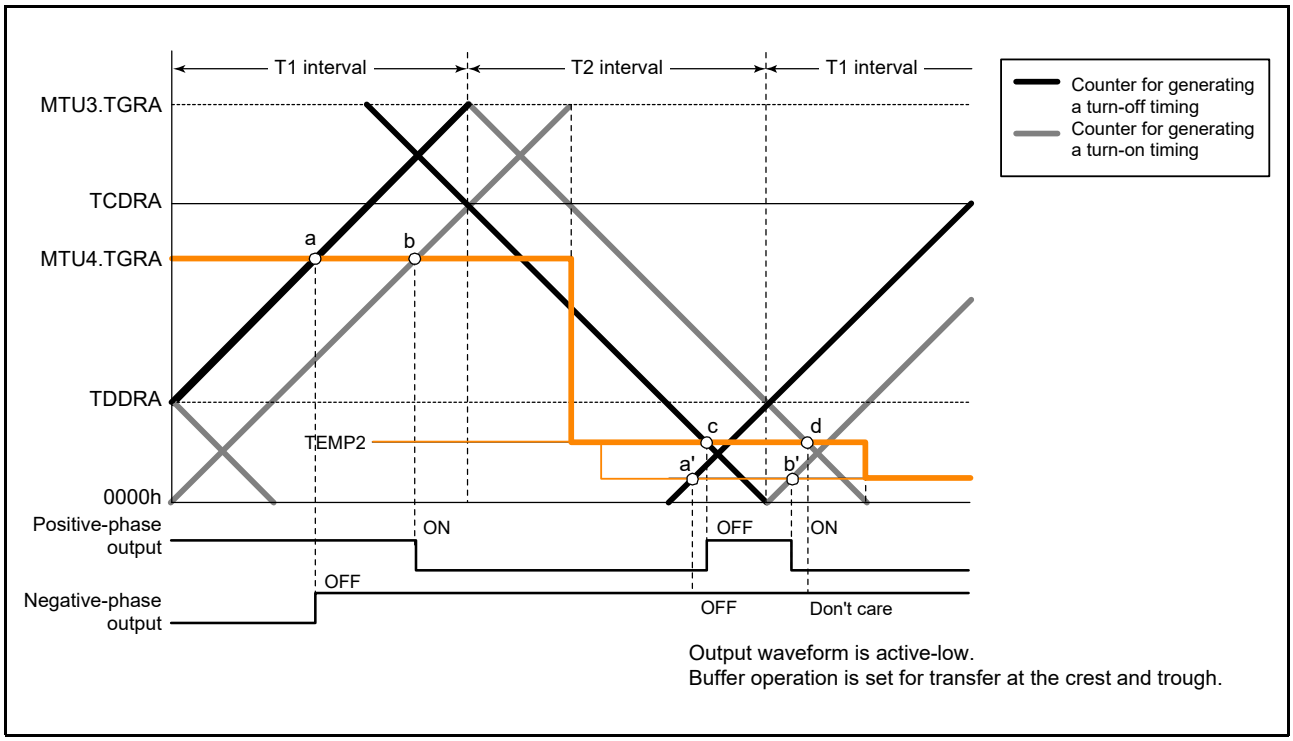


Figure 22.57 Example of Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (3)

(k) 0% and 100% Duty Ratio Output in Complementary PWM Mode

In complementary PWM mode, 0% and 100% duty PWM output can be output as required. Figure 22.58 to Figure 22.62 show output examples.

A 100% duty waveform is output when the compare register value is set to 0000h. The waveform in this case has a positive phase with a 100% on-state. A 0% duty waveform is output when the compare register value is set to the same value as MTU3.TGRA (MTU6.TGRA). The waveform in this case has a positive phase with a 100% off-state.

Turn-on and turn-off compare matches occur simultaneously, but if a turn-on compare match and turn-off compare match for the same phase occur simultaneously, both compare matches are ignored and the waveform does not change.

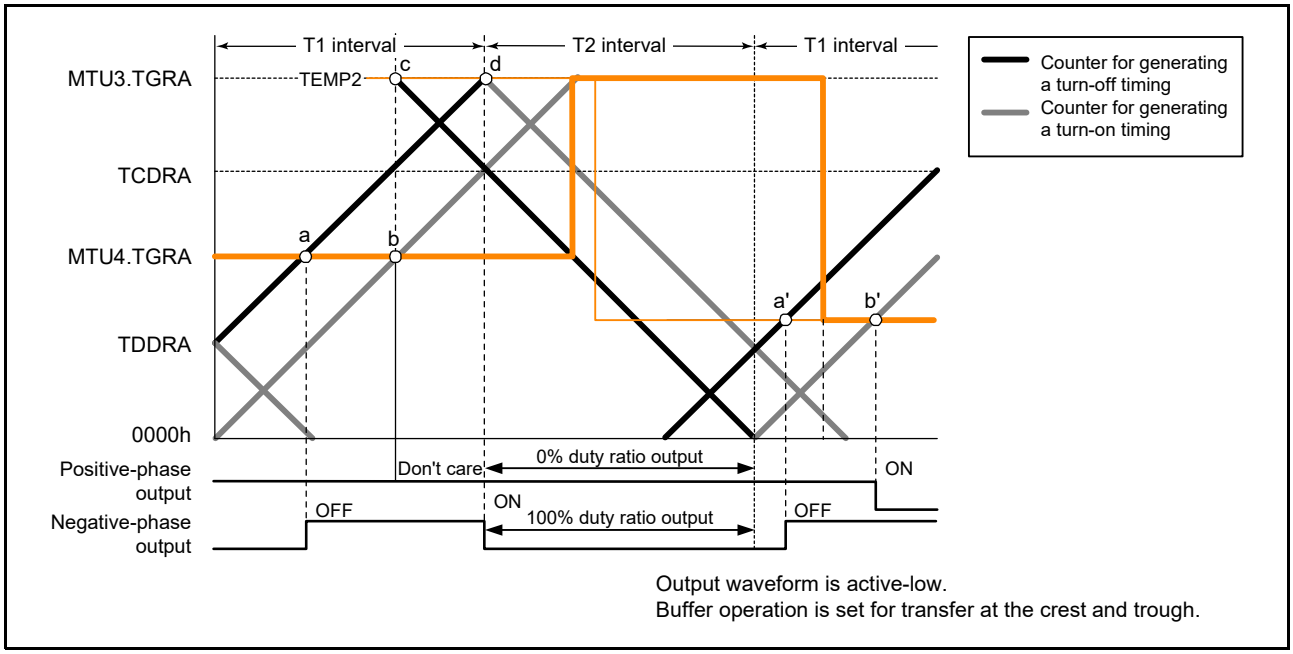


Figure 22.58 Example of 0% and 100% Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (1)

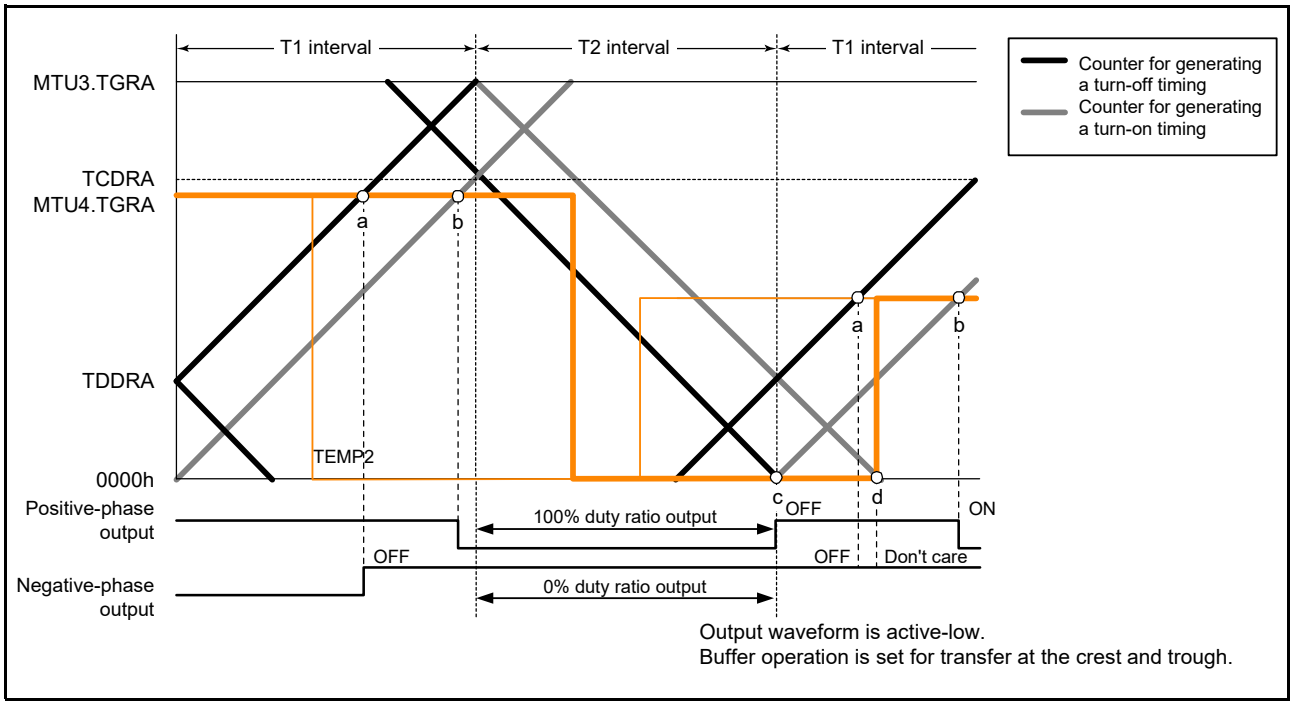


Figure 22.59 Example of 0% and 100% Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (2)

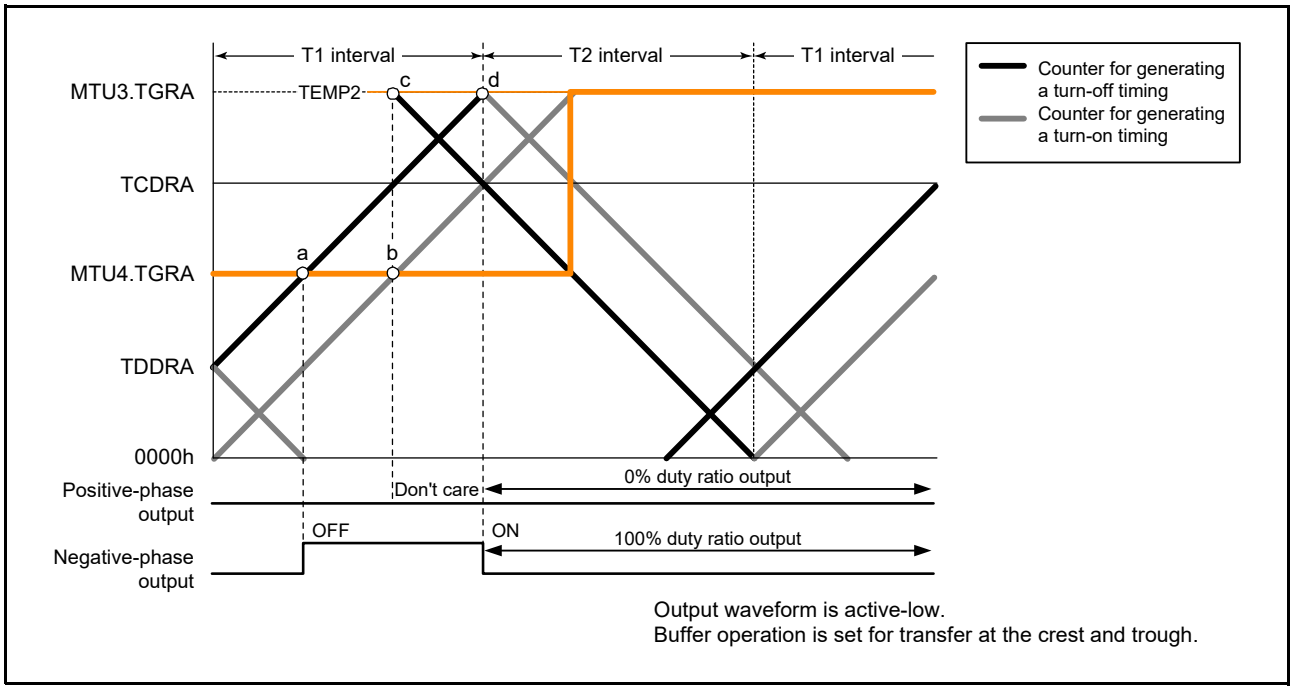


Figure 22.60 Example of 0% and 100% Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (3)

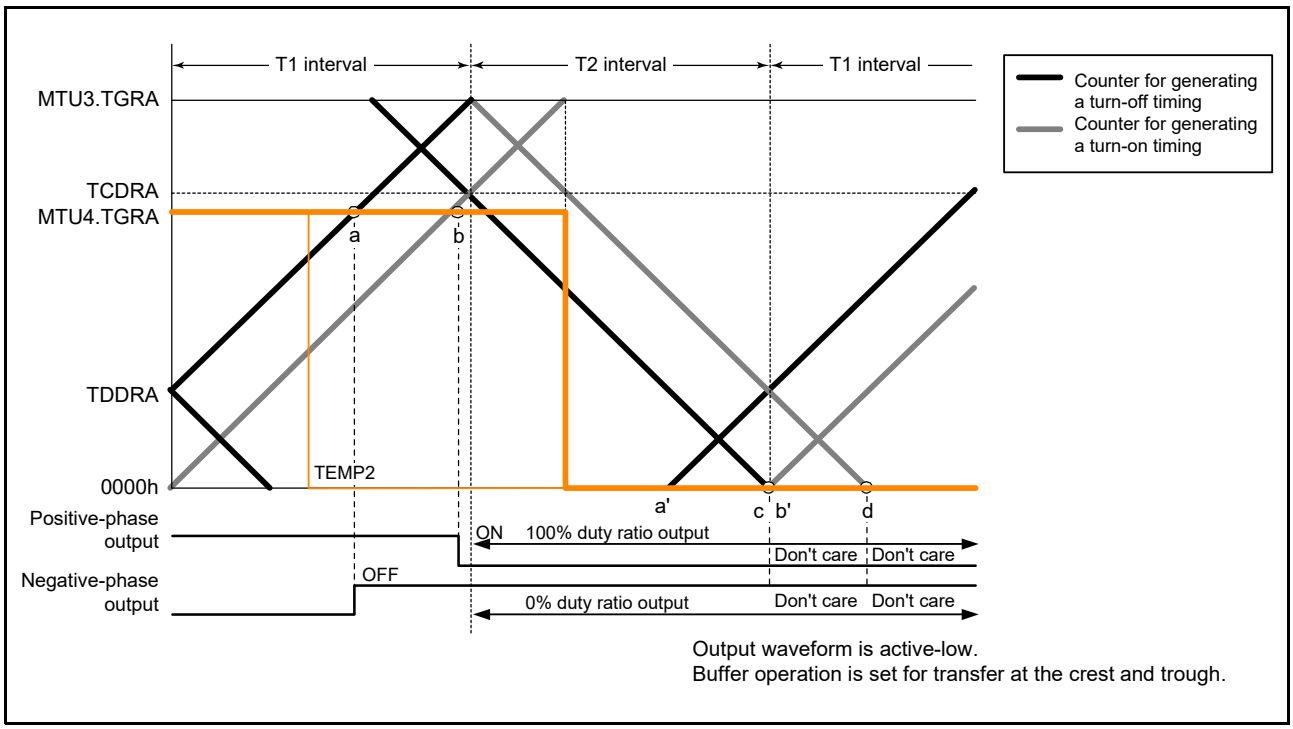


Figure 22.61 Example of 0% and 100% Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (4)

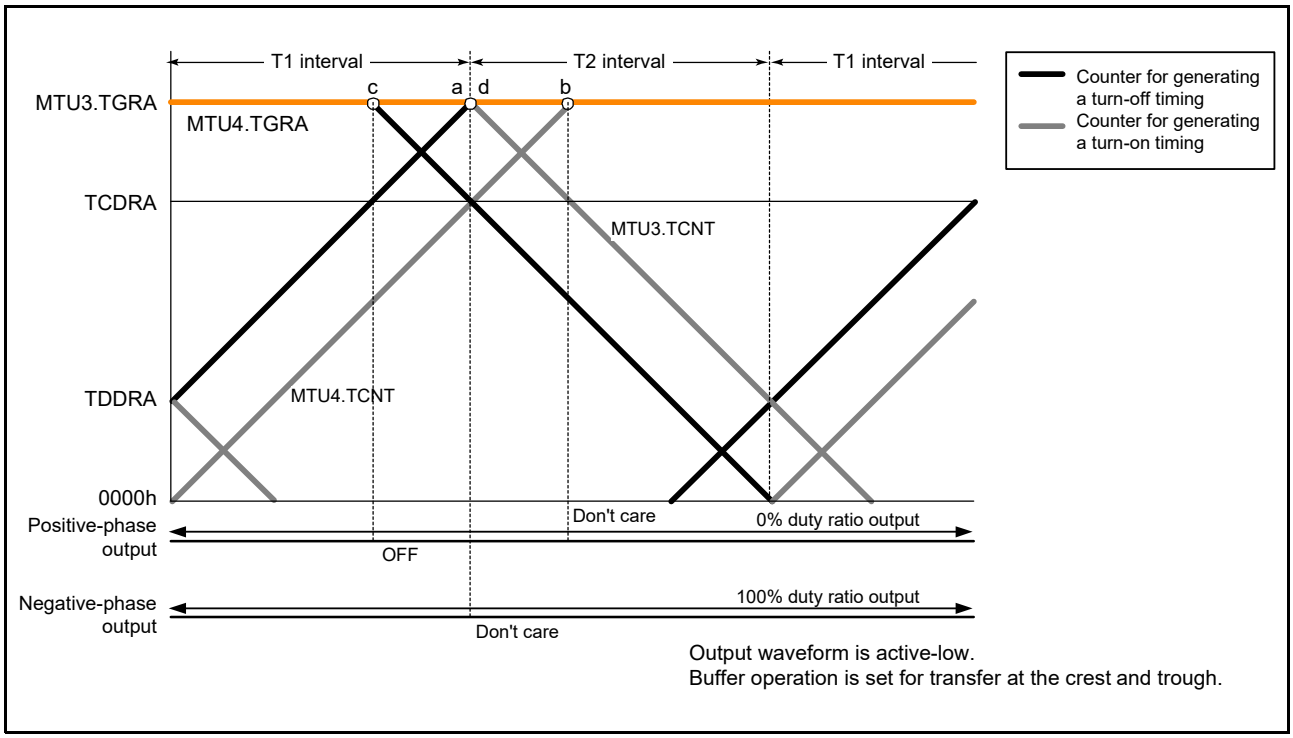


Figure 22.62 Example of 0% and 100% Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (5)

(l) Toggle Output Synchronized with PWM Period

In complementary PWM mode, toggle output from the PWM output pin in synchronization with the PWM period can be enabled by setting the PSYE bit in the TOCR1A (TOCR1B) register to 1. An example of a toggle output waveform is shown in Figure 22.63.

This output is toggled by a compare match between MTU3.TCNT and MTU3.TGRA (MTU6.TCNT and MTU6.TGRA) and a compare match between MTU4.TCNT (MTU7.TCNT) and 0000h.

The MTIOC3A (MTIOC6A) pin is assigned for this toggle output. The initial output is high-level output.

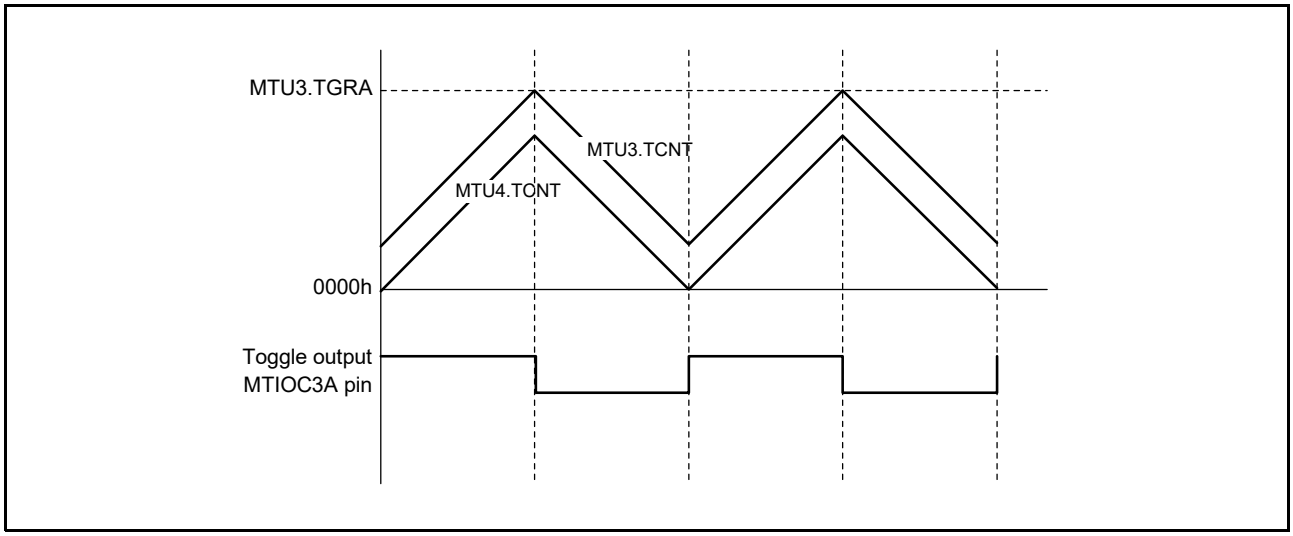


Figure 22.63 Example of Toggle Output Waveform Synchronized with PWM Output (MTU3 and MTU4)

(m) Counter Clearing by Another Channel

In complementary PWM mode, MTU3.TCNT, MTU4.TCNT, and TCNTSA (MTU6.TCNT, MTU7.TCNT, and TCNTSB) can be cleared by another channel source when a mode for synchronization with another channel is specified through the TSYRA (TSYRB) register and synchronous clearing is selected with MTU3.TCR.CCLR[2:0] (MTU6.TCR.CCLR[2:0]) bits.

Figure 22.64 illustrates an example of this operation.

Use of this function enables a counter to be cleared and restarted through an external signal.

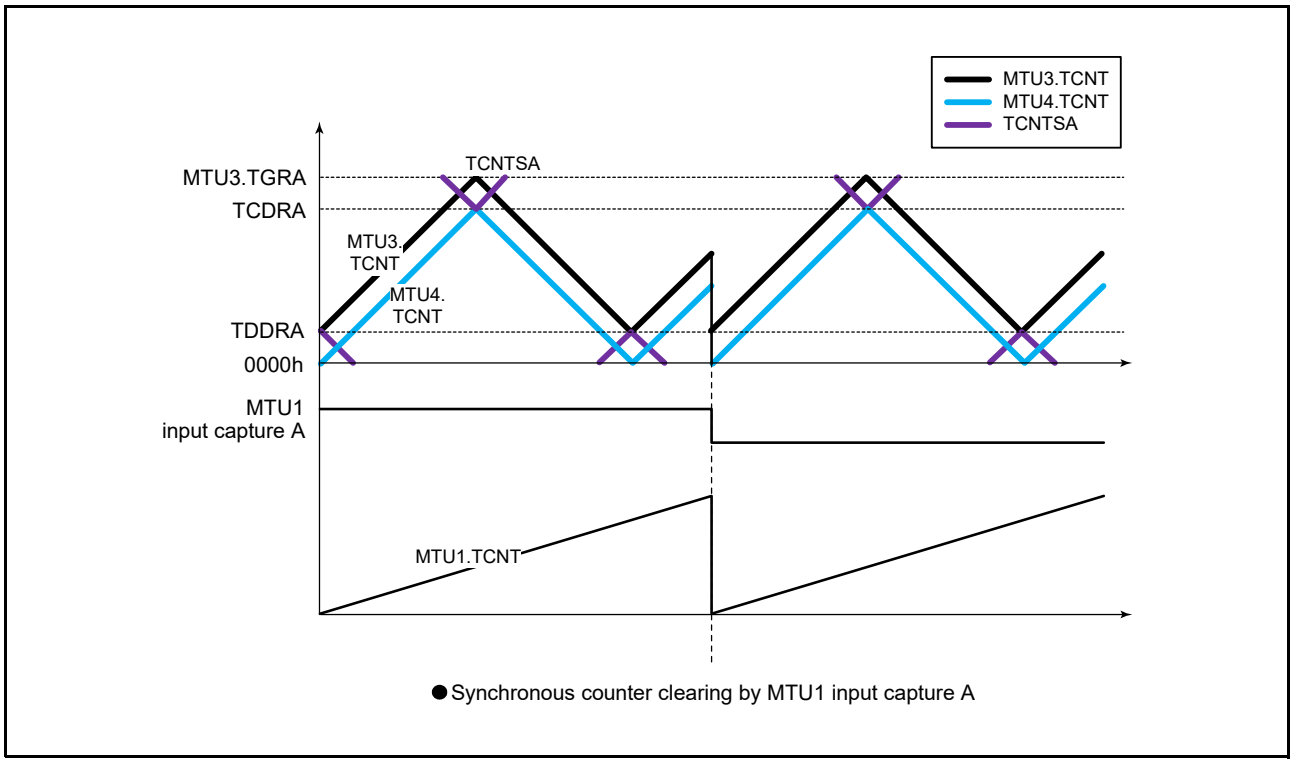


Figure 22.64 Counter Clearing Synchronized with Another Channel (MTU3 and MTU4)

(n) Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

Setting the WRE bit in TWCRA (TWCRB) to 1 suppresses initial output when synchronous counter clearing occurs in the Tb interval (Tb2 interval) at the trough in complementary PWM mode and controls abrupt change in duty cycle at synchronous counter clearing.

Initial output suppression through the WRE bit = 1 is applicable only when synchronous clearing occurs in the Tb2 interval as indicated by (10) or (11) in Figure 22.65. When synchronous clearing occurs outside that interval, the initial value specified by the OLSN and OLSP bits in TOCR1A (TOCR1B) is output. Even in the Tb2 interval, if synchronous clearing occurs in the initial value output period (indicated by (1) in Figure 22.65) immediately after the counters start operation, initial value output is not suppressed.

This function can be used in both channel combinations of MTU3 and MTU4, and MTU6 and MTU7. In MTU3 and MTU4, synchronous clearing in any of MTU0 to MTU2 can cause counter clearing; in MTU6 and MTU7,compare match or input capture in any of MTU0 to MTU2 can cause counter clearing.

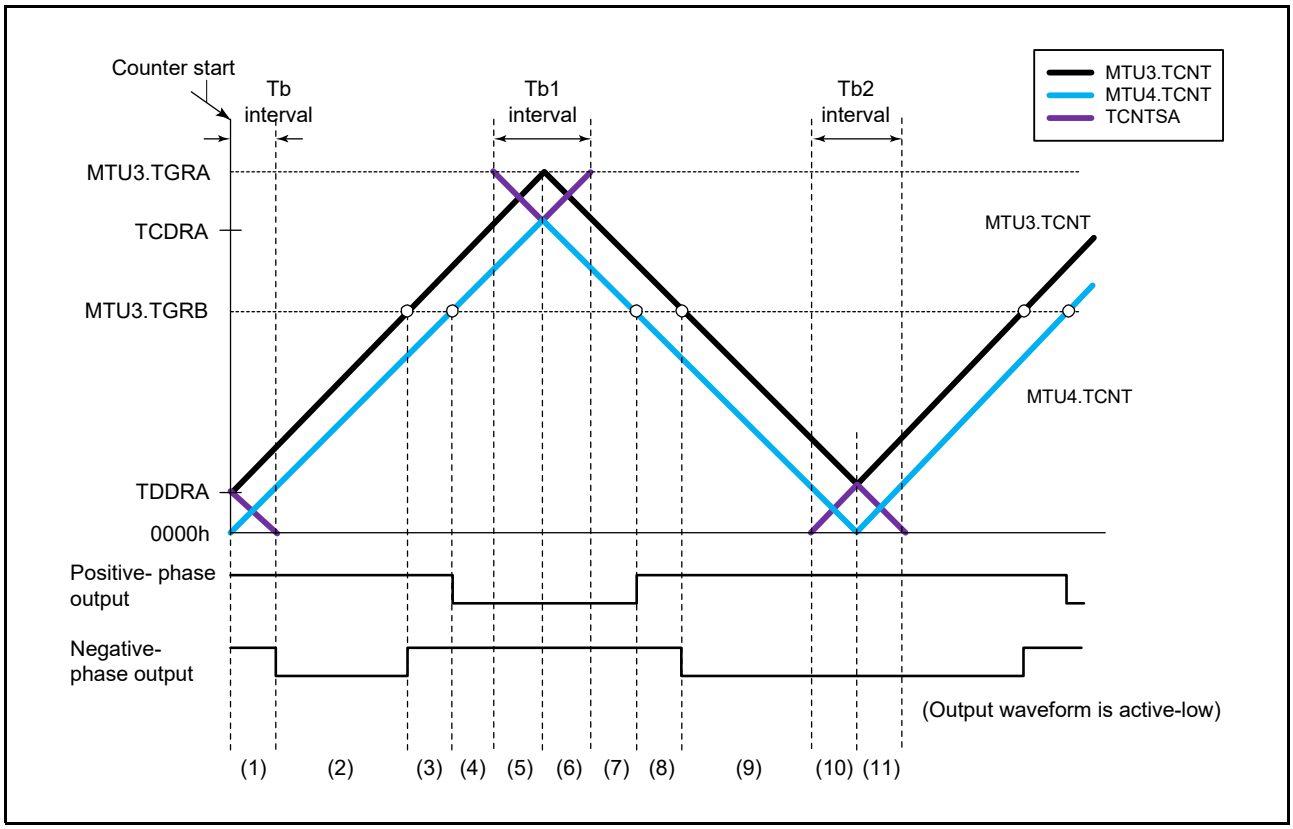


Figure 22.65 Timing for Synchronous Counter Clearing (MTU3 and MTU4)

- Example of Procedure for Setting Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode.

An example of the procedure for setting output waveform control at synchronous counter clearing in complementary PWM mode is shown in Figure 22.66.

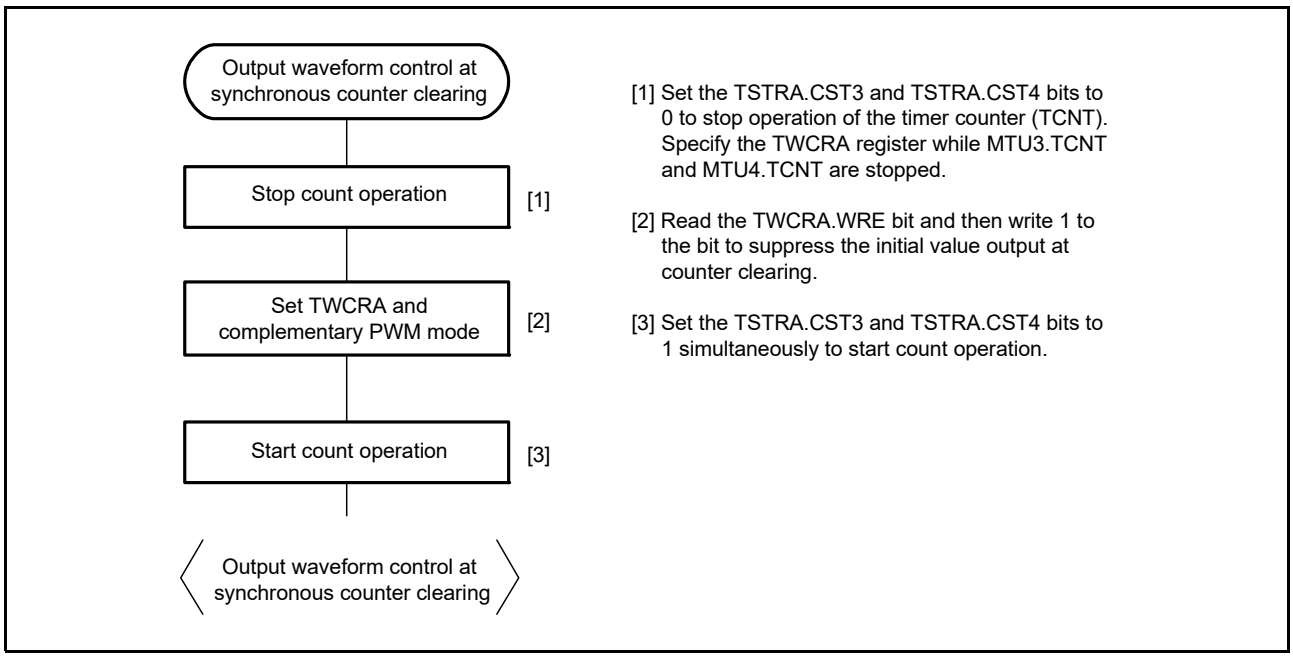


Figure 22.66 Example of Procedure for Setting Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode (MTU3 and MTU4)

- Examples of Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

Figure 22.67 to Figure 22.70 show examples of output waveform control in which MTU3 and MTU4 operate in complementary PWM mode and synchronous counter clearing is generated while the WRE bit in TWCRA is set to 1. In the examples shown in Figure 22.67 to Figure 22.70, synchronous counter clearing occurs at timing (3), (6), (8), and (11) shown in Figure 22.65, respectively.

In MTU6 and MTU7, these examples are equivalent to the cases when MTU6 and MTU7 operate in complementary PWM mode and synchronous counter clearing is generated while the SCC bit is set to 0 and the WRE bit is set to 1 in TWCRB.

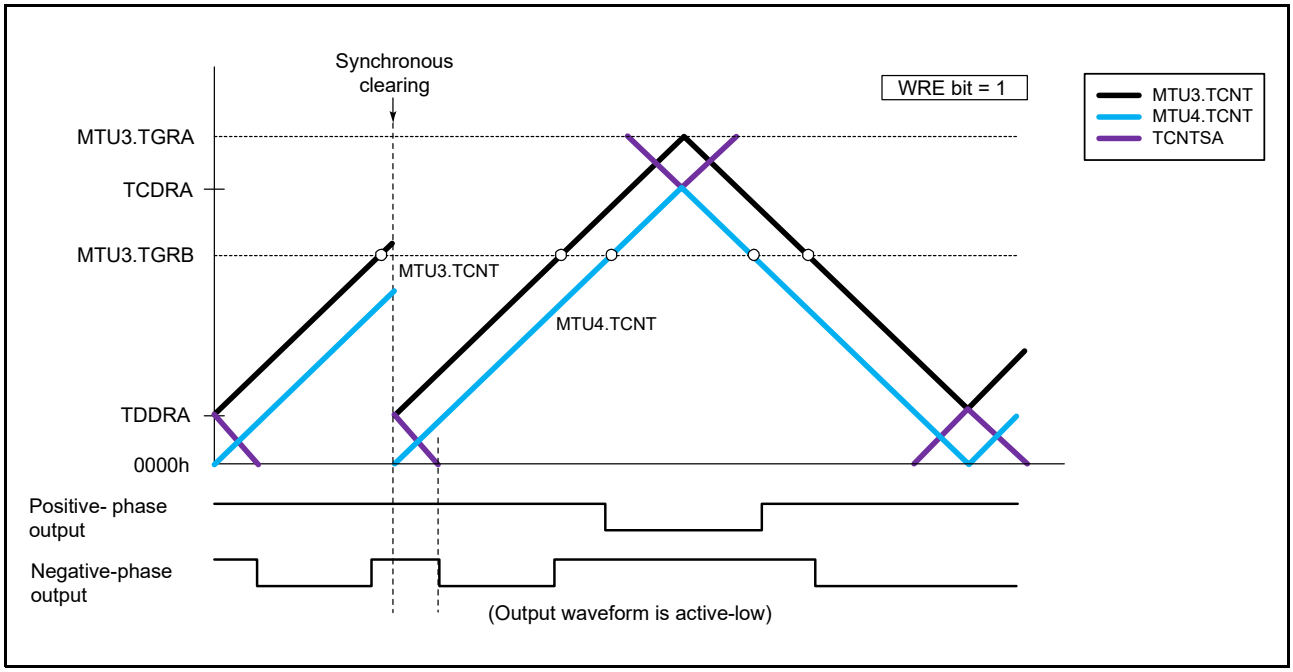


Figure 22.67 Example of Synchronous Clearing in Dead Time during Up-Counting (Timing (3) in Figure 22.65; TWCRA.WRE Bit is 1)

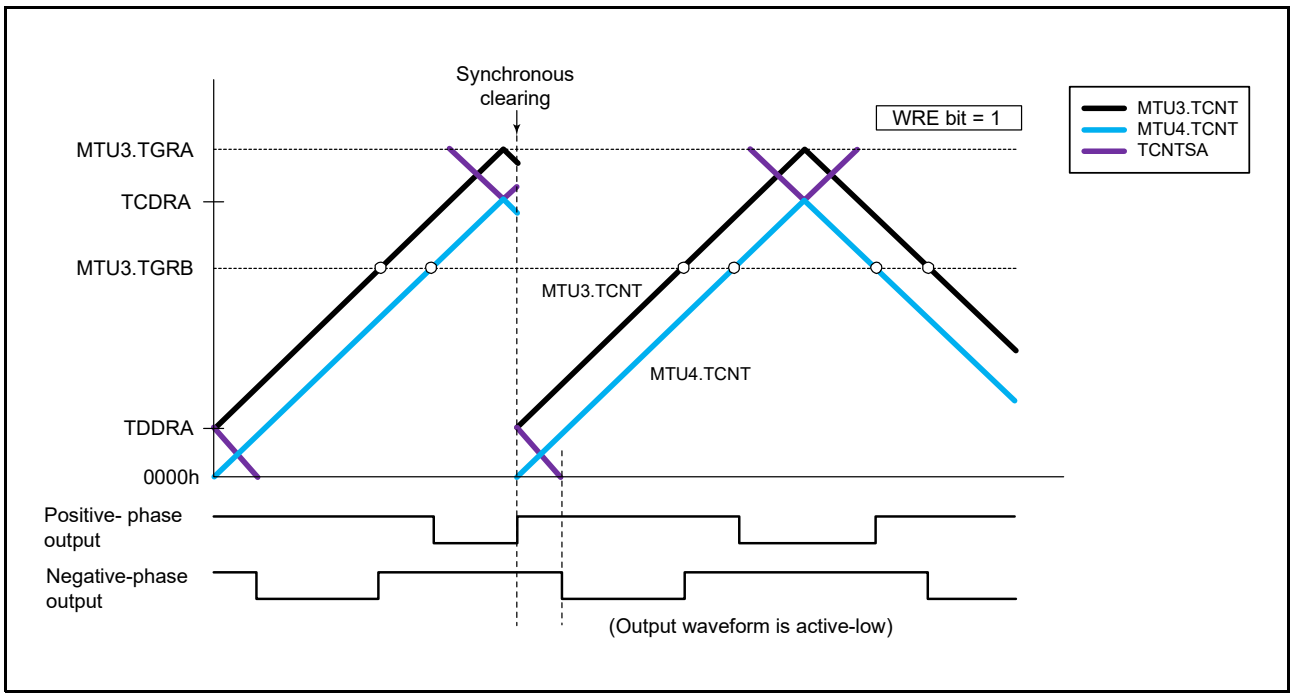


Figure 22.68 Example of Synchronous Clearing in Tb1 interval (Timing (6) in Figure 22.65; TWCRA.WRE Bit is 1)

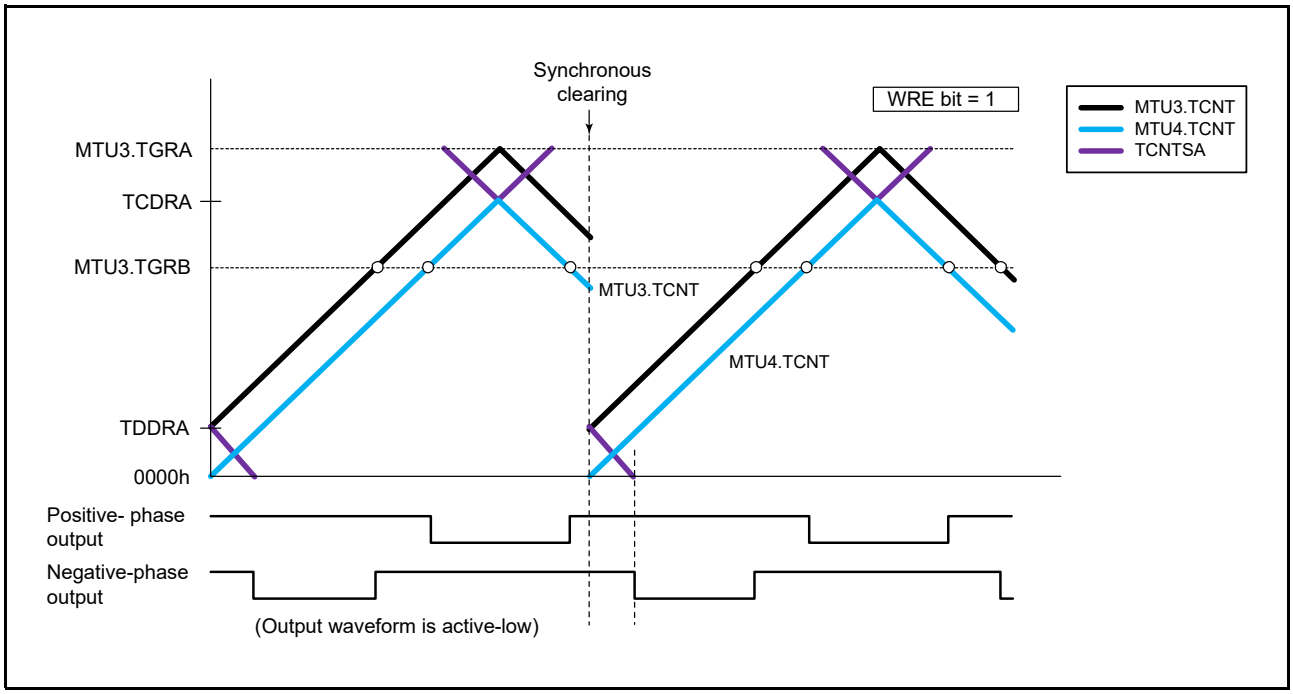


Figure 22.69 Example of Synchronous Clearing in Dead Time during Down-Counting (Timing (8) in Figure 22.65; TWCRA.WRE Bit is 1)

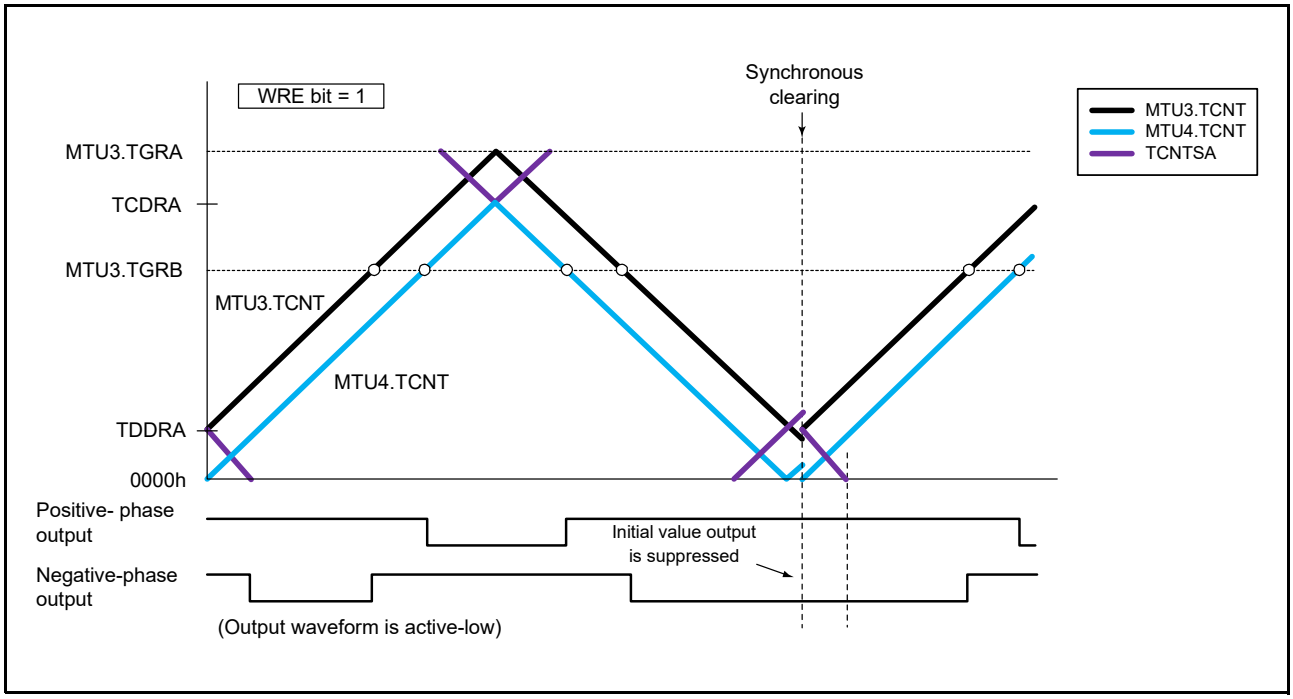


Figure 22.70 Example of Synchronous Clearing in Tb2 interval (Timing (11) in Figure 22.65; TWCRA.WRE Bit is 1)

(o) Suppressing Synchronous Counter Clearing for MTU0 to MTU2, and MTU6 and MTU7

In MTU6 and MTU7, setting the SCC bit in TWCRB to 1 suppresses synchronous counter clearing caused by MTU0 to MTU2.

Synchronous counter clearing is suppressed only within the interval shown in Figure 22.71. When using this function, MTU6 and MTU7 should be set to complementary PWM mode.

For details of synchronous clearing caused by MTU0 to MTU2, refer to section 22.3.10 (2), Synchronous Counter Clearing for MTU6 and MTU7.

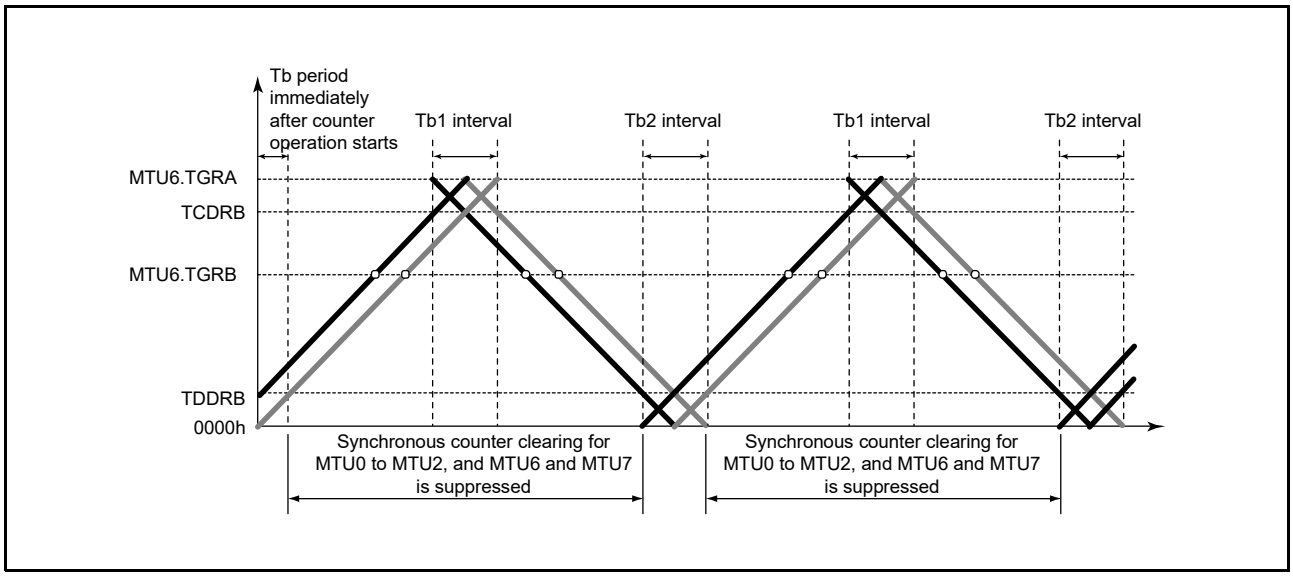


Figure 22.71 Synchronous Clearing-Suppressed Interval Specified by TWCRB.SCC Bit for MTU0 to MTU2, and MTU6 and MTU7

- Example of Procedure for Suppressing Synchronous Counter Clearing for MTU0 to MTU2, and MTU6 and MTU7

An example of the procedure for suppressing synchronous counter clearing for MTU0 to MTU2, and MTU6 and MTU7 is shown in Figure 22.72.

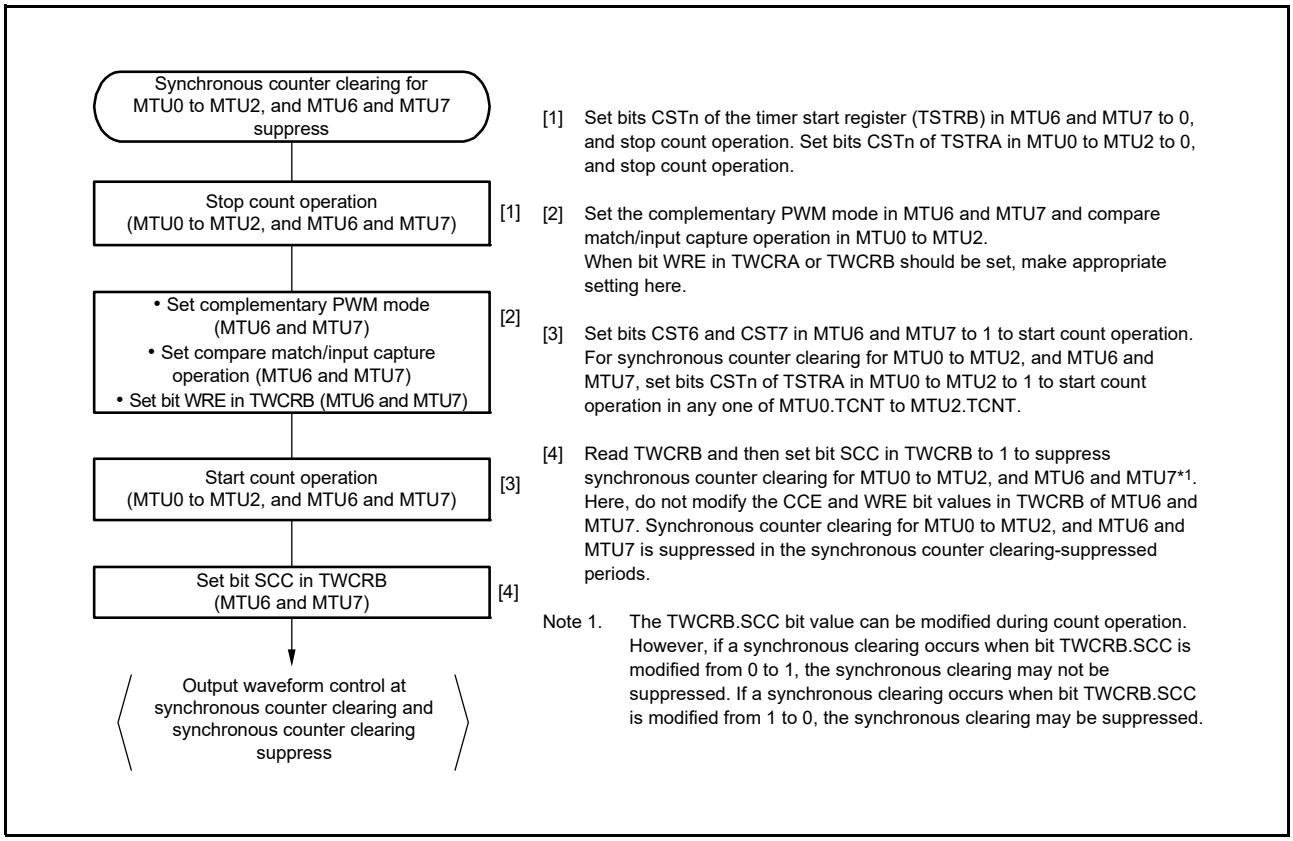


Figure 22.72 Example of Procedure for Suppressing Synchronous Counter Clearing for MTU0 to MTU2, and MTU6 and MTU7

- Examples of Suppression of Synchronous Counter Clearing for MTU 0 to MTU2, and MTU6 and MTU7

Figure 22.73 to Figure 22.76 show examples of operation in which MTU6 and MTU7 operate in complementary PWM mode and synchronous counter clearing for MTU 0 to MTU2, and MTU6 and MTU7 is suppressed by setting the SCC bit in TWCRB in MTU6 and MTU7 to 1. In the examples shown in Figure 22.73 to Figure 22.76, synchronous counter clearing occurs at timing (3), (6), (8), and (11) shown in Figure 22.65, respectively. In these examples, the WRE bit in TWCRB in MTU6 and MTU7 is set to 1.

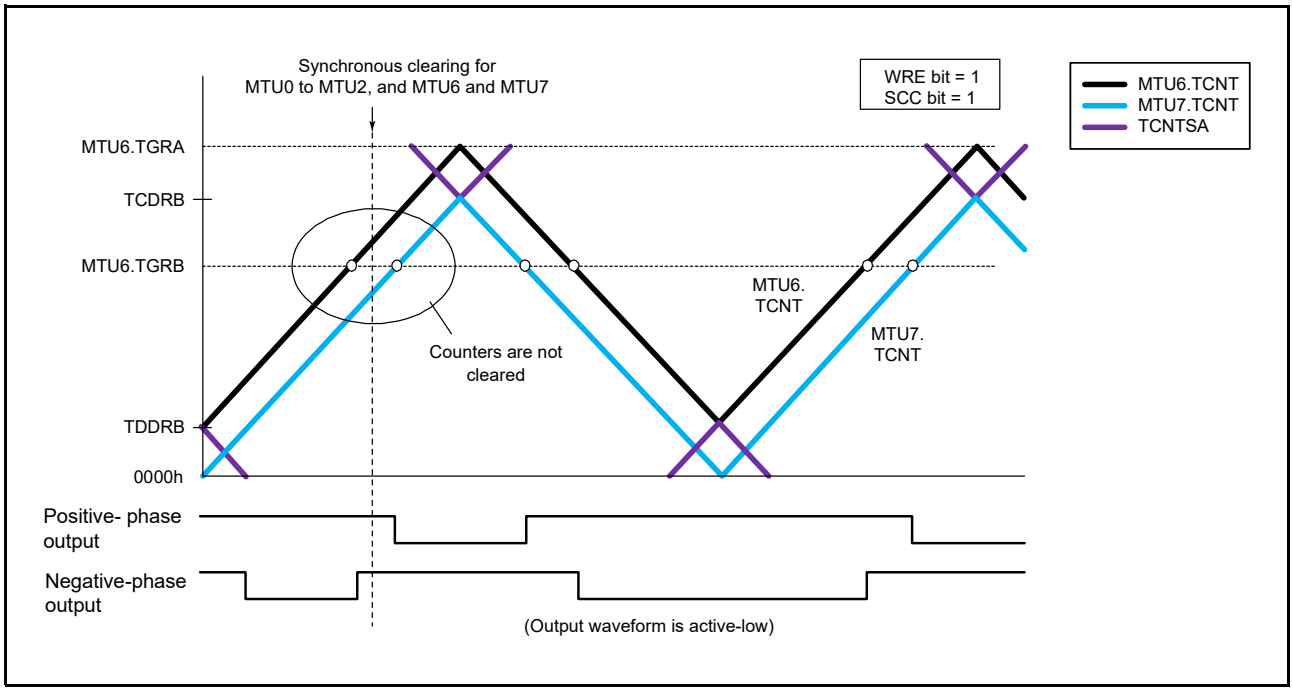


Figure 22.73 Example of Synchronous Clearing in Dead Time during Up-Counting (Timing (3) in Figure 22.65; TWCRB.WRE Bit is 1 and SCC Bit is 1 in MTU6 and MTU7)

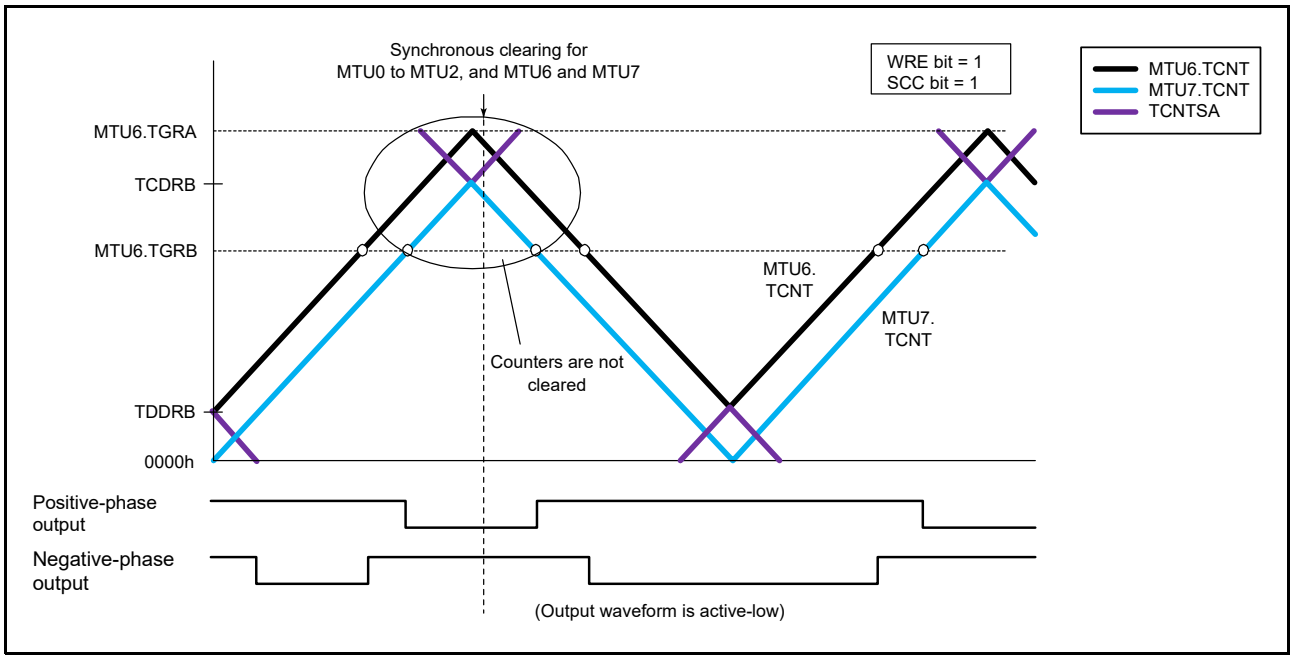


Figure 22.74 Example of Synchronous Clearing in Tb1 interval (Timing (6) in Figure 22.65; TWCRB.WRE Bit is 1 and SCC Bit is 1 in MTU6 and MTU7)

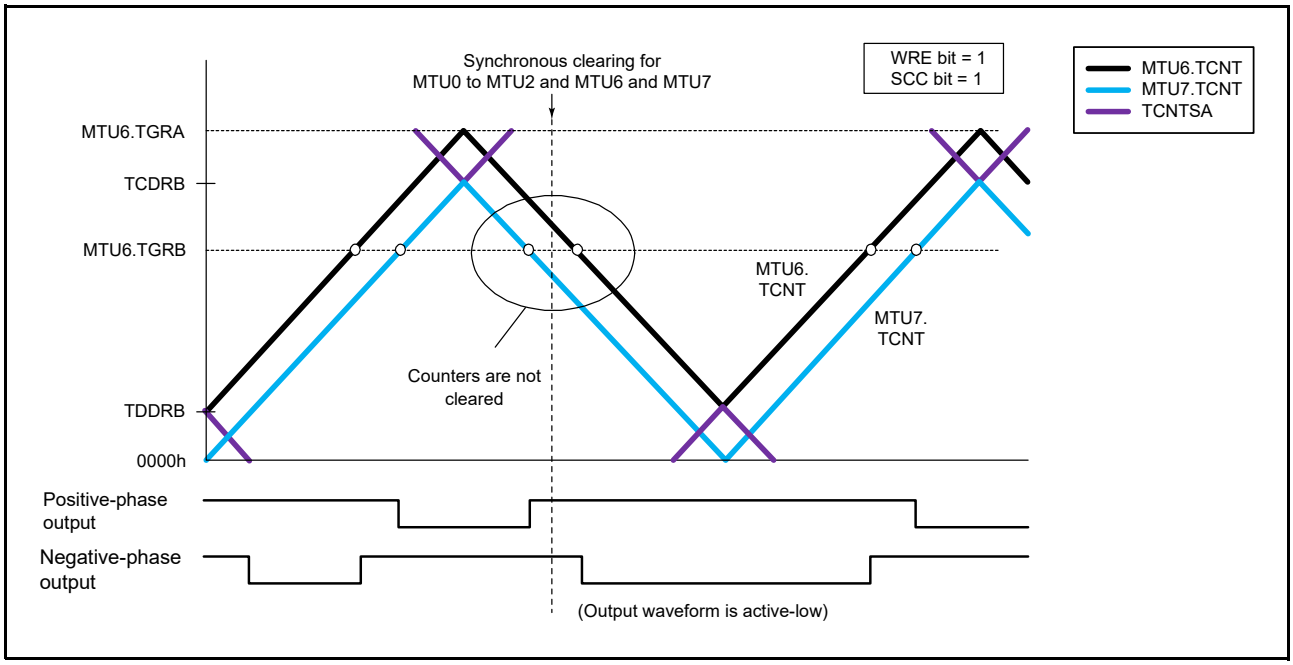


Figure 22.75 Example of Synchronous Clearing in Dead Time during Down-Counting (Timing (8) in Figure 22.65; TWCRB.WRE Bit is 1 and SCC Bit is 1 in MTU6 and MTU7)

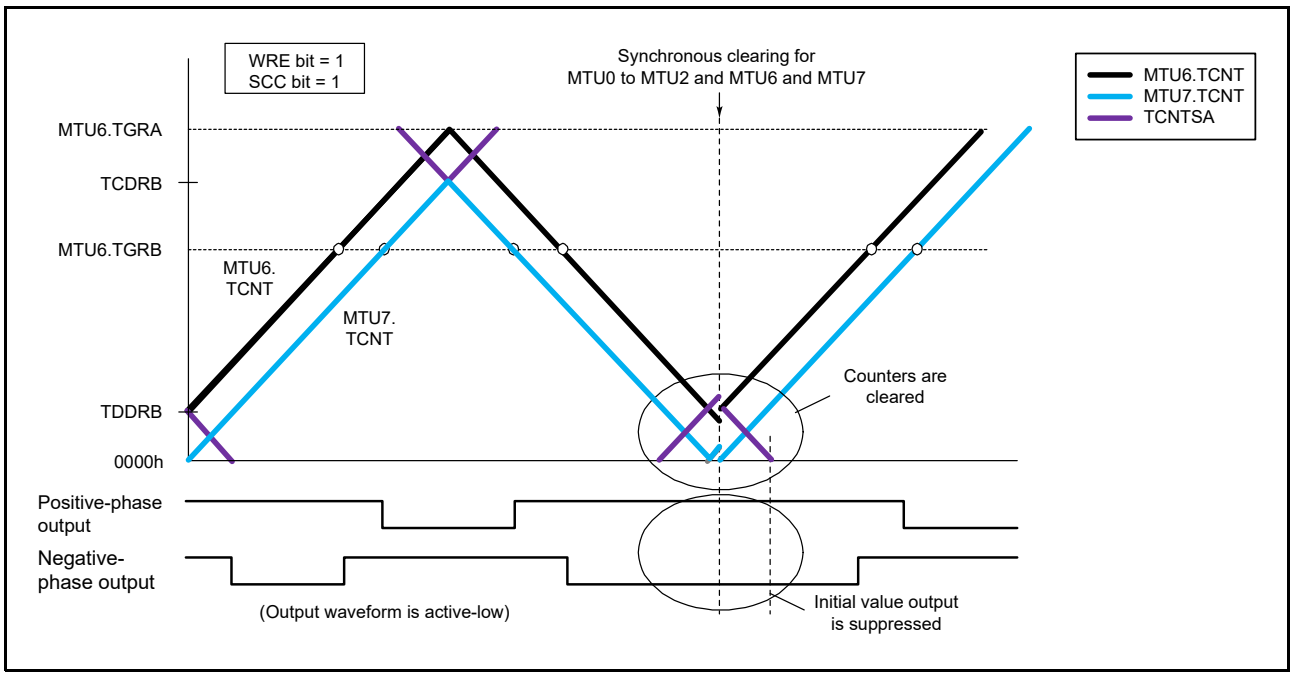


Figure 22.76 Example of Synchronous Clearing in Tb2 interval (Timing (11) in Figure 22.65; TWCRB.WRE Bit is 1 and SCC Bit is 1 in MTU6 and MTU7)

(p) Counter Clearing by MTU3.TGRA (MTU6.TGRA) Compare Match

In complementary PWM mode, MTU3.TCNT, MTU4.TCNT, and TCNTSA (MTU6.TCNT, MTU7.TCNT, and TCNTSB) can be cleared by MTU3.TGRA (MTU6.TGRA) compare match when the TWCRA.CCE (TWCRB.CCE) bit. Figure 22.77 illustrates an operation example.

Note 1. Use this function only in complementary PWM mode 1 (transfer at crest).
Note 2. Do not specify synchronous clearing by another channel (do not set 1 in the SYNC0 to SYNC4, SYNC9, or SYNC6 and SYNC7 bits in the timer synchronous register (TSYRA or TSYRB) and the CE0A to CE0D, CE1A, CE1B, CE2A, or CE2B bits in TSYCR).
Note 3. Do not set the PWM duty value to 0000h.
Note 4. Do not set the PSYE bit in timer output control register 1 (TOCR1A or TOCR1B) to 1.

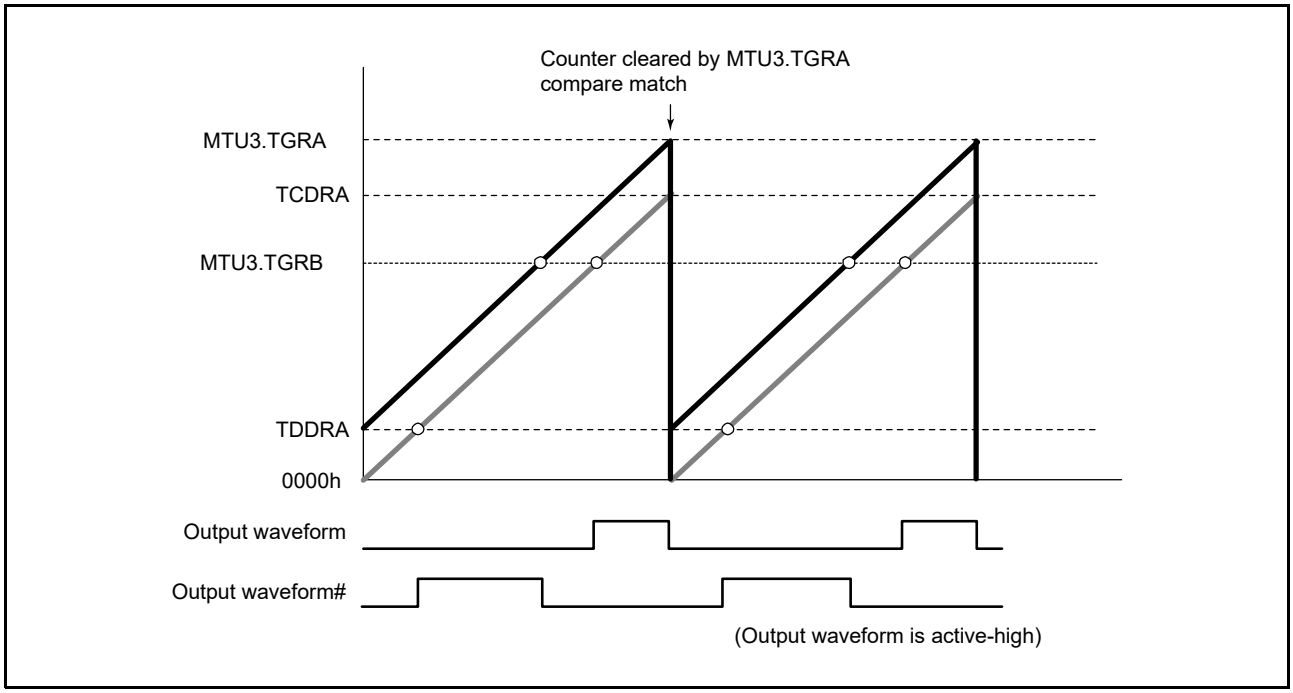


Figure 22.77 Example of Counter Clearing Operation by MTU3.TGRA Compare Match

(q) Example of Waveform Output for Driving AC Synchronous Motor (Brushless DC Motor)

In complementary PWM mode, a brushless DC motor can easily be controlled using the TGCRA (TGCRB) register. Figure 22.78 to Figure 22.81 show examples of brushless DC motor driving waveforms when MTU3 and MTU4 are used.

The TGCRB register enables driving waveform output from MTU6 and MTU7. The MTIOC9A, MTIOC9B, and MTIOC9C pins are used for external signals.

To switch the output phases for a 3-phase brushless DC motor by means of external signals detected with a Hall element, etc., set the TGCRA.FB (TGCRB.FB) bit to 0. In this case, the external signals indicating the magnetic pole position should be input to timer input pins MTIOC0A, MTIOC0B, and MTIOC0C (MTIOC9A, MTIOC9B, and MTIOC9C) in MTU0 (MTU9) (make appropriate settings with the MPC and port mode registers (PMR) of the I/O ports). When an edge is detected at pin MTIOC0A, MTIOC0B, or MTIOC0C (MTIOC9A, MTIOC9B, and MTIOC9C), the output on/off state is switched automatically.

When the TGCRA.FB (TGCRB.FB) bit is 1, the output on/off state is switched when the UF, VF, or WF bit in TGCRA (TGCRB) is set to 0 or 1.

The driving waveforms are output from the 6-phase PWM output pins for complementary PWM mode.

With this 6-phase output, while the output is turned on, chopping output is available through complementary PWM mode output function by setting the N bit or P bit in TGCRA (TGCRB) to 1. When the N bit or P bit is 0, the level output is selected.

The active level of the 6-phase output (on output level) can be set with the TOCR1A.OLSN (TOCR1B.OLSN) and TOCR1A.OLSP (TOCR1B.OLSP) bits regardless of the setting of the N and P bits.

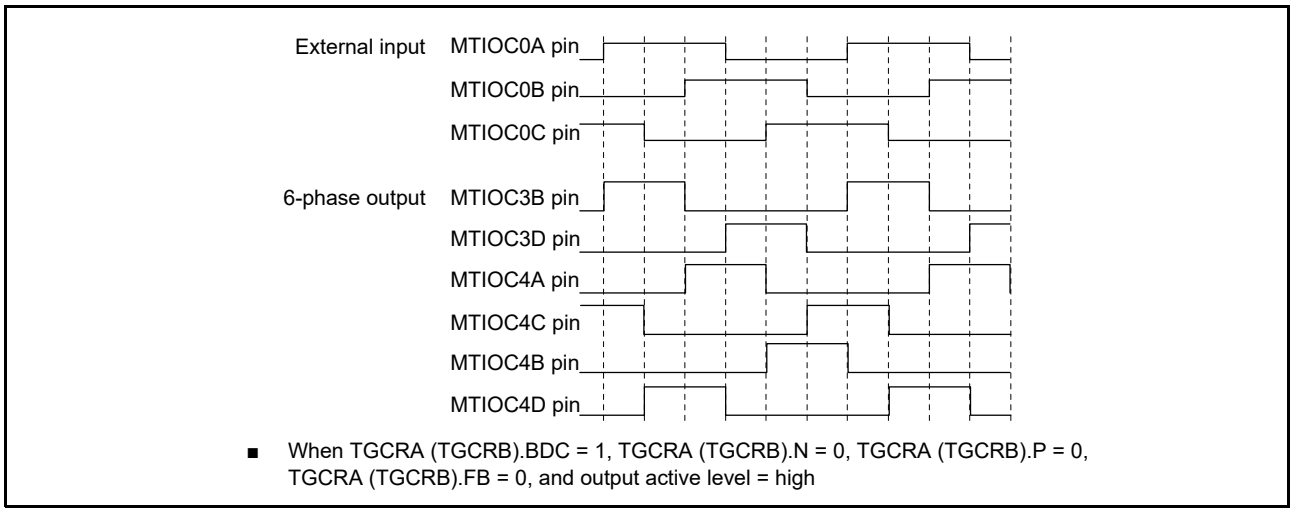
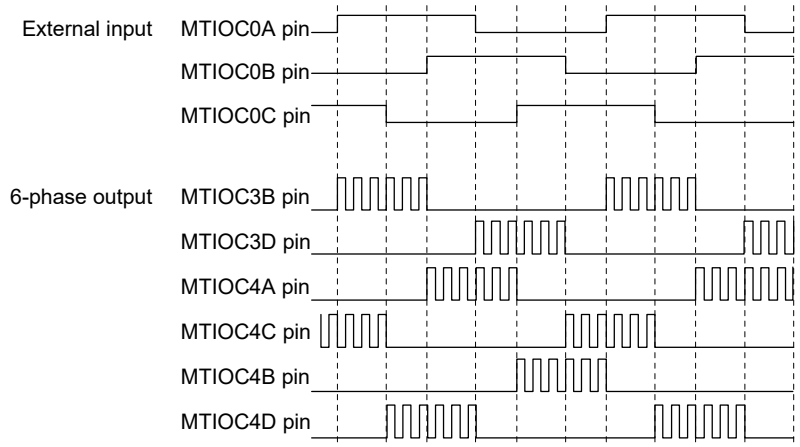
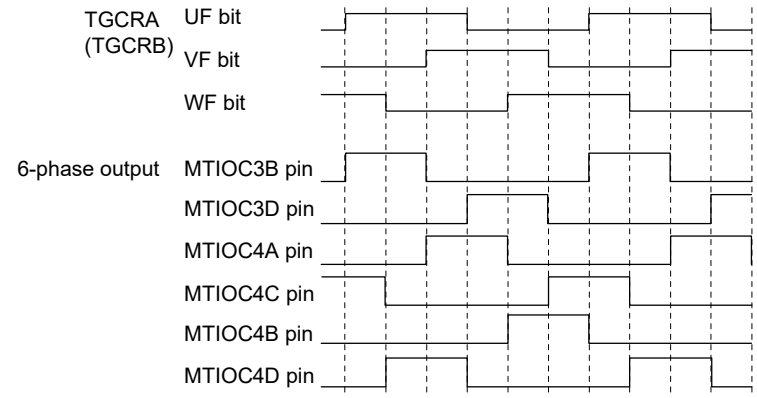


Figure 22.78 Example of Output Phase Switching by External Input (1)



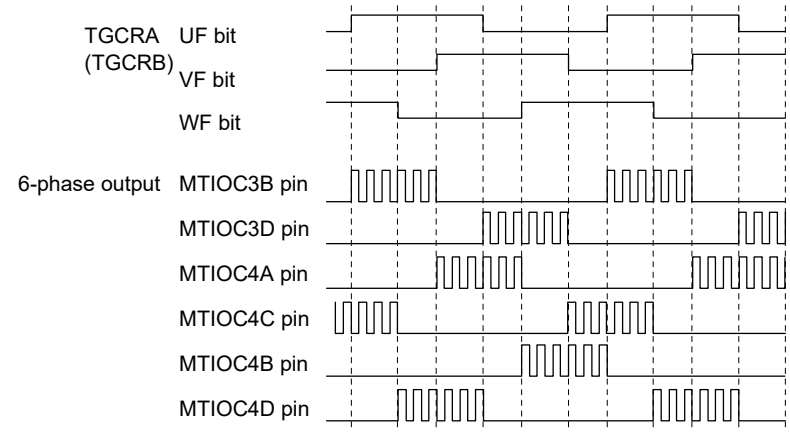
- When TGCRA (TGCRB).BDC = 1, TGCRA (TGCRB).N = 1, TGCRA (TGCRB).P = 1, TGCRA (TGCRB).FB = 0, and output active level = high

Figure 22.79 Example of Output Phase Switching by External Input (2)



- When TGCRA (TGCRB).BDC = 1, TGCRA (TGCRB).N = 0, TGCRA (TGCRB).P = 0, TGCRA (TGCRB).FB = 1, and output active level = high

Figure 22.80 Example of Output Phase Switching through UF, VF, and WF Bit Settings (1)



- When TGCRA (TGCRB).BDC = 1, TGCRA (TGCRB).N = 1, TGCRA (TGCRB).P = 1, TGCRA (TGCRB).FB = 1, and output active level = high

Figure 22.81 Example of Output Phase Switching through UF, VF, and WF Bit Settings (2)

(r) A/D Conversion Start Request Setting

In complementary PWM mode, an A/D conversion start request can be issued using MTU3.TGRA (MTU6.TGRA) compare match, MTU4.TCNT (MTU7.TCNT) underflow (trough), or compare match on a channel other than MTU3 and MTU4 (MTU6 and MTU7).
When start requests using MTU3.TGRA (MTU6.TGRA) compare match are specified, A/D conversion can be started at the crest of the MTU3.TCNT (MTU6.TCNT) count.
A/D conversion start requests can be specified by setting the TIER.TTGE bit. To issue an A/D conversion start request at an MTU4.TCNT (MTU7.TCNT) underflow (trough), set the MTU4.TIER.TTGE2 (MTU7.TIER.TTGE2) bit to 1.

(s) Double Buffer Function in Complementary PWM Mode

In complementary PWM mode 3 (transfer at the crest and trough), the PWM output setting resolution can be improved from ±2 to ±1 by setting the TMDR2A.DRS (TMDR2B.DRS) bit to 1.
When setting buffer registers A (MTU3.TGRD, MTU4.TGRC, MTU4.TGRD, MTU6.TGRD, MTU7.TGRC, and MTU7.TGRD), set also buffer registers B (MTU3.TGRE, MTU4.TGRE, MTU4.TGRF, MTU6.TGRE, MTU7.TGRE, and MTU7.TGRF) at the same time. For details of the setting procedure, refer to section 22.3.8 (1), Example of Complementary PWM Mode Setting Procedure

Note: When a buffer register B is set to the buffer register A value, symmetric PWM waveforms are output. When a buffer register B is not set to the buffer register A value, asymmetric PWM waveforms are output.

Figure 22.82 shows an example of double buffer operation.
Each register data is transferred as follows.

- After MTU4.TGRD or MTU7.TGRD (buffer A) is written to, data is transferred from MTU4.TGRD or MTU7.TGRD (buffer A) to TEMP3A or TEMP6A (temporary A) and from MTU4.TGRF or MTU7.TGRF (buffer B) to TEMP3B or TEMP6B (temporary B).
- With timing (1) in the figure, data is transferred from TEMP3A or TEMP6A (temporary A) to MTU4.TGRB or MTU7.TGRB (compare).
- With timing (2) in the figure, data is transferred from TEMP3B or TEMP6B (temporary B) to MTU4.TGRB or MTU7.TGRB (compare).

In the crest interval (Tb1 interval), the compare register and temporary register A are valid; in the trough interval (Tb2 interval), the compare register and temporary register B are valid.

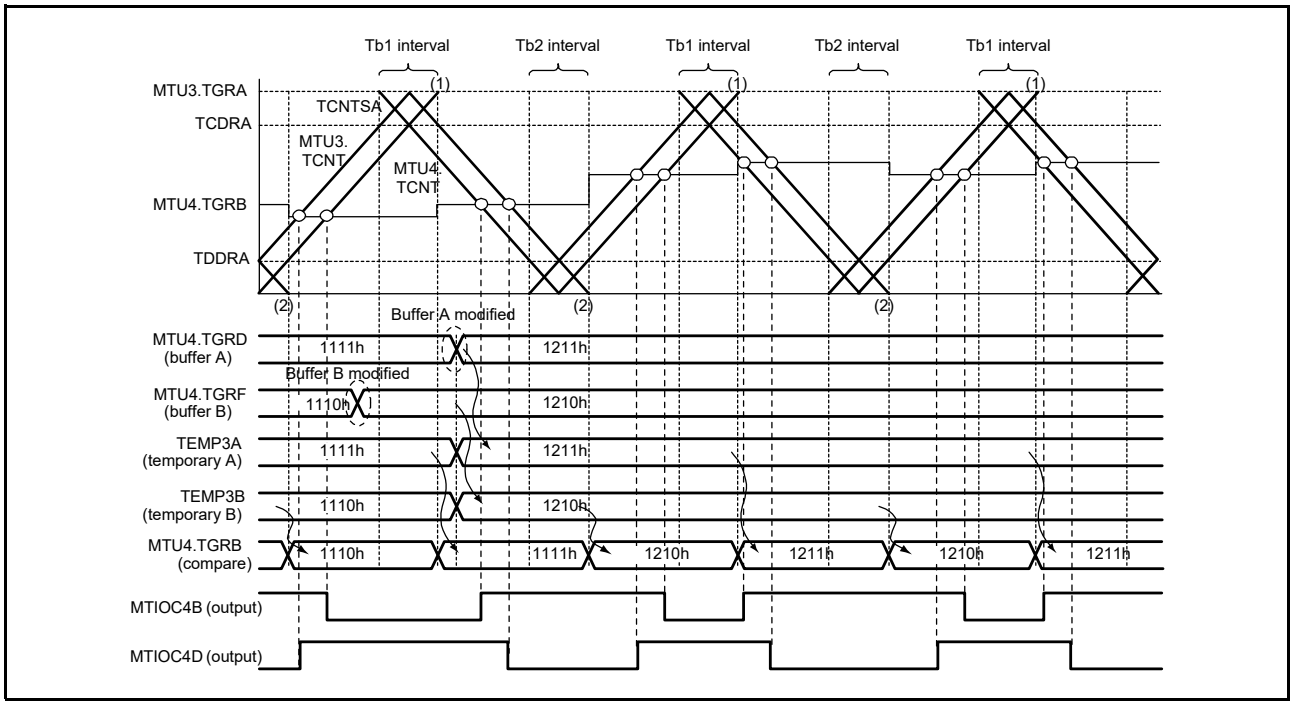


Figure 22.82 Example of Double Buffer Operation

Figure 22.83 shows an example when the buffer write value is smaller than the TDDRA (TDDRB) value, and Figure 22.84 shows an example when the write value is greater than TCDRA (TCDRB).
In the crest interval, the output is controlled according to the compare match with the compare register or temporary register A; in the trough interval, the output is controlled according to the compare match with the compare register or temporary register B.

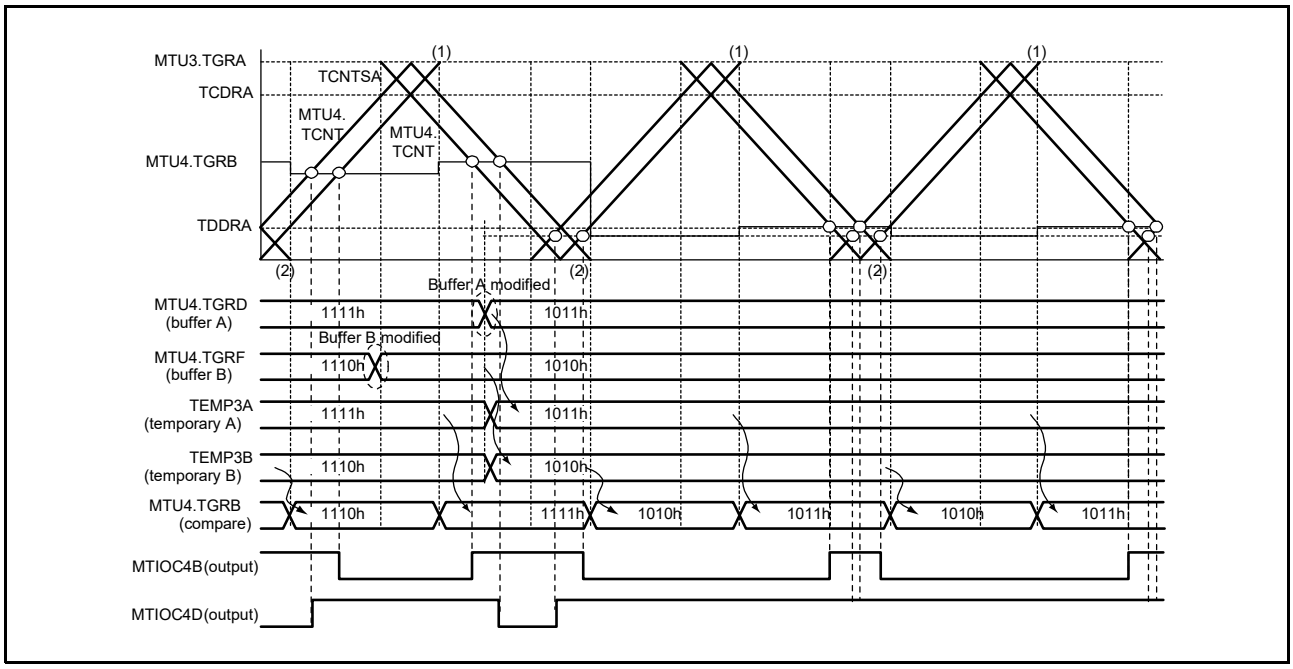


Figure 22.83 Example of Double Buffer Operation (Buffer Write Value is Smaller than TDDRA)

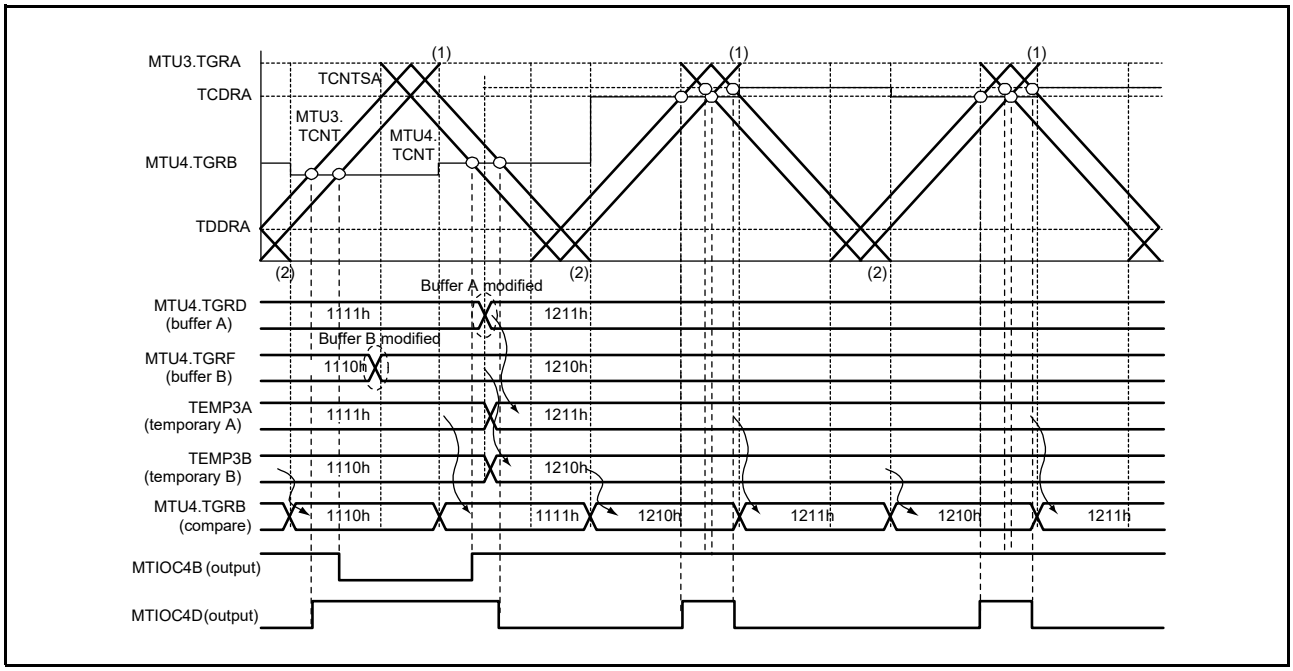


Figure 22.84 Example of Double Buffer Operation (Buffer Write Value is Greater than TCDRA)

(3) Interrupt Skipping Function 1 in Complementary PWM Mode

Interrupts TGIA3 (TGIA6) (at the crest) and TCIV4 (TCIV7) (at the trough) in MTU3 and MTU4 (MTU6 and MTU7) can be skipped up to seven times by making settings in the TITCR1A (TITCR1B) register.
Transfers from a buffer register to a temporary register or a compare register can be skipped in coordination with interrupt skipping by making settings in the TBTERA (TBTERB) register. For the linkage with buffer registers, refer to description (c), Buffer Transfer Control Linked with Interrupt Skipping, below.
A/D conversion start requests generated by the A/D conversion start request delaying function can also be skipped in coordination with interrupt skipping by making settings in the MTU4.TADCR (MTU7.TADCR) register. For the linkage with the A/D conversion start request delaying function, refer to section 22.3.9, A/D Conversion Start Request Delaying Function.
The TITCR1A (TITCR1B) register should be set while interrupt skipping function 1 is selected by setting the TITM bit in the timer interrupt skipping mode register (TITMRA or TITMRB) to 0, TGIA3 (TGIA6) interrupt requests are disabled by setting the MTU3.TIER (MTU6.TIER) register, TCIV4 (TCIV7) interrupt requests are disabled by setting the MTU4.TIER (MTU7.TIER) register, and a compare match is not generated. Before changing the skipping count, be sure to set the T3AEN (T6AEN) and T4VEN (T7VEN) bits to 0 to clear the skipping counter.

(a) Example of Interrupt Skipping Function 1 Setting Procedure

Figure 22.85 shows an example of the interrupt skipping function 1 setting procedure. Figure 22.86 shows the periods during which interrupt skipping count can be changed.

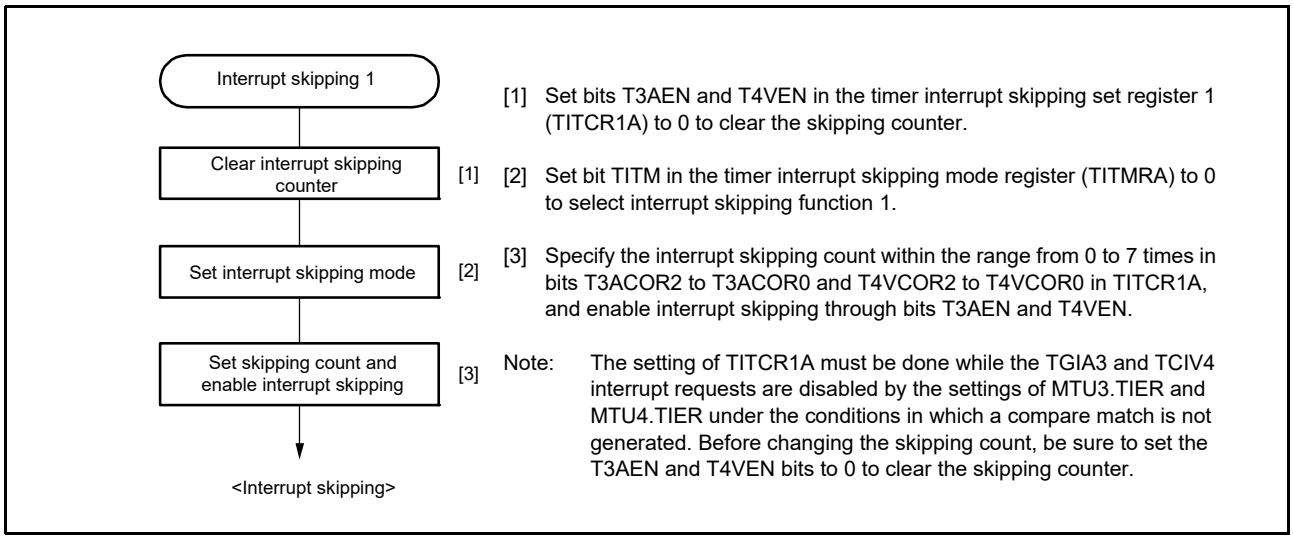


Figure 22.85 Example of Interrupt Skipping Function 1 Setting Procedure

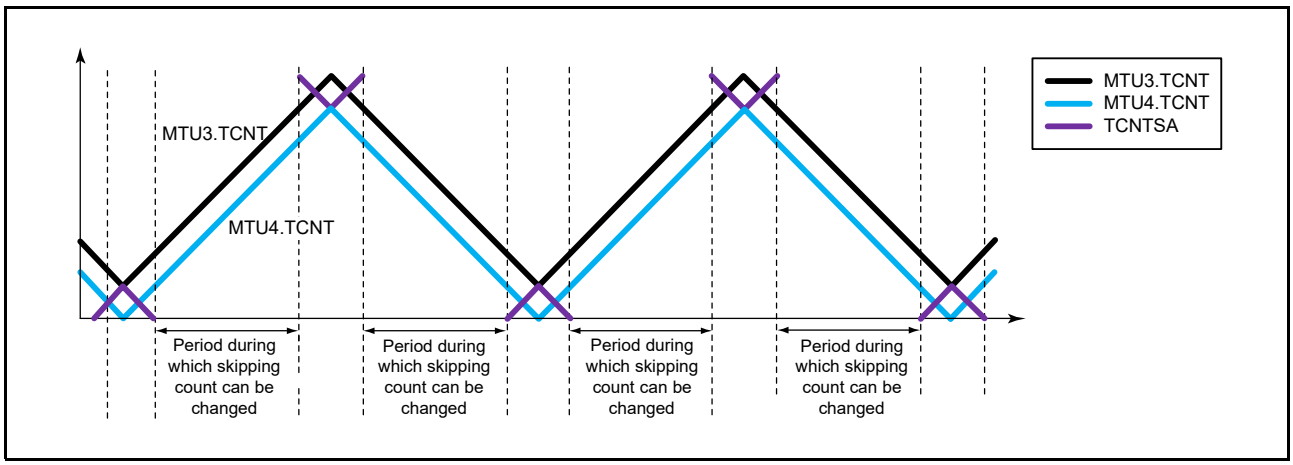


Figure 22.86 Periods during which Interrupt Skipping Count can be Changed

(b) Example of Interrupt Skipping Function 1

Figure 22.87 shows an example of TGIA3 (TGIA6) interrupt skipping in which the interrupt skipping count is set to three by the T3ACOR (T6ACOR) bits and the T3AEN (T6AEN) bit is set to 1 in the TITCR1A (TITCR1B) register.

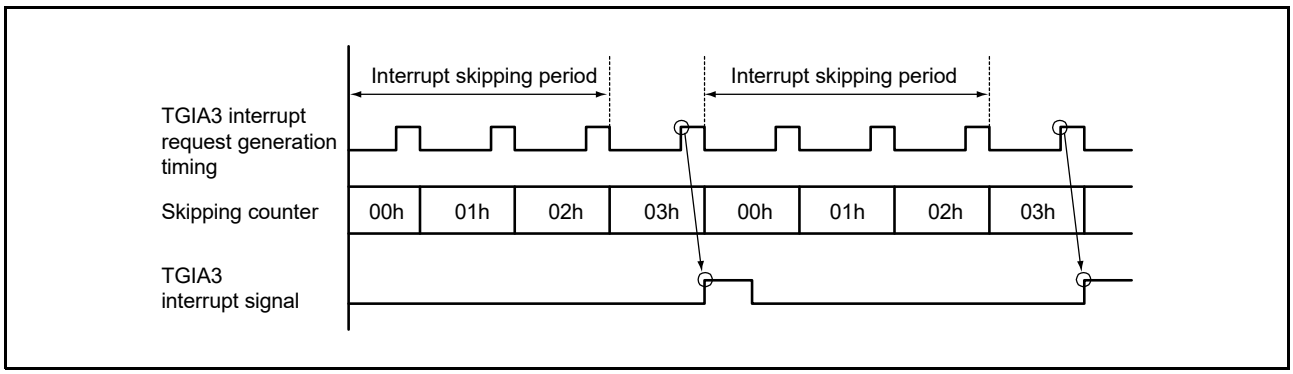
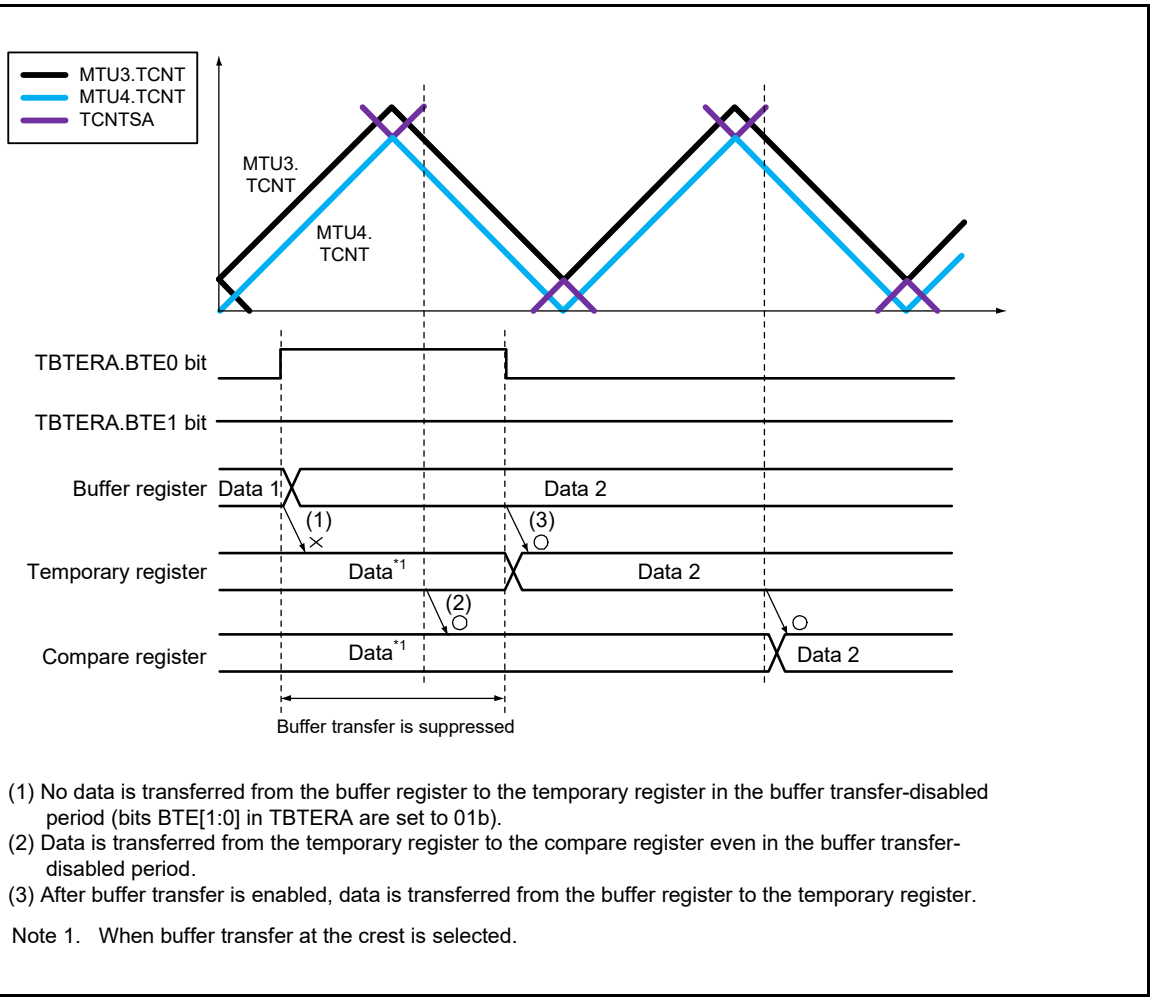


Figure 22.87 Example of Interrupt Skipping Function 1

(c) Buffer Transfer Control Linked with Interrupt Skipping

In complementary PWM mode, whether to transfer data from a buffer register to a temporary register and whether to link the transfer with interrupt skipping can be specified with the BTE[1:0] bits in the TBTERA (TBTERB) register.
Figure 22.88 shows an example of operation when buffer transfer is disabled (BTE[1:0] = 01b). While this setting is valid, data is not transferred from the buffer register to the temporary register.
Figure 22.89 shows an example of operation when buffer transfer is linked with interrupt skipping (BTE[1:0] = 10b). While this setting is valid, data is not transferred from the buffer register outside the buffer transfer-enabled period. Note that the buffer transfer-enabled period differs depending on whether only the T3AEN (T6AEN) bit in the TITCR1A (TITCR1B) register is set to 1, only the T4VEN (T7VEN) bit in the TITCR1A (TITCR1B) register is set to 1, or both the T3AEN and T4VEN (T6AEN and T7VEN) bits are set to 1. Figure 22.90 shows the relationship between the T3AEN (T6AEN) and T4VEN (T7VEN) bit settings in TITCR1A (TITCR1B) and buffer transfer-enabled period.

Note: This function must be used in combination with interrupt skipping function 1.
When interrupt skipping is disabled (the T3AEN and T4VEN (T6AEN and T7VEN) bits in the timer interrupt skipping set register 1 (TITCR1A or TITCR1B) are set to 0 or the skipping count set bits (T3ACOR and T4VCOR (T6ACOR and T7VCOR)) in TITCR1A (TITCR1B) are set to 0), make sure that buffer transfer is not linked with interrupt skipping (set the BTE1 bit in TBTERA or TBTERB to 0).
If buffer transfer is linked with interrupt skipping while interrupt skipping is disabled, buffer transfer is never performed.



(1) No data is transferred from the buffer register to the temporary register in the buffer transfer-disabled period (bits BTE[1:0] in TBTERA are set to 01b).
(2) Data is transferred from the temporary register to the compare register even in the buffer transfer-disabled period.
(3) After buffer transfer is enabled, data is transferred from the buffer register to the temporary register.

Note 1. When buffer transfer at the crest is selected.

Figure 22.88 Example of Operation When Buffer Transfer is Disabled (BTE[1:0] = 01b)

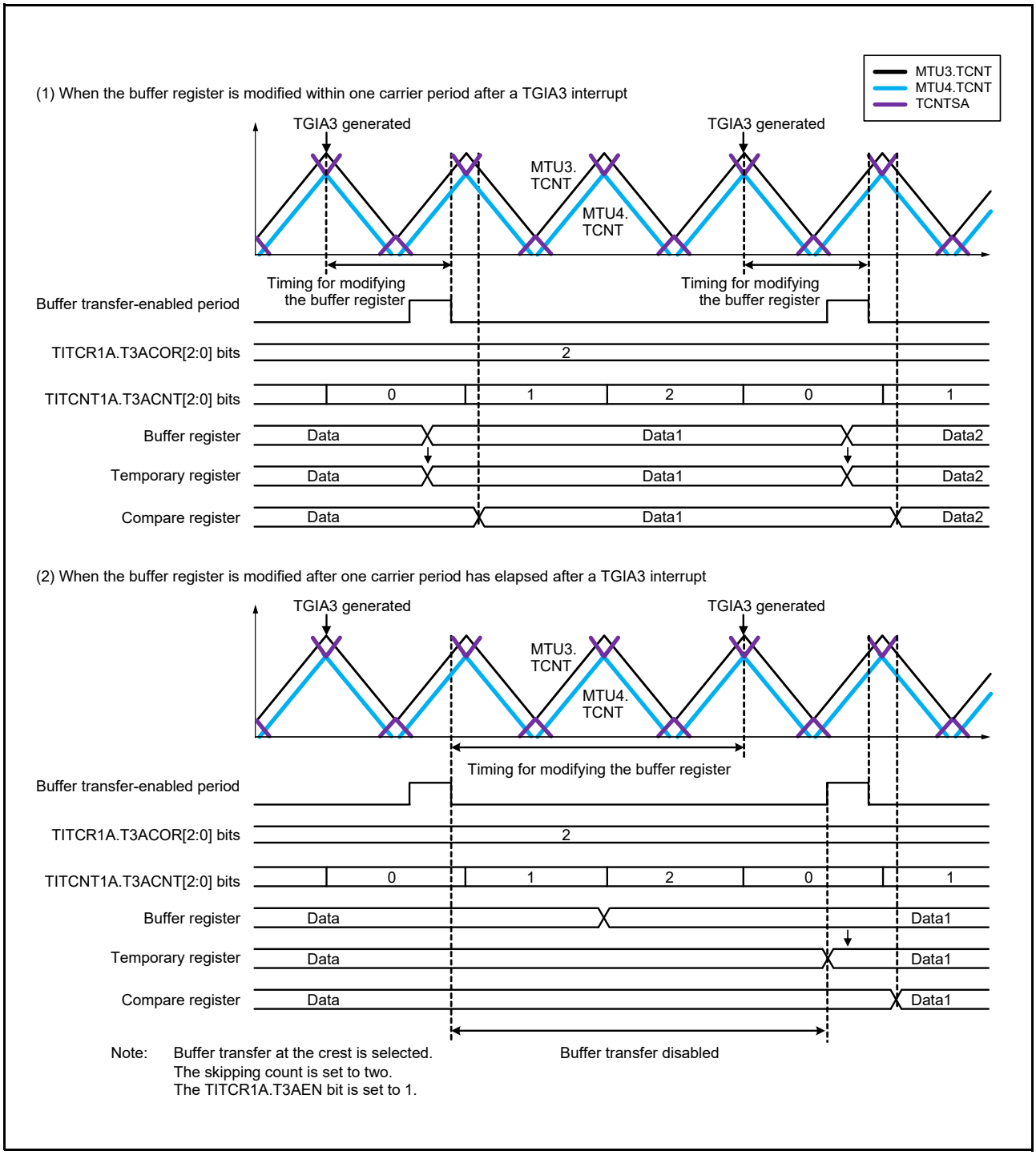


Figure 22.89 Example of Operation When Buffer Transfer is Linked with Interrupt Skipping (BTE[1:0] = 10b)

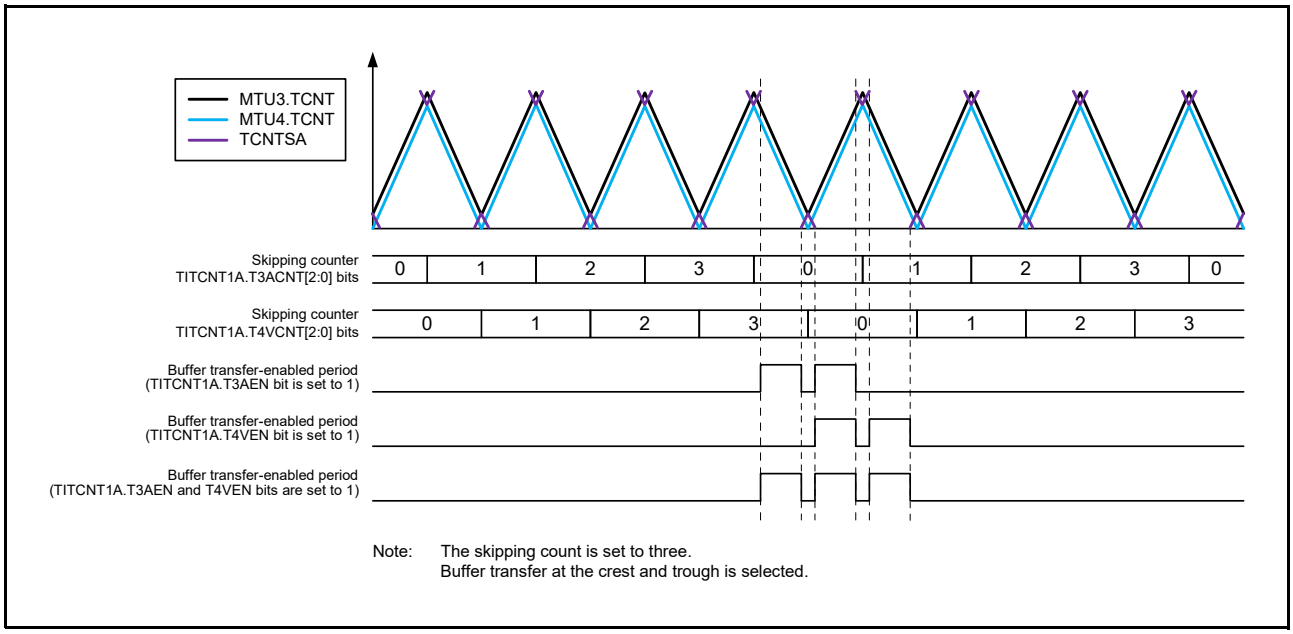


Figure 22.90 Relationship between Bits T3AEN and T4VEN in TITCR1A and Buffer Transfer-Enabled Period

(4) Complementary PWM Mode Output Protection Functions

The following output protection functions are provided for complementary PWM mode.

(a) Register and Counter Miswrite Prevention Function

Access from the CPU to the mode registers, control registers, compare registers, and counters can be enabled or disabled by setting the RWE bit in the TRWERA (TRWERB) register. The applicable registers are some of the registers in MTU3, MTU4, MTU6, and MTU7 shown below:
48 registers in total
MTU3.TCR, MTU4.TCR, MTU3.TCR2, MTU4.TCR2, MTU3.TMDR1, MTU4.TMDR1, MTU3.TIORH, MTU4.TIORH, MTU3.TIORL, MTU4.TIORL, MTU3.TIER, MTU4.TIER, MTU3.TCNT, MTU4.TCNT, MTU3.TGRA, MTU4.TGRA, MTU3.TGRB, MTU4.TGRB, MTU.TOERA, MTU.TOCR1A, MTU.TOCR2A, MTU.TGCRA, MTU.TCDRA, MTU.TDDRA,
MTU6.TCR, MTU7.TCR, MTU6.TCR2, MTU7.TCR2, MTU6.TMDR1, MTU7.TMDR1, MTU6.TIORH, MTU7.TIORH, MTU6.TIORL, MTU7.TIORL, MTU6.TIER, MTU7.TIER, MTU6.TCNT, MTU7.TCNT, MTU6.TGRA, MTU7.TGRA, MTU6.TGRB, MTU7.TGRB, MTU.TOERB, MTU.TOCR1B, MTU.TOCR2B, MTU.TGCRB, MTU.TCDRB, and MTU.TDDRB
This function can disable CPU access to the mode registers, control registers, and counters to prevent miswriting due to CPU runaway. In the access-disabled state, the applicable registers are read as undefined and writing to these registers is ignored.

(b) Halting of PWM Output by External Signal

The PWM output pins of MTU0, MTU3, MTU4, MTU6, MTU7, and MTU9 can be set to the high-impedance state automatically.
Refer to section 23, Port Output Enable 3 (POE3D), for details.

22.3.9 A/D Conversion Start Request Delaying Function

A/D conversion start requests can be issued in MTU4 or MTU7 by making settings in the timer A/D conversion start request control register (MTU4.TADCR or MTU7.TADCR), timer A/D conversion start request cycle set registers (MTU4.TADCORA and MTU4.TADCORB, or MTU7.TADCORA and MTU7.TADCORB), and timer A/D conversion start request cycle set buffer registers (MTU4.TADCOBRA and MTU4.TADCOBRB, or MTU7.TADCOBRA and MTU7.TADCOBRB).

The A/D conversion start request delaying function compares MTU4.TCNT with MTU4.TADCORA or MTU4.TADCORB (MTU7.TCNT with MTU7.TADCORA or MTU7.TADCORB), and when their values match, the function issues a respective A/D conversion start request (TRG4AN or TRG4BN (TRG7AN or TRG7BN)).

A/D conversion start requests (TRG4AN and TRG4BN (TRG7AN and TRG7BN)) can be skipped in coordination with interrupt skipping by making settings in the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in MTU4.TADCR (the ITA6AE, ITA7VE, ITB6AE, and ITB7VE bits in MTU7.TADCR).

(1) Example of Procedure for Specifying A/D Conversion Start Request Delaying Function

Figure 22.91 shows an example of procedure for specifying the A/D conversion start request delaying function.

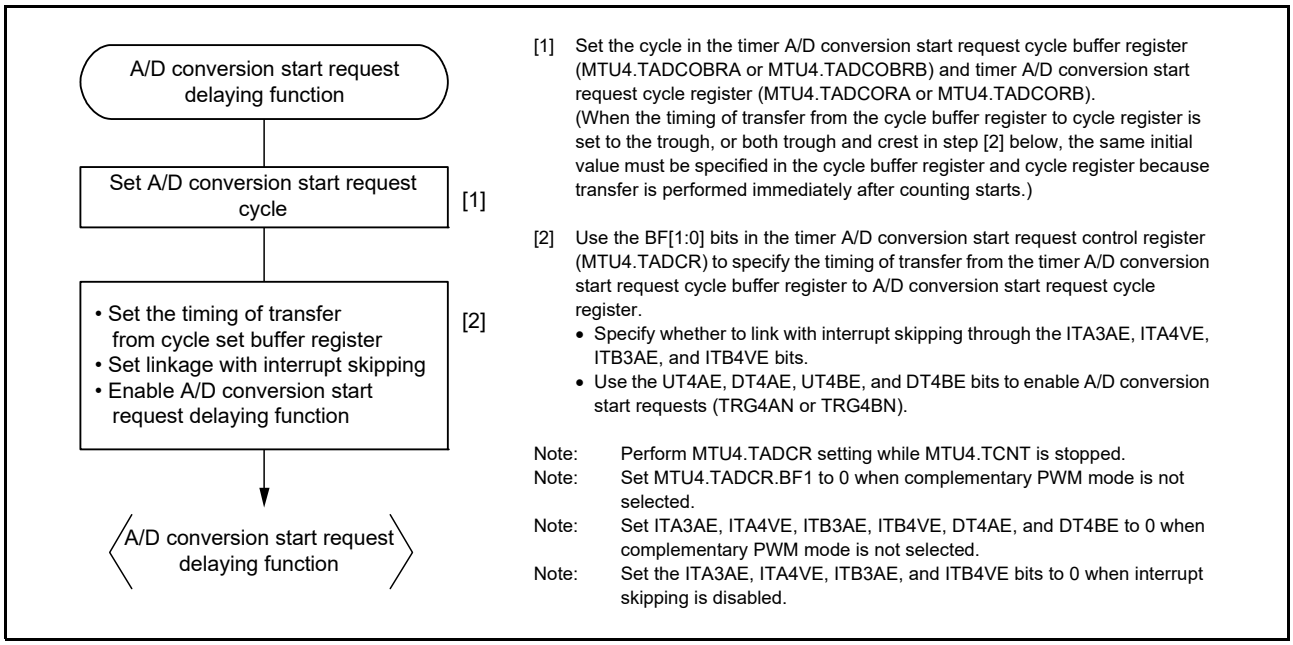
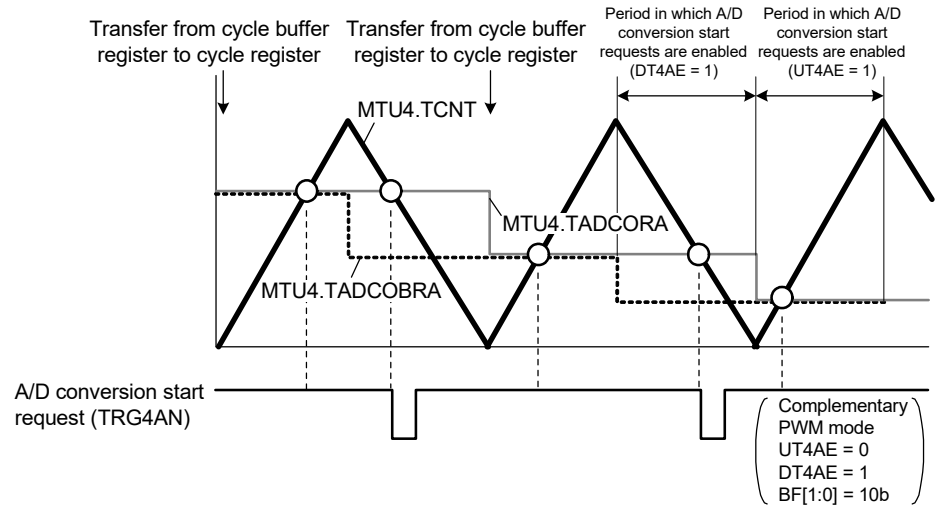


Figure 22.91 Example of Procedure for Specifying A/D Conversion Start Request Delaying Function (MTU3 and MTU4)

(2) Basic Example of A/D Conversion Start Request Delaying Function Operation

Figure 22.92 shows a basic example of A/D conversion start request signal (TRG4AN (TRG7AN)) operation when the trough of MTU4.TCNT (MTU7.TCNT) is specified for the buffer transfer timing and an A/D conversion start request signal is output during MTU4.TCNT (MTU7.TCNT) down-counting.



Period in which A/D conversion start requests are enabled (UT4AE = 1): 0 ≤ MTU4.TCNT ≤ TCDR – 1
Period in which A/D conversion start requests are enabled (DT4AE = 1): TCDR ≥ MTU4.TCNT ≥ 1

UT4AE, DT4AE, BF[1:0]: Bits in TADCR

Figure 22.92 Basic Example of A/D Conversion Start Request Signal (TRG4AN) Operation

(3) Period in Which A/D Conversion Start Requests are Enabled

When the MTU4.TCNT (MTU7.TCNT) counter and the MTU4.TADCORA or MTU4.TADCORB (MTU7.TADCORA or MTU7.TADCORB) register matches within the period enabled by the UT4AE and UT4BE (UT7AE and UT7BE) bits, the corresponding A/D conversion start request (TRG4AN or TRG4BN) is issued.
When the UT4AE and UT4BE (UT7AE and UT7BE) bits in the MTU4.TADCR (MTU7.TADCR) register are set to 1 in complementary PWM mode, A/D conversion start requests are enabled during the MTU4.TCNT (MTU7.TCNT) up-counting (0 ≤ MTU4.TCNT (MTU7.TCNT) ≤ TCDR – 1). When the DT4AE and DT4BE (DT7AE and DT7BE) bits in the MTU4.TADCR (MTU7.TADCR) register are set to 1, A/D conversion start requests are enabled during MTU4.TCNT (MTU7.TCNT) down-counting (TCDR ≥ MTU4.TCNT (MTU7.TCNT) ≥ 1). Refer to Figure 22.92.

(4) Buffer Transfer

The data in the timer A/D conversion start request cycle set registers (MTU4.TADCORA and MTU4.TADCORB, or MTU7.TADCORA and MTU7.TADCORB) is updated by writing data to the timer A/D conversion start request cycle set buffer registers (MTU4.TADCOBRA and MTU4.TADCOBRB, or MTU7.TADCOBRA and MTU7.TADCOBRB). Data is transferred from the buffer registers to the respective cycle set registers at the timing selected with the BF[1:0] bits in the MTU4.TADCR (MTU7.TADCR) register.

In complementary PWM mode, data is also transferred from the timer A/D conversion start request cycle set buffer registers to the timer A/D conversion start request cycle set registers when MTU4.TGRD (MTU7.TGRD) register is updated.

There are notes on the timing for transferring data when using buffer transfer in complementary PWM mode.

For details, section 22.6.28, Usage Notes on A/D Conversion Start Request Delaying Function in Complementary PWM Mode.

In modes other than complementary PWM mode, set the BF1 bit in the MTU4.TADCR (MTU7.TADCR) register to 0.

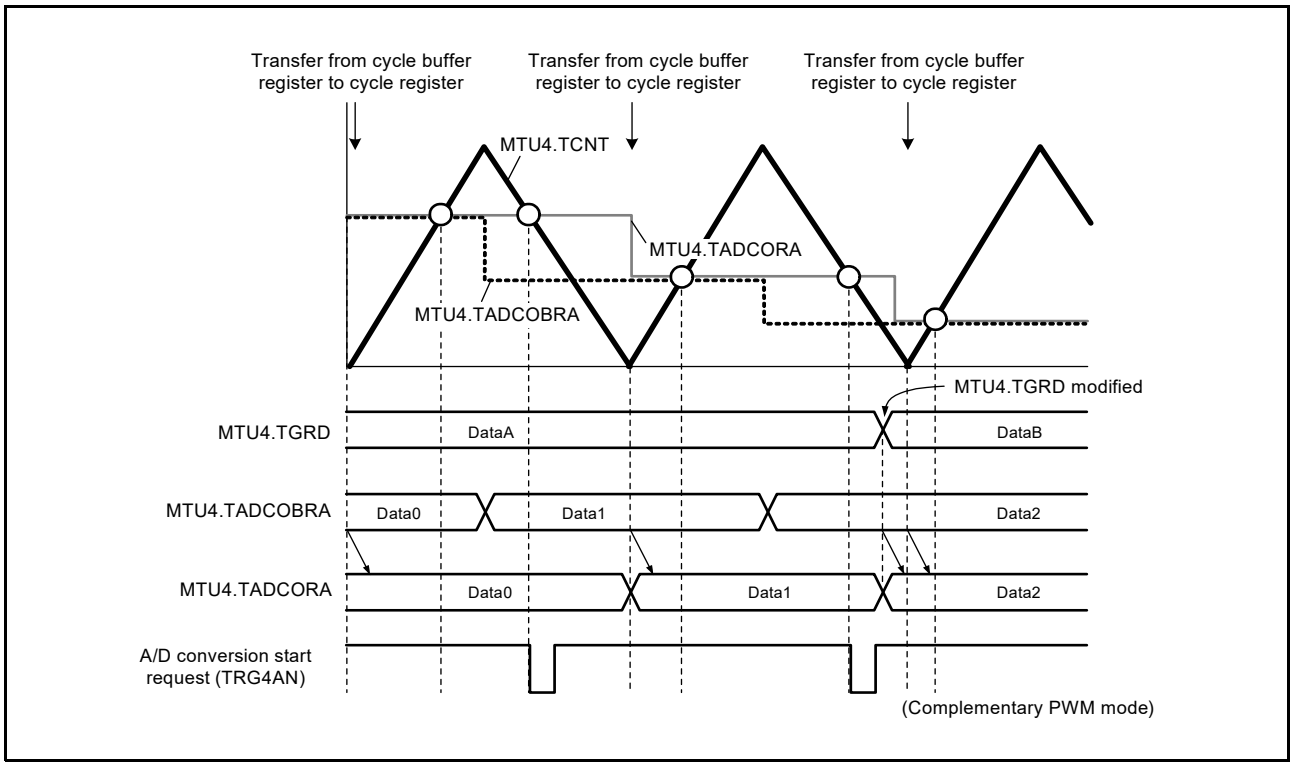


Figure 22.93 Example of A/D Conversion Start Request Signal (TRG4AN) and Buffer Transfer Operation

(5) A/D Conversion Start Request Delaying Function Linked with Interrupt Skipping Function 1

In complementary PWM mode, A/D conversion start requests (TRG4AN and TRG4BN (TRG7AN and TRG7BN)) can be issued in coordination with interrupt skipping 1 by making settings in the ITA3AE, ITA4VE, ITB3AE, and ITB4VE (ITA6AE, ITA7VE, ITB6AE, and ITB7VE) bits in the MTU4.TADCR (MTU7.TADCR) register.

Figure 22.94 shows an example of A/D conversion start request signal (TRG4AN (TRG7AN)) operation when TRG4AN (TRG7AN) output is enabled during MTU4.TCNT (MTU7.TCNT) up-counting and down-counting and A/D conversion start requests are linked with interrupt skipping 1.

Figure 22.95 shows another example of A/D conversion start request signal (TRG4AN (TRG7AN)) operation when TRG4AN (TRG7AN) output is enabled during MTU4.TCNT (MTU7.TCNT) up-counting and A/D conversion start requests are linked with interrupt skipping 1.

In modes other than complementary PWM mode, do not use the A/D conversion start request delaying function linked with the interrupt skipping function 1.

Set the ITA3AE, ITA4VE, ITB3AE, and ITB4VE (ITA6AE, ITA7VE, ITB6AE, and ITB7VE) bits in the MTU4.TADCR (MTU7.TADCR) register to 0.

Note: This function should be used in combination with interrupt skipping 1.
When interrupt skipping is disabled (the T3AEN and T4VEN (T6AEN and T7VEN) bits in the timer interrupt skipping set register (TITCR1A (TITCR1B)) are set to 0 or the skipping count set bits (T3ACOR and T4VCOR (T6ACOR and T7VCOR)) in TITCR1A (TITCR1B) are set to 0), make sure that A/D conversion start requests are not linked with interrupt skipping 1 (set the ITA3AE, ITA4VE, ITB3AE, and ITB4VE (ITA6AE, ITA7VE, ITB6AE, and ITB7VE) bits in the timer A/D conversion start request control register (MTU4.TADCR (MTU7.TADCR)) to 0).
When this function is used, MTU4.TADCORA and MTU4.TADCORB (MTU7.TADCORA and MTU7.TADCORB) should be set with the value ranging 0002h to the value set in TCDRA minus 2 (value set in TCDRB minus 2).

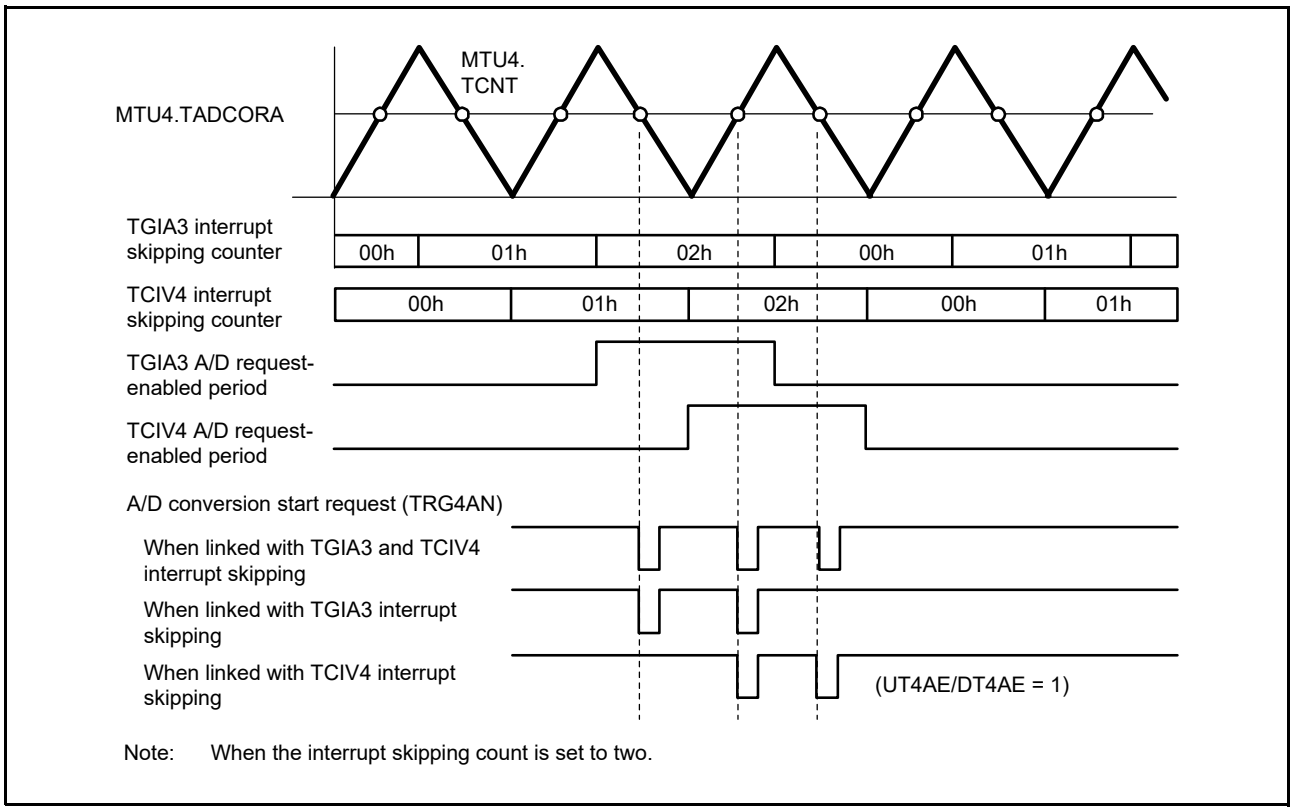


Figure 22.94 Example of A/D Conversion Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping Function 1 (UT4AE and DT4AE = 1)

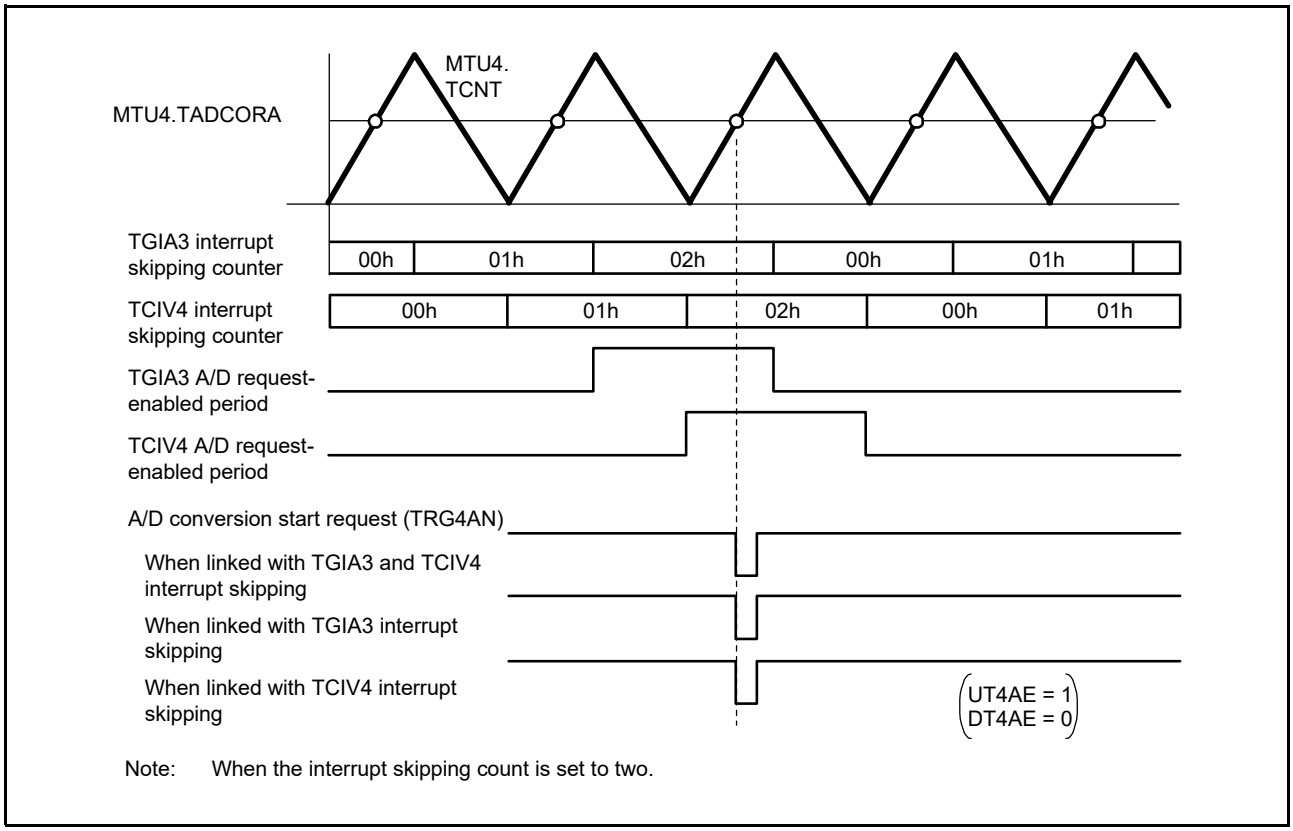


Figure 22.95 Example of A/D Conversion Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping Function 1 (UT4AE = 1, DT4AE = 0)

(6) A/D Conversion Start Request Delaying Function Linked with Interrupt Skipping Function 2

By setting the TITM bit to 1 in the TITMRA (TITMRB) register, the counter starts down-counting from the value (0 to 7) set in the TRG4COR[2:0] (TRG7COR[2:0]) bits in TITCR2A (TITCR2B) register every time an A/D conversion start trigger (TRG4AN or TRG4BN (TRG7AN or TRG7BN)) is generated. When the counter value reaches 0 and is reloaded, the TRG4AN and TRG4BN (TRG7AN and TRG7BN) interrupts become valid and an A/D conversion start request signal (TRG4ABN (TRG7ABN)) is output.
This function is valid only when the A/D conversion start request delaying function is enabled.

(a) Example of Procedure for Setting Interrupt Skipping Function 2

Figure 22.96 shows an example of procedure for setting interrupt skipping function 2.

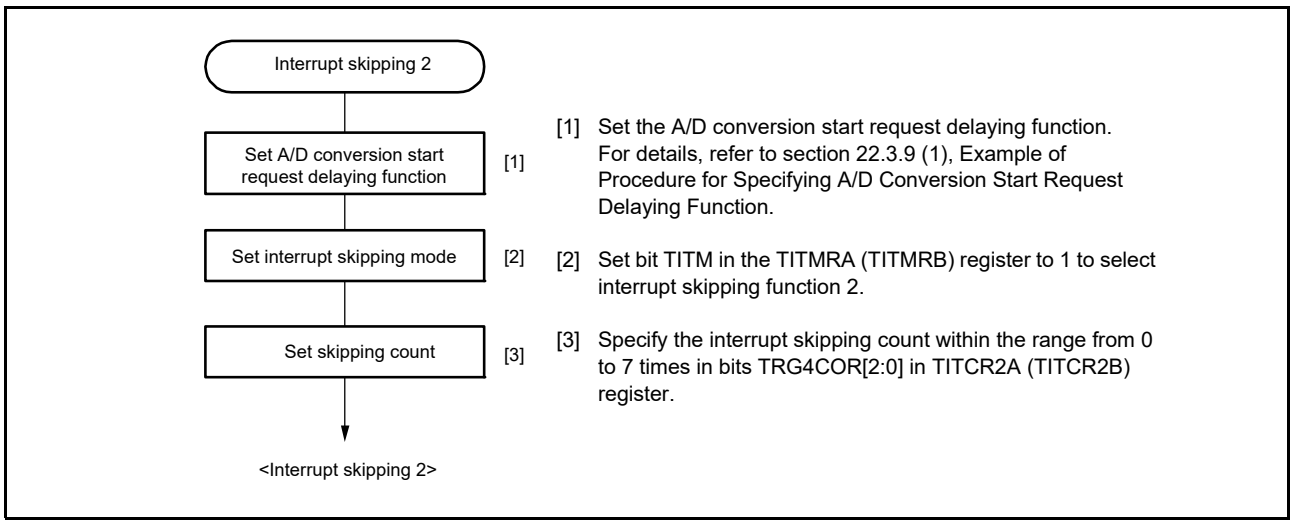


Figure 22.96 Example of Procedure for Setting Interrupt Skipping Function 2

(b) Example of Interrupt Skipping Function 2 Operation

Figure 22.97 shows an example of interrupt skipping function 2 operation.

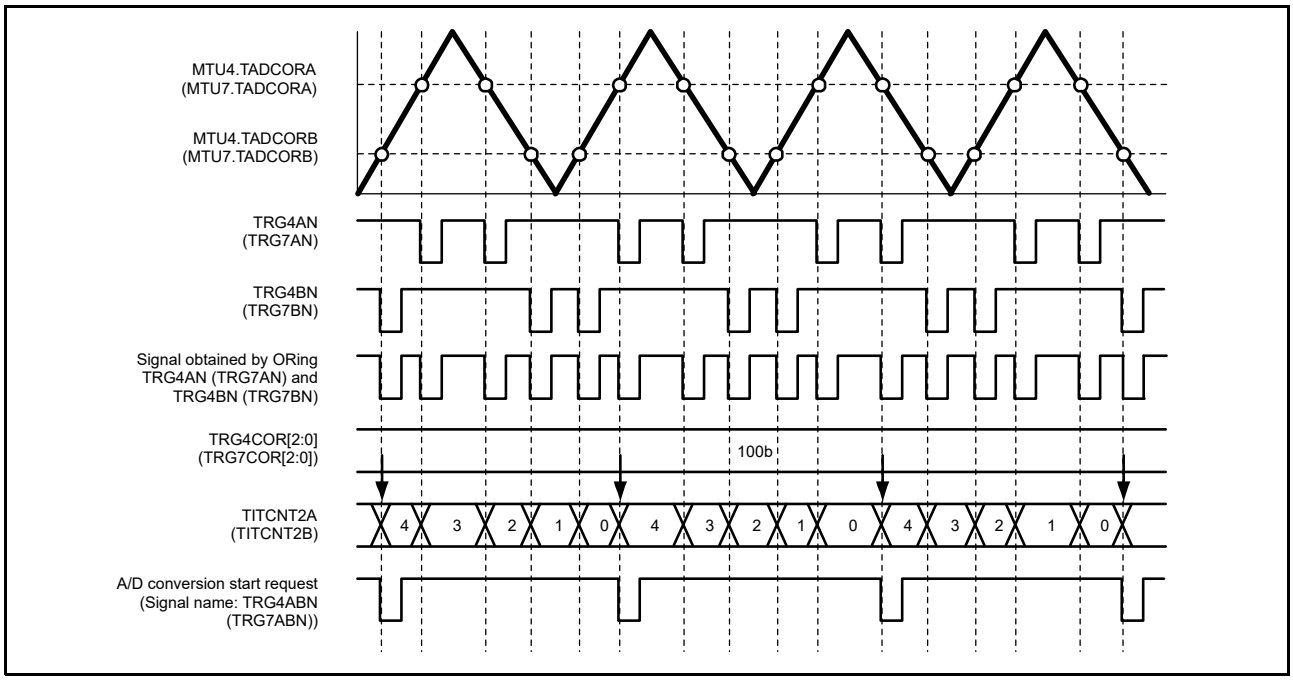


Figure 22.97 Example of Interrupt Skipping Function 2 Operation (Skipping Count is Set to Four)

22.3.10 Synchronous Operation of MTU0 to MTU4, MTU6, MTU7, and MTU9

(1) Synchronous Counter Start for MTU0 to MTU4, MTU6, MTU7, and MTU9

The counters in MTU0 to MTU4, MTU6, MTU7, and MTU9 can be started synchronously by making the TCSYSTR settings.

(a) Example of Procedure for Setting Synchronous Counter Start for MTU0 to MTU4, MTU6, MTU7, and MTU9

Figure 22.98 shows an example of procedure for setting synchronous counter start for MTU0 to MTU4, MTU6, MTU7, and MTU9.

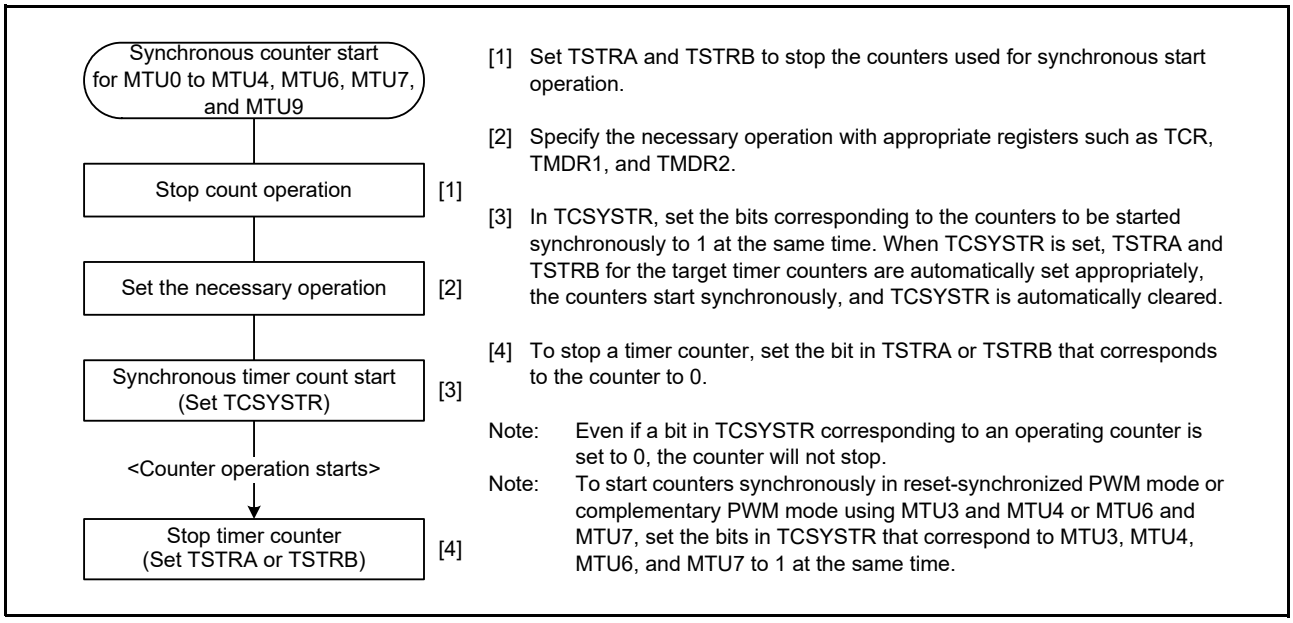


Figure 22.98 Example of Procedure for Setting Synchronous Counter Start for MTU0 to MTU4, MTU6, MTU7, and MTU9

(b) Examples of Synchronous Counter Start Operation

Figure 22.99 shows an examples of synchronous counter start operation for MTU0 to MTU4, MTU6, MTU7, and MTU9.

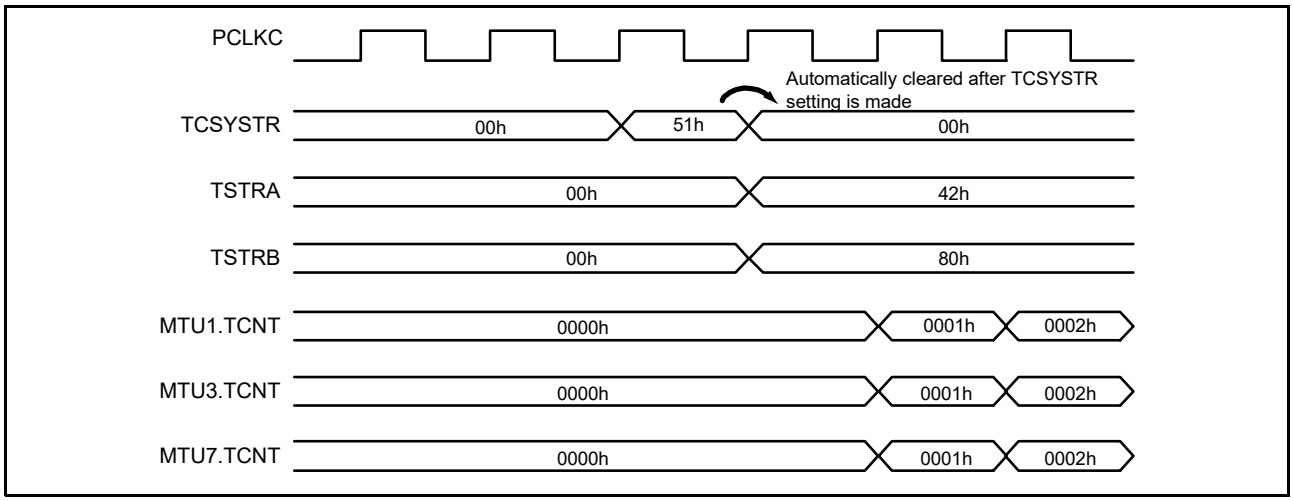


Figure 22.99 Examples of Synchronous Counter Start Operation for MTU0 to MTU4, MTU6, MTU7, and MTU9

(2) Synchronous Counter Clearing for MTU6 and MTU7

The counters in MTU6 and MTU7 can be cleared by the TGImn interrupt generation timing (m = A to D; n = 0 to 2) through the TSYCR setting.

(a) Example of Procedure for Specifying Synchronous Counter Clearing for MTU6 and MTU7

Figure 22.100 shows an example of procedure for specifying synchronous counter clearing for MTU6 and MTU7 by interrupt generation timing.

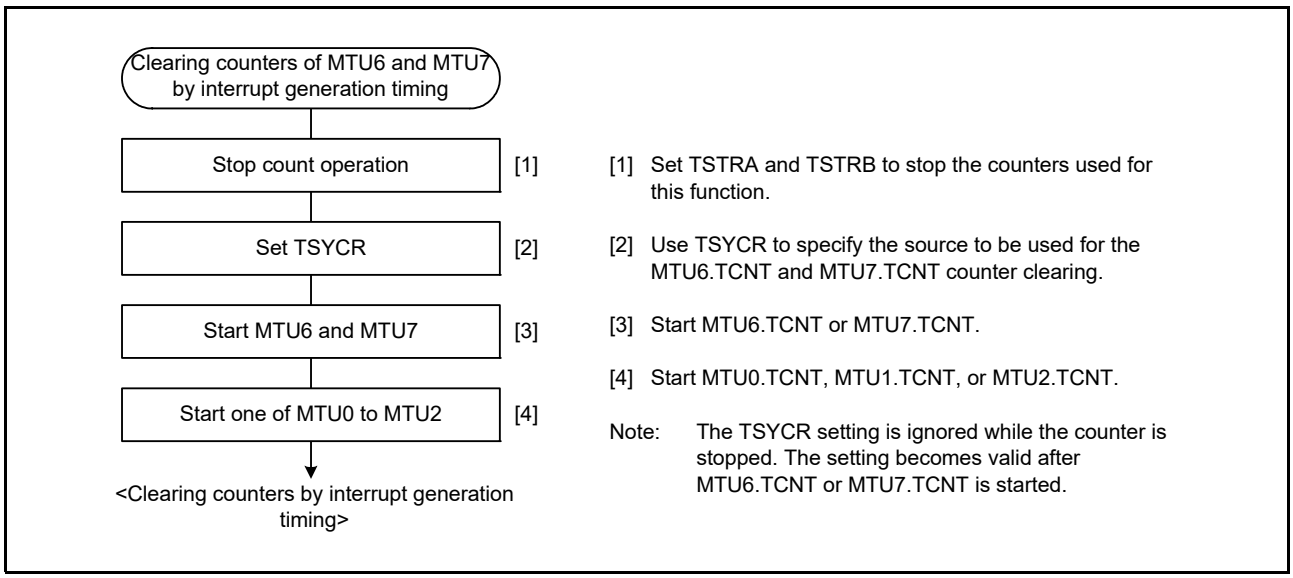


Figure 22.100 Example of Procedure for Specifying Synchronous Counter Clearing for MTU6 and MTU7

(b) Examples of Synchronous Counter Clearing for MTU6 and MTU7

Figure 22.101 and Figure 22.102 show examples of synchronous counter clearing for MTU6 and MTU7 by interrupt generation timing.

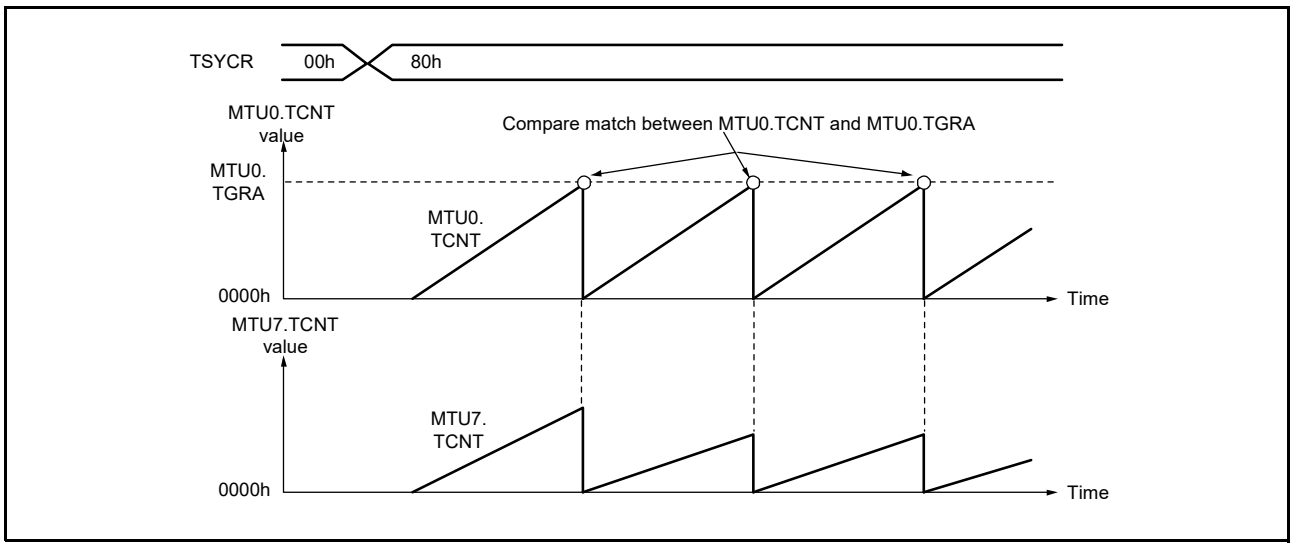


Figure 22.101 Example of Synchronous Counter Clearing for MTU6 and MTU7 (1)

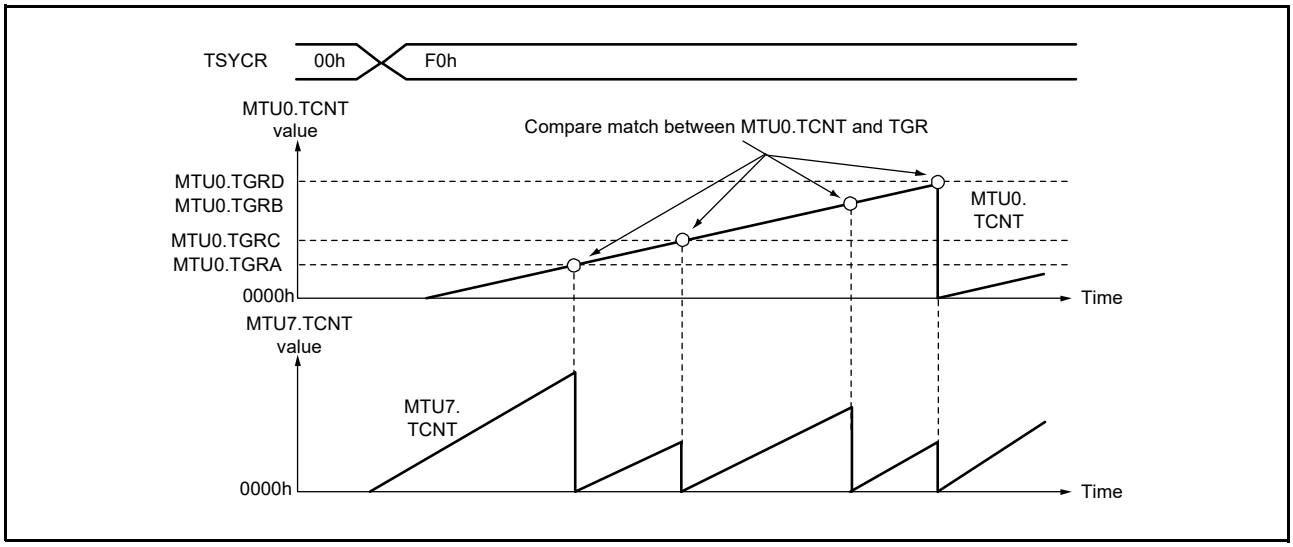


Figure 22.102 Example of Synchronous Counter Clearing for MTU6 and MTU7 (2)

22.3.11 External Pulse Width Measurement

The pulse widths of up to three external input lines can be measured in MTU5.
When the IOC[4:0] bits in MTU5.TIORU, MTU5.TIORV, MTU5.TIORW are set for pulse width measurement, the pulse width of the signal input to the MTIC5U, MTIC5V, and MTIC5W pins are measured. TCNTU, TCNTV, and TCNTW count up while the level specified by the IOC[4:0] bits is input.
Figure 22.103 shows an example of setting external pulse width measurement, and Figure 22.104 an example of external pulse width measurement.

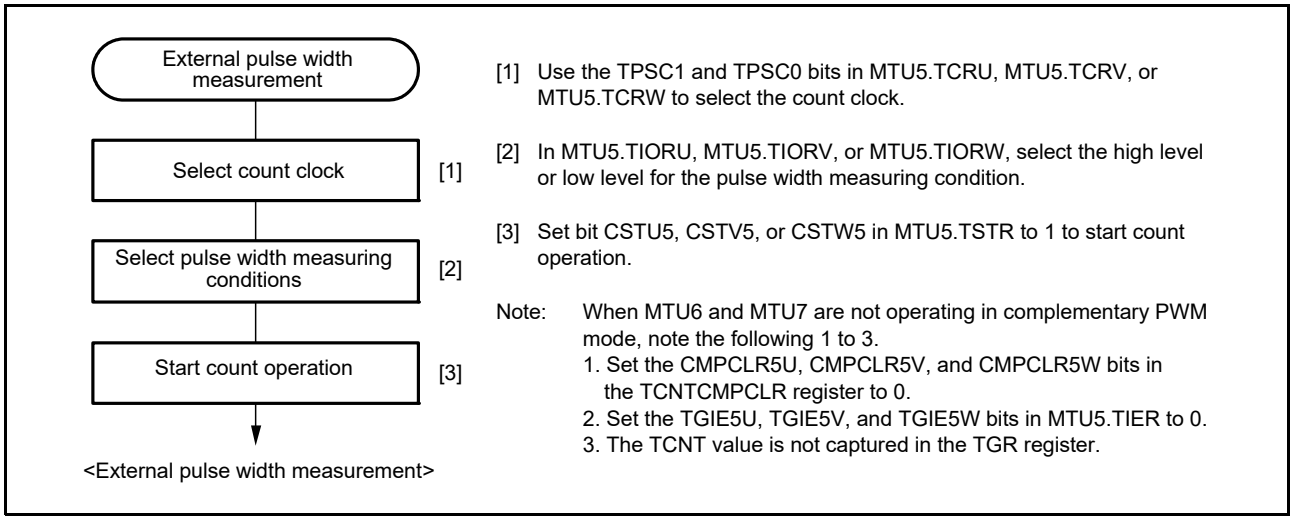


Figure 22.103 Example of External Pulse Width Measurement Setting Procedure

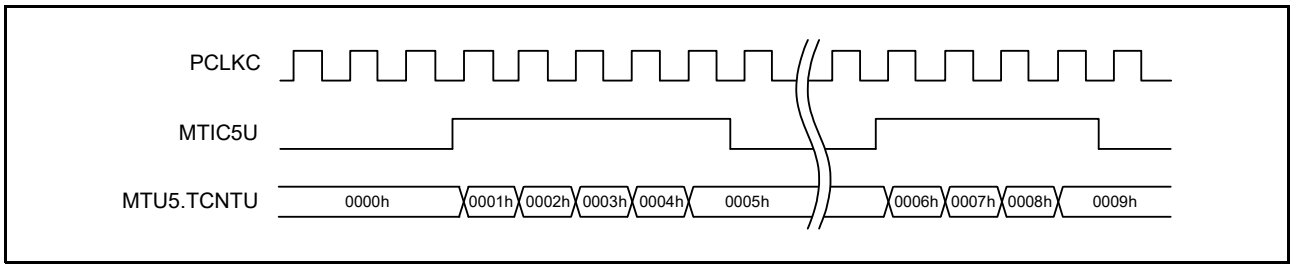


Figure 22.104 Example of External Pulse Width Measurement (Measuring High Pulse Width)

22.3.12 Dead Time Compensation

A dead time delay (a propagation delay of the inverter output from the complementary PWM output) can be compensated by combining MTU5 with MTU6 and MTU7. Figure 22.105 shows an example of the motor control circuit compensating a dead time delay by combining MTU5 with MTU6 and MTU7. A dead time for the PWM output waveform during complementary PWM operation using MTU6 and MTU7 can be compensated by adjusting a duty ratio set in a compare register for the PWM output after measuring a delay of the inverter output from the complementary PWM output by an external pulse measurement function for MTU5 (Figure 22.106). Figure 22.107 shows the procedure for setting dead time compensation using MTU5 to MTU7. For details on MTU5 operation at this time, refer to section 22.3.13, TCNTU, TCNTV, and TCNTW Capture at Crest and/or Trough in Complementary PWM Mode.

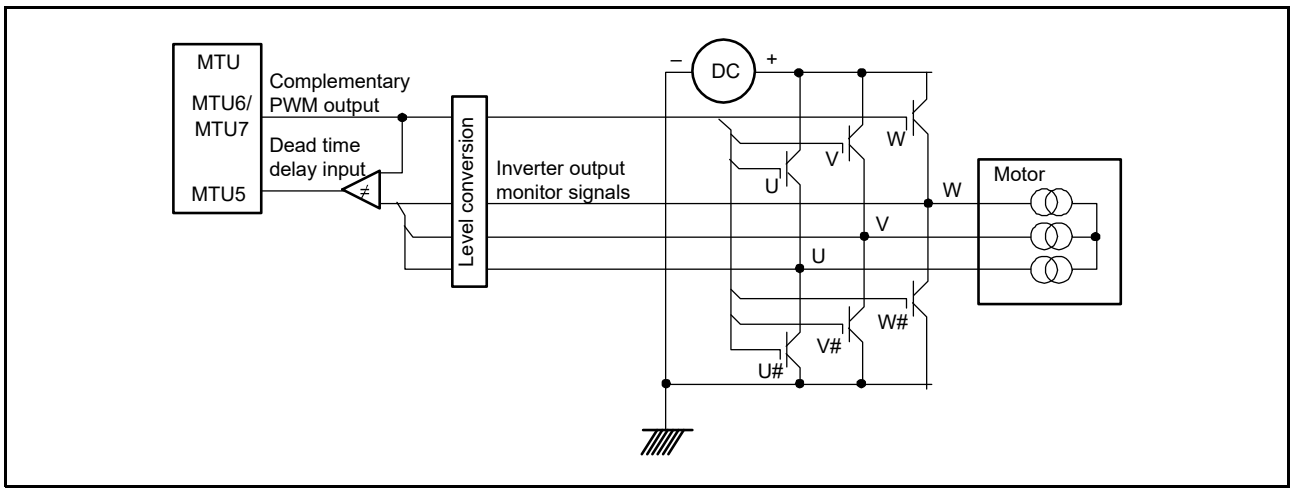


Figure 22.105 Motor Control Circuit Example

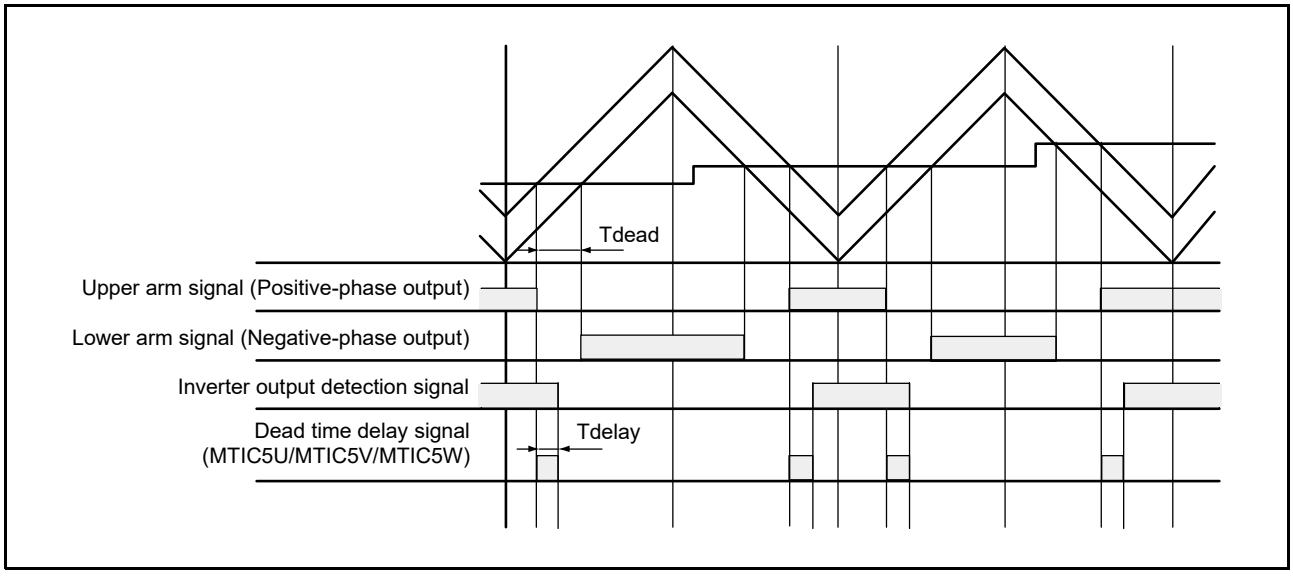
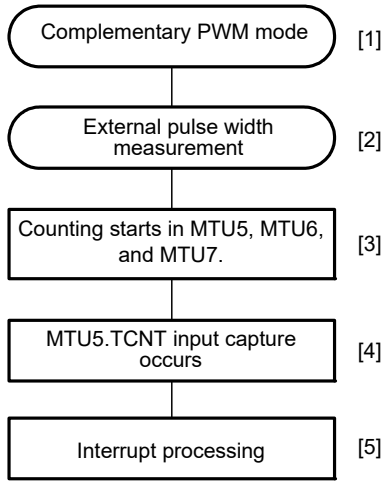


Figure 22.106 Delay in Dead Time in Complementary PWM Operation

(1) Example of Dead Time Compensation Setting Procedure

Figure 22.107 shows an example of dead time compensation setting procedure by using three counters in MTU5.



[1] Set MTU6 and MTU7 to complementary PWM mode. For details, refer to section 22.3.8, Complementary PWM Mode.

[2] Specify the external pulse width measurement function for the target TIOR in MTU5. For details, refer to section 22.3.11, External Pulse Width Measurement.

[3] Set bits CST6 and CST7 in TSTRB and bits CSTU5, CSTV5, and CSTW5 in MTU5.TSTR to 1 to start count operation.

[4] When the capture condition specified in TIOR is satisfied, the MTU5.TCNT value is captured in MTU5.TGR.
To prevent the generation of the MTU5.TGIm5 interrupt (m = U, V, W) request under the capture condition specified in TIOR, do not enable interrupt requests in MTU5.TIER.

[5] For U-phase dead time compensation, when an interrupt is generated at the crest (TGIA6) or trough (TCIV7) in complementary PWM mode, read the MTU5.TGRU value, calculate the difference in time between MTU6.TGRB and MTU5.TGRU, and write the corrected value to MTU6.TGRD in the interrupt processing.
For the V phase and W phase, read the MTU5.TGRV and MTU5.TGRW values and write the corrected values to MTU7.TGRC and TGRD, respectively, in the same way as for U-phase compensation.
The MTU5.TCNT value should be cleared through the TCNTCMPCLR setting or by software.

Note: The MPC and port mode registers (PMR) of the I/O ports should be set up in advance.
As the interrupt flag is set under the capture condition specified in TIOR, do not enable interrupt requests in MTU5.TIER.

Figure 22.107 Example of Dead Time Compensation Setting Procedure

22.3.13 TCNTU, TCNTV, and TCNTW Capture at Crest and/or Trough in Complementary PWM Mode

The MTU5 external pulse width measurement function allows to transfer the value in TCNTU, TCNTV, and TCNTW to TGRU, TGRV, and TGRW at the crest, or trough, or crest and trough when MTU6 and MTU7 operate in complementary PWM mode. The transfer timing is set in TIORU, TIORV, and TIORW. When the CMPCLR5U, CMPCLR5V, and CMPCLR5W bits in the TCNTCMPCLR register are set to 1, TCNTU, TCNTV, and TCNTW become 0000h at the transfer timing for TGRU, TGRV, and TGRW.

When MTU3 and MTU4 operate in complementary PWM mode, MTU5 cannot operate a capture operation of TCNTU, TCNTV, and TCNTW at the crest, or trough, or crest and trough in complementary PWM mode.

Figure 22.108 shows an operation example in which TCNTU is used as a free-running counter without being cleared, and the value is captured in TGRU at the crest and trough in complementary PWM mode.

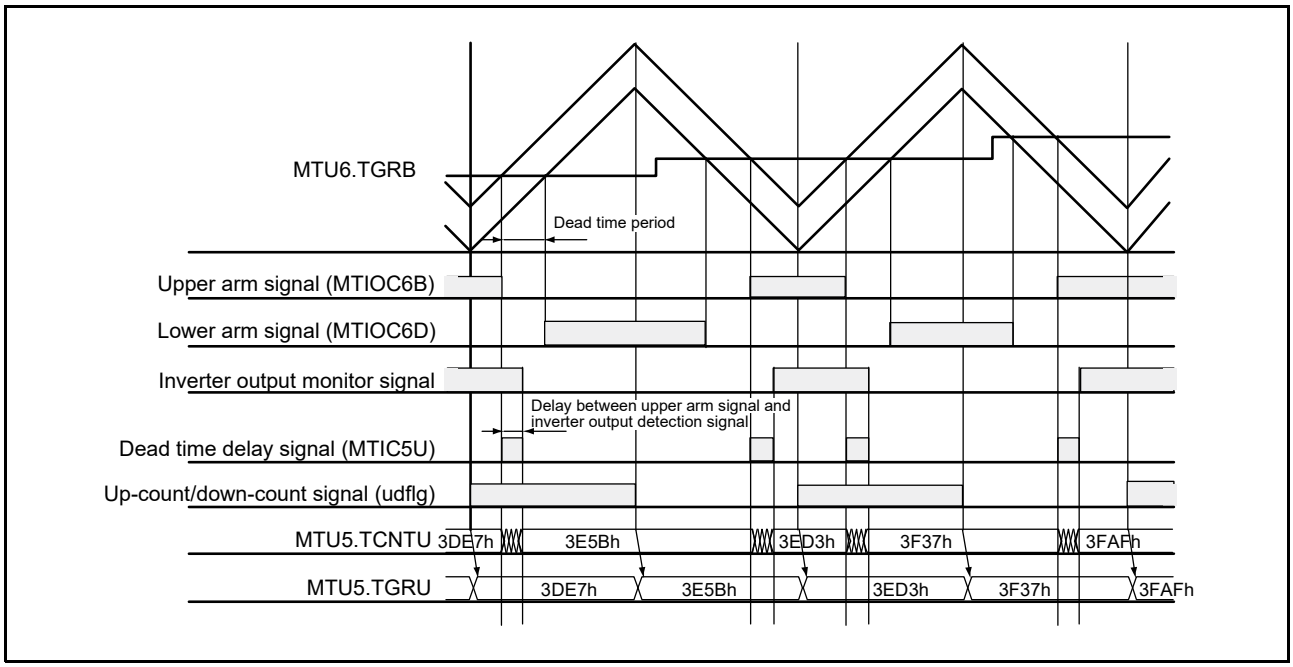


Figure 22.108 TCNTU Capture at Crest and Trough in Complementary PWM Operation

22.3.14 Noise Filter Function

The input capture input pins and external pulse input pins have a noise filter function.
Set the NFCRn register (n = 0 to 7, 9, C) to enable or disable the noise filter function and set the sampling clock. The noise filter for each pin can be enabled or disabled individually, and the sampling clock can be set for each channel.
Figure 22.109 shows the timing of noise filtering.

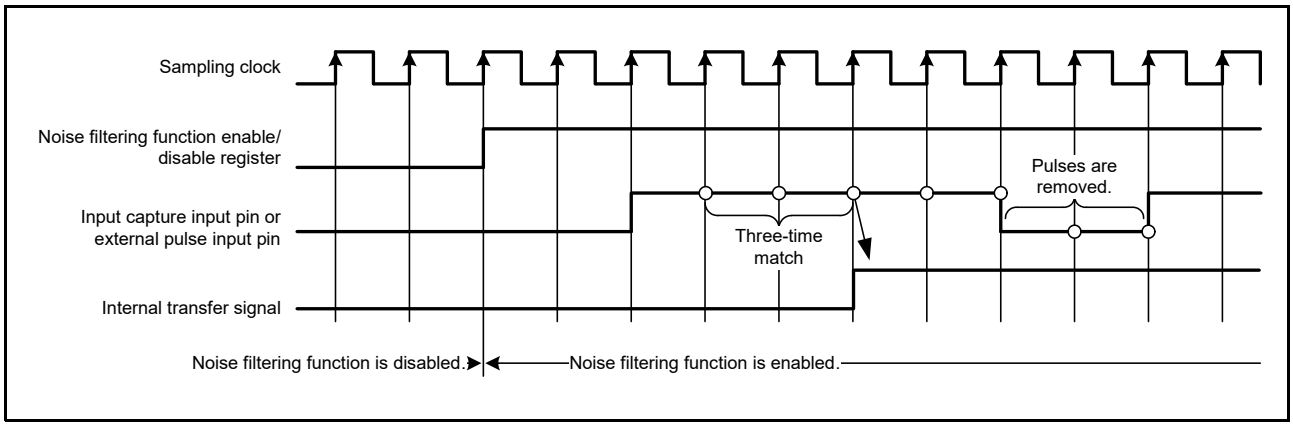
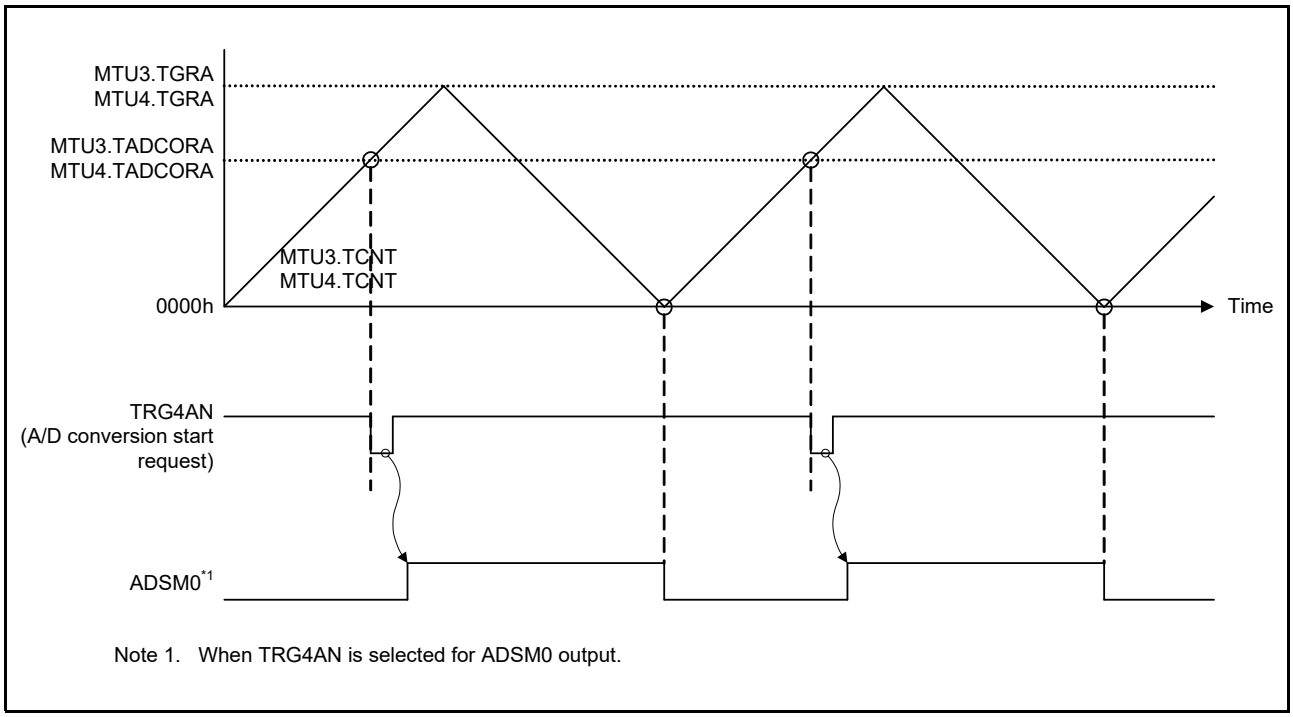


Figure 22.109 Timing of Noise Filtering

22.3.15 A/D Conversion Start Request Frame Synchronization Signal

This function can be used to monitor the generation timing of the A/D conversion start request signal using an external pin.
When the A/D conversion start request signal to be monitored is selected by the TADSTRGRn register (n = 0, 1) and the ADSMn pin output is enabled by TADSTRGRn.TADSMENn, a pulse signal is output from the ADSMn pin that is at the high level when the A/D conversion start request signal is generated, and at the low level in the timer cycle used to generate the A/D conversion start request signal.
Figure 22.110 shows an example of outputting the A/D conversion start request frame synchronization signal.



Note 1. When TRG4AN is selected for ADSM0 output.

Figure 22.110 Example of Outputting A/D Conversion Start Request Frame Synchronization Signal

22.4 Interrupt Sources

22.4.1 Interrupt Sources and Priorities

There are three kinds of interrupt source; TGR input capture/compare match, TCNT overflow, and TCNT underflow. Each interrupt source has its own enable/disable bit, allowing the generation of interrupt request signals to be enabled or disabled individually.

When an interrupt source is generated, if the corresponding enable/disable bit in TIER is set to 1, an interrupt is requested.

Relative channel priorities can be changed by the interrupt controller; however the priority within a channel is fixed. For details, refer to section 14, Interrupt Controller (ICUG). Table 22.78 lists the MTU interrupt sources.

Table 22.78 MTU Interrupt Sources

Channel	Name	Interrupt Source	DMAC/DTC Activation
MTU0	TGIA0	MTU0.TGRA input capture/compare match	Possible
	TGIB0	MTU0.TGRB input capture/compare match	Possible
	TGIC0	MTU0.TGRC input capture/compare match	Possible
	TGID0	MTU0.TGRD input capture/compare match	Possible
	TCIV0	MTU0.TCNT overflow	Not possible
	TGIE0	MTU0.TGRE compare match	Not possible
	TGIF0	MTU0.TGRF compare match	Not possible
MTU1	TGIA1	MTU1.TGRA input capture/compare match	Possible
	TGIB1	MTU1.TGRB input capture/compare match	Possible
	TCIV1	MTU1.TCNT overflow	Not possible
	TCIU1	MTU1.TCNT underflow	Not possible
MTU2	TGIA2	MTU2.TGRA input capture/compare match	Possible
	TGIB2	MTU2.TGRB input capture/compare match	Possible
	TCIV2	MTU2.TCNT overflow	Not possible
	TCIU2	MTU2.TCNT underflow	Not possible
MTU3	TGIA3	MTU3.TGRA input capture/compare match	Possible
	TGIB3	MTU3.TGRB input capture/compare match	Possible
	TGIC3	MTU3.TGRC input capture/compare match	Possible
	TGID3	MTU3.TGRD input capture/compare match	Possible
	TCIV3	MTU3.TCNT overflow	Not possible
MTU4	TGIA4	MTU4.TGRA input capture/compare match	Possible
	TGIB4	MTU4.TGRB input capture/compare match	Possible
	TGIC4	MTU4.TGRC input capture/compare match	Possible
	TGID4	MTU4.TGRD input capture/compare match	Possible
	TCIV4	MTU4.TCNT overflow/underflow*1	Possible
MTU5	TGIU5	MTU5.TGRU input capture/compare match	Possible
	TGIV5	MTU5.TGRV input capture/compare match	Possible
	TGIW5	MTU5.TGRW input capture/compare match	Possible
MTU6	TGIA6	MTU6.TGRA input capture/compare match	Possible
	TGIB6	MTU6.TGRB input capture/compare match	Possible
	TGIC6	MTU6.TGRC input capture/compare match	Possible
	TGID6	MTU6.TGRD input capture/compare match	Possible
	TCIV6	MTU6.TCNT overflow	Not possible
MTU7	TGIA7	MTU7.TGRA input capture/compare match	Possible
	TGIB7	MTU7.TGRB input capture/compare match	Possible
	TGIC7	MTU7.TGRC input capture/compare match	Possible
	TGID7	MTU7.TGRD input capture/compare match	Possible
	TCIV7	MTU7.TCNT overflow/underflow*1	Possible
MTU9	TGIA9	MTU9.TGRA input capture/compare match	Possible
	TGIB9	MTU9.TGRB input capture/compare match	Possible
	TGIC9	MTU9.TGRC input capture/compare match	Possible
	TGID9	MTU9.TGRD input capture/compare match	Possible
	TCIV9	MTU9.TCNT overflow	Not possible
	TGIE9	MTU9.TGRE compare match	Not possible
	TGIF9	MTU9.TGRF compare match	Not possible

Note: This table shows the initial state immediately after a reset. The relative channel priorities can be changed by the interrupt controller.

Note 1. Underflow is available only in complementary PWM mode.

(1) Input Capture/Compare Match Interrupt

If the TIER.TGIE bit is set to 1 when a TGR input capture/compare match occurs on a channel, an interrupt is requested. The MTU has 35 input capture/compare match interrupts (six for MTU0 and MTU9, four each for MTU3, MTU4, MTU6, and MTU7, two each for MTU1 and MTU2, and three for MTU5).

(2) Overflow Interrupt

If the TIER.TCIEV bit is set to 1 when a TCNT overflow occurs on a channel, an interrupt is requested. The MTU has eight overflow interrupts (one for each channel except MTU5).
Note that an overflow interrupt is generated also when an underflow of the MTU4.TCNT and MTU7.TCNT occurs while operating in complementary PWM mode.

(3) Underflow Interrupt

If the TIER.TCIEU bit is set to 1 when a TCNT underflow occurs on a channel, an interrupt is requested. The MTU has two underflow interrupts (one each for MTU1, and MTU2).

22.4.2 DTC/DMAC Trigger Sources

(1) DTC Trigger Sources

The DTC can be triggered by the TGR input capture/compare match interrupt in each channel or the overflow interrupt in MTU4 and MTU7. For details, refer to section 18, Data Transfer Controller (DTCb).
The MTU provides a total of 33 input capture/compare match interrupts and overflow interrupts that can be used as DTC trigger sources: four each for MTU0, MTU3, MTU6, and MTU9, two each for MTU1 and MTU2, five each for MTU4 and MTU7, and three for MTU5.

(2) DMAC Trigger Sources

The DMAC can be triggered by the TGR input capture/compare match interrupt in each channel and the overflow interrupt in MTU4 and MTU7. For details, refer to section 17, DMA Controller (DMACAa).
The MTU provides a total of 33 input capture/compare match interrupts and overflow interrupts that can be used as DMAC trigger sources: four each for MTU0, MTU3, MTU6, and MTU9, two each for MTU1 and MTU2, five each for MTU4 and MTU7, and three for MTU5.
If a DMA transfer is initiated by the MTU, the trigger signal is cleared when the DMAC requests ownership of the internal bus. Accordingly, the state of the internal bus may lead to a wait for the start of DMA transfer.

22.4.3 A/D Converter Trigger Sources

The A/D converter can be triggered by one of the following three methods in the MTU. Table 22.79 shows the relationship between interrupt sources and A/D conversion start request signals.

(1) A/D Conversion Start by TGRA Input Capture/Compare Match or at Trough of MTU4.TCNT (MTU7.TCNT) in Complementary PWM Mode

The A/D converter can be triggered by the occurrence of a TGRA input capture/compare match in each channel. In addition, if complementary PWM operation is performed while the TTGE2 bit in MTU4.TIER (MTU7.TIER) is set to 1, the A/D converter can be triggered at the trough of MTU4.TCNT (MTU7.TCNT) count (MTU4.TCNT (MTU7.TCNT) = 0000h).
A/D conversion start request TRGAnN is issued to the A/D converter under either of the following conditions (n = 0 to 4, 6, 7, 9).

- When a TGRA input capture/compare match occurs on a channel while the TIER.TTGE bit is set to 1
- When the MTU4.TCNT (MTU7.TCNT) count reaches the trough (MTU4.TCNT (MTU7.TCNT) = 0000h) during complementary PWM operation while the TTGE2 bit in MTU4.TIER (MTU7.TIER) is set to 1

When either condition is satisfied, if A/D conversion start request signal TRGAnN from the MTU is selected as the trigger in the A/D converter, A/D conversion will start.

(2) A/D Conversion Start by Compare Match between MTU0.TCNT and MTU0.TGRE, or MTU9.TCNT and MTU9.TGRE

A/D conversion start request TRG0N and TRG9N is issued to the A/D converter when a compare match occurs between MTU0.TCNT and MTU0.TGRE, MTU9.TCNT and MTU9.TGRE.
When a compare match occurs between MTU0.TCNT and MTU0.TGRE, MTU9.TCNT and MTU9.TGRE while the TTGE2 bit in MTU0.TIER2, MTU9.TIER2 are set to 1, A/D conversion start request TRG0N, TRG0AEN, TRG9N, TRG9AEN, or TRG09N is issued to the A/D converter. If A/D conversion start request signal TRG0N, TRG0AEN, TRG9N, TRG9AEN, or TRG09N from the MTU is selected as the trigger in the A/D converter, A/D conversion will start.

(3) A/D Conversion Start by A/D Conversion Start Request Delaying Function

The A/D converter can be triggered by generating A/D conversion start request signal TRG4AN or TRG4BN (TRG7AN or TRG7BN) when the MTU4.TCNT (MTU7.TCNT) count matches the MTU4.TADCORA or MTU4.TADCORB (MTU7.TADCORA or MTU7.TADCORB) value if the UT4AE, DT4AE, UT4BE, or DT4BE (UT7AE, DT7AE, UT7BE, or DT7BE) bit in the A/D conversion start request control register (MTU4.TADCR (MTU7.TADCR)) is set to 1. For details, refer to section 22.3.9, A/D Conversion Start Request Delaying Function.
A/D conversion will start when TRG4AN (TRG7AN) is generated if A/D conversion start request signal TRG4AN (TRG7AN) from the MTU is selected as the trigger in the A/D converter, when TRG4BN (TRG7BN) is generated if TRG4BN (TRG7BN) from the MTU is selected as the trigger in the A/D converter, or when TRG4ABN (TRG7ABN) is generated if TRG4ABN (TRG7ABN) from the MTU is selected as the trigger in the A/D converter.

Table 22.79 Interrupt Sources and A/D Conversion Start Request Signals

Target Registers	Interrupt Source	A/D Conversion Start Request Signal
MTU0.TGRA and MTU0.TCNT	Input capture/compare match	TRGA0N
MTU9.TGRA and MTU9.TCNT		TRGA9N
MTU9.TGRA and MTU9.TCNT, MTU9.TGRE and MTU9.TCNT*1		TRG9AEN
MTU0.TGRA and MTU0.TCNT, MTU0.TGRE and MTU0.TCNT*1		TRG0AEN
MTU0.TGRA and MTU0.TCNT, MTU9.TGRA and MTU9.TCNT		TRGA09N
MTU1.TGRA and MTU1.TCNT		TRGA1N
MTU2.TGRA and MTU2.TCNT		TRGA2N
MTU3.TGRA and MTU3.TCNT		TRGA3N
MTU4.TGRA and MTU4.TCNT*2		TRGA4N
MTU4.TCNT	MTU4.TCNT trough in complementary PWM mode	
MTU6.TGRA and MTU6.TCNT	Input capture/compare match	TRGA6N
MTU7.TGRA and MTU7.TCNT*2		TRGA7N
MTU7.TCNT	MTU7.TCNT trough in complementary PWM mode	
MTU0.TGRE and MTU0.TCNT	Compare match	TRG0N
MTU9.TGRE and MTU9.TCNT		TRG9N
MTU0.TGRE and MTU0.TCNT*1, MTU9.TGRE and MTU9.TCNT*1		TRG09N
MTU4.TADCORA and MTU4.TCNT		TRG4AN
MTU4.TADCORB and MTU4.TCNT		TRG4BN
MTU7.TADCORA and MTU7.TCNT		TRG7AN
MTU7.TADCORB and MTU7.TCNT		TRG7BN
MTU4.TADCORA and MTU4.TCNT, MTU4.TADCORB and MTU4.TCNT	Compare match (interrupt skipping function 2)	TRG4ABN
MTU7.TADCORA and MTU7.TCNT, MTU7.TADCORB and MTU7.TCNT		TRG7ABN

Note 1. Since the compare match source of TGRE is used as the A/D trigger start source, set the MTU0.TIER2.TTGE2 and MTU9.TIER2.TTGE2 to 1.

Note 2. Since PWM waveforms are generated in complementary PWM mode, MTU4.TGRA (MTU7.TGRA) compare match not only with MTU4.TCNT (MTU7.TCNT) but also with MTU3.TCNT (MTU6.TCNT) and TCNTSA (TCNTSB) is detected. Accordingly, when compare match with MTU3.TCNT (MTU6.TCNT) and TCNTSA (TCNTSB) occurs, TRGA4N (TRGA7N) is also generated. When MTU3 and MTU 4 (MTU 6 and MTU7) are made to operate in complementary PWM mode for generating an A/D conversion start request, use the A/D conversion start request by compare match between MTU4.TCNT (MTU7.TCNT) and MTU4.TADCORA/TADCORB (MTU7.TADCORA/TADCORB).

22.5 Operation Timing

22.5.1 Input/Output Timing

(1) TCNT Count Timing

Figure 22.111 and Figure 22.112 show the TCNT count timing in internal clock operation, Figure 22.113 shows the TCNT count timing in external clock operation (normal mode), and Figure 22.114 shows the TCNT count timing in external clock operation (phase counting mode).

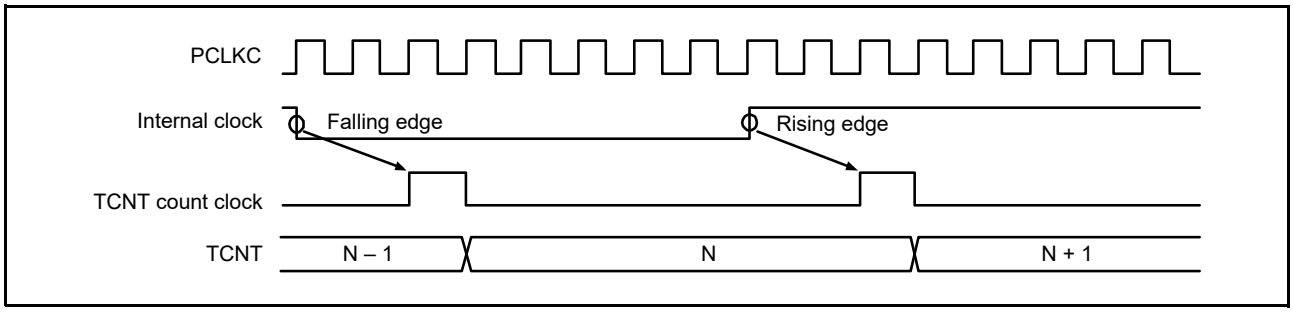


Figure 22.111 Count Timing in Internal Clock Operation (MTU0 to MTU4, MTU6, MTU7, and MTU9)

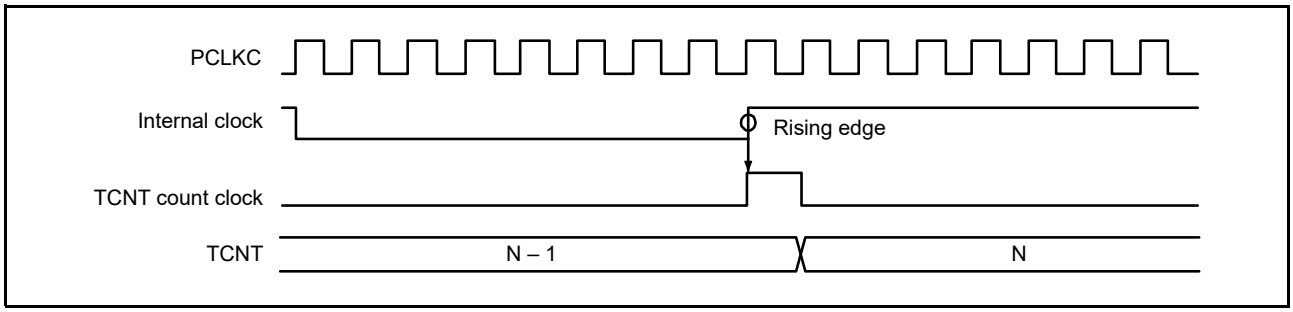


Figure 22.112 Count Timing in Internal Clock Operation (MTU5)

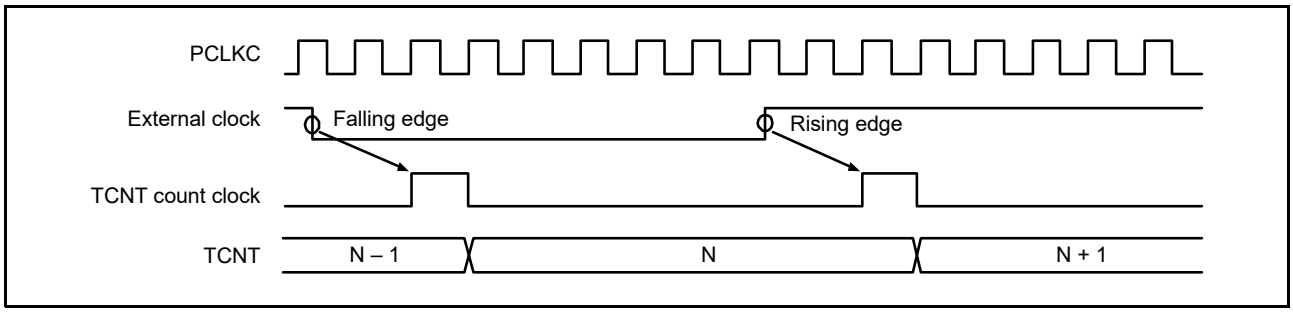


Figure 22.113 Count Timing in External Clock Operation (MTU0 to MTU4, MTU6, MTU7, and MTU9)

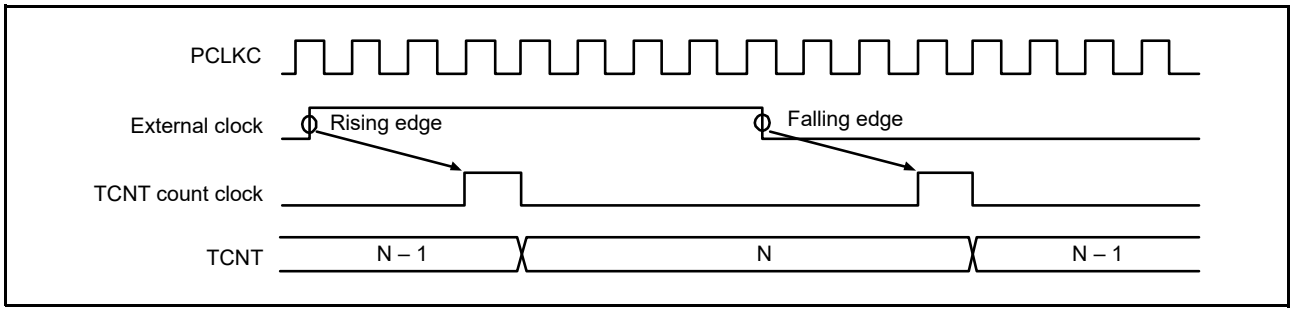


Figure 22.114 Count Timing in External Clock Operation (Phase Counting Mode)

(2) Output Compare Output Timing

A compare match signal is generated in the final state in which TCNT and TGR match (the point at which the count value matched by TCNT is updated). When a compare match signal is generated, the value set in TIOR is output from MTIOCnm pin (n = 0 to 4, 6, 7, 9; m = A to D). After a match between TCNT and TGR, the compare match signal is not generated until the TCNT count clock is generated.

Figure 22.115 shows the output compare output timing (normal mode or PWM mode) and Figure 22.116 shows the output compare output timing (complementary PWM mode or reset-synchronized PWM mode).

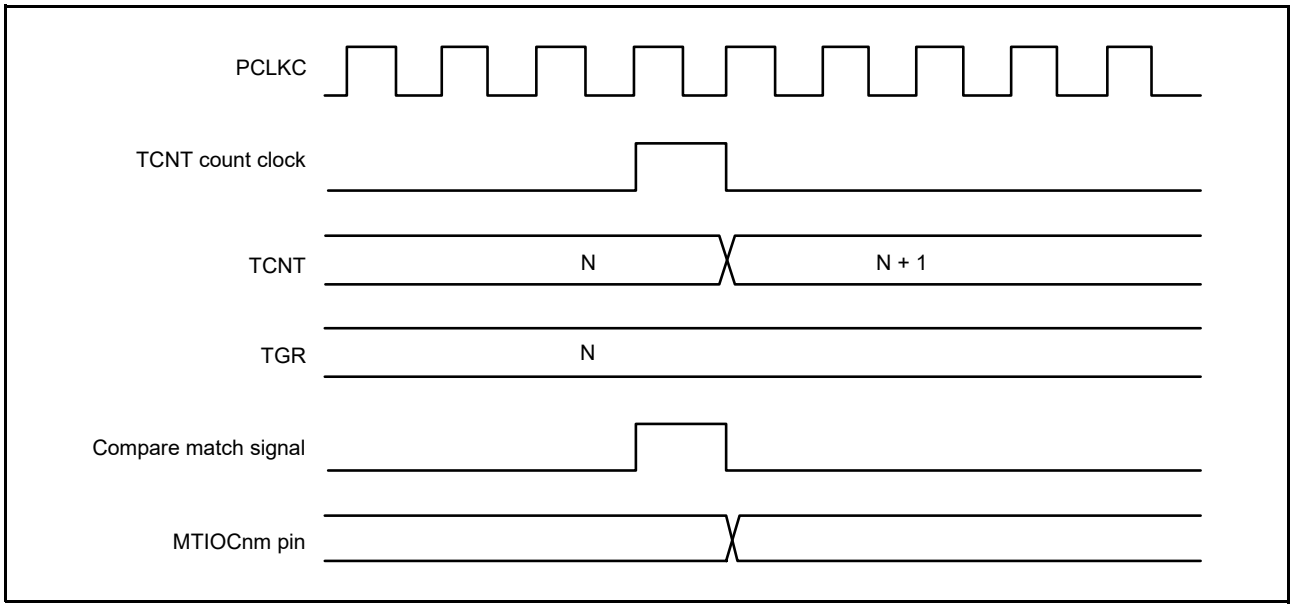


Figure 22.115 Output Compare Output Timing (Normal Mode or PWM Mode) (n = 0 to 4, 6, 7, 9; m = A to D)

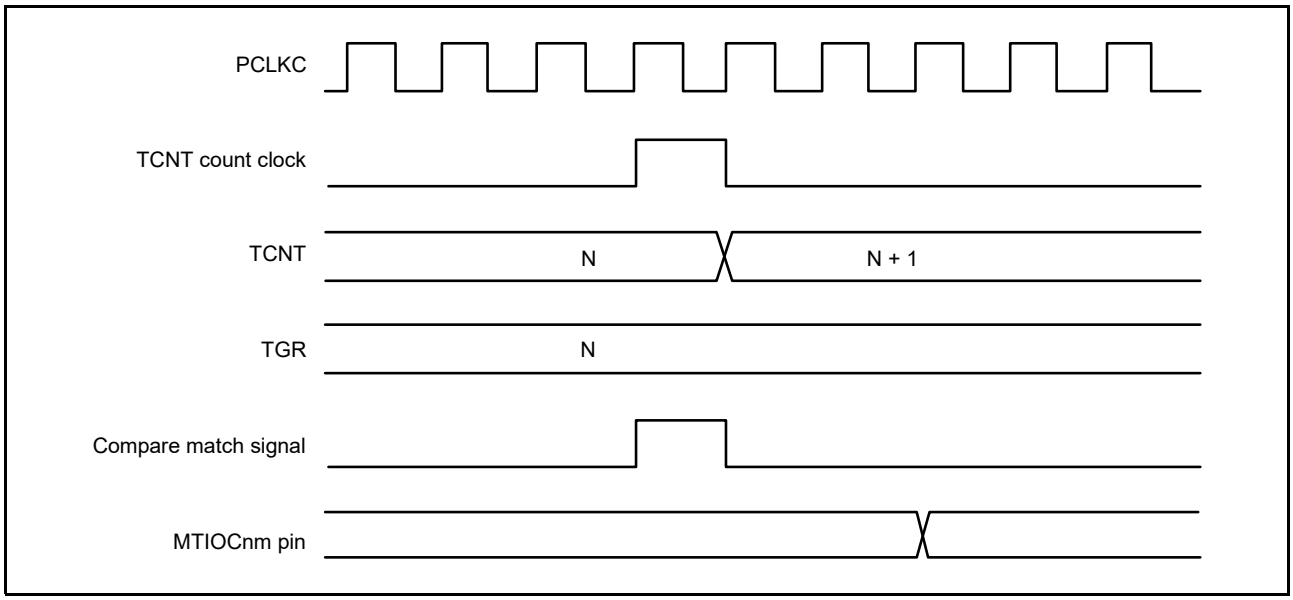


Figure 22.116 Output Compare Output Timing (Complementary PWM Mode or Reset-Synchronized PWM Mode) (n = 0 to 4, 6, 7, 9; m = A to D)

(3) Input Capture Signal Timing

Figure 22.117 shows the input capture signal timing.

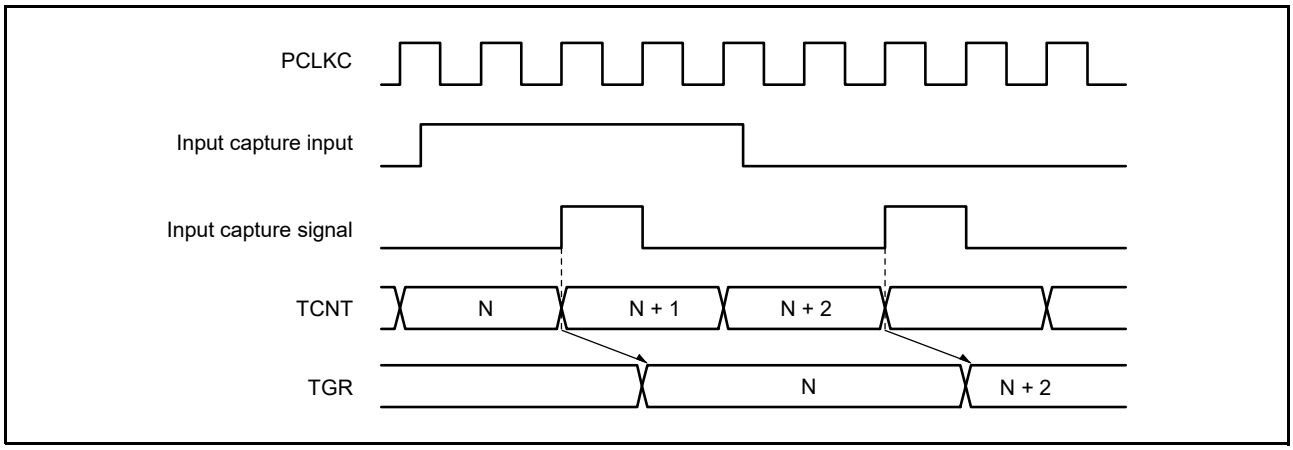


Figure 22.117 Input Capture Input Signal Timing

(4) Timing for Counter Clearing by Compare Match/Input Capture

Figure 22.118 and Figure 22.119 show the timing when counter clearing on compare match is specified, and Figure 22.120 shows the timing when counter clearing on input capture is specified.

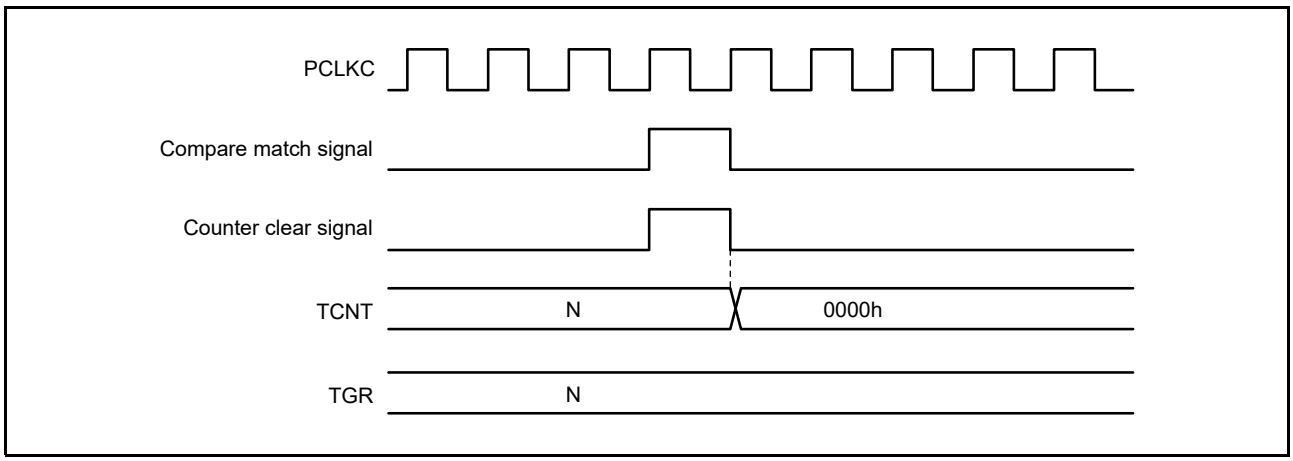


Figure 22.118 Counter Clear Timing (Compare Match) (MTU0 to MTU4, MTU6, MTU7, and MTU9)

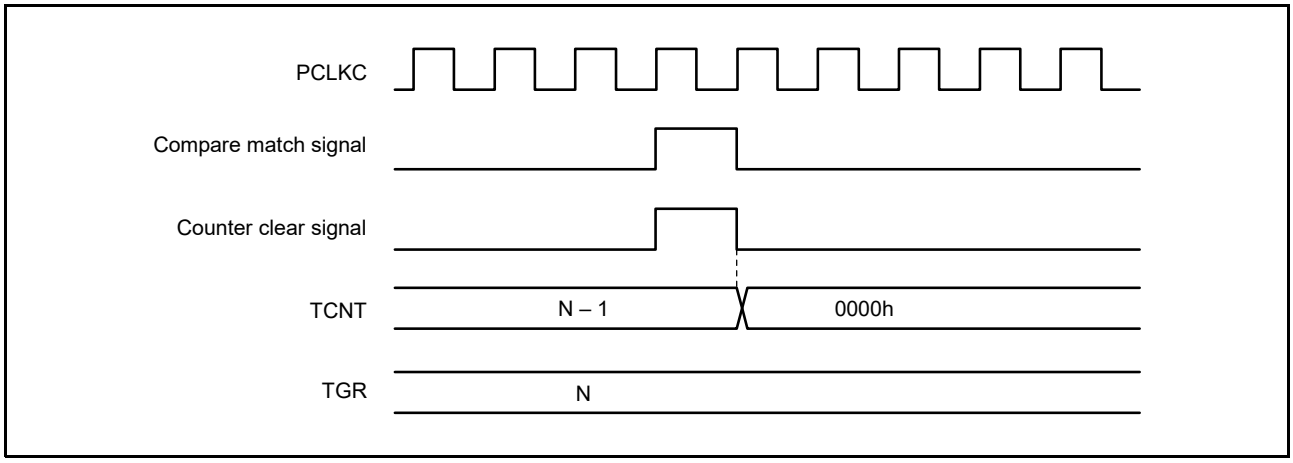


Figure 22.119 Counter Clear Timing (Compare Match) (MTU5)

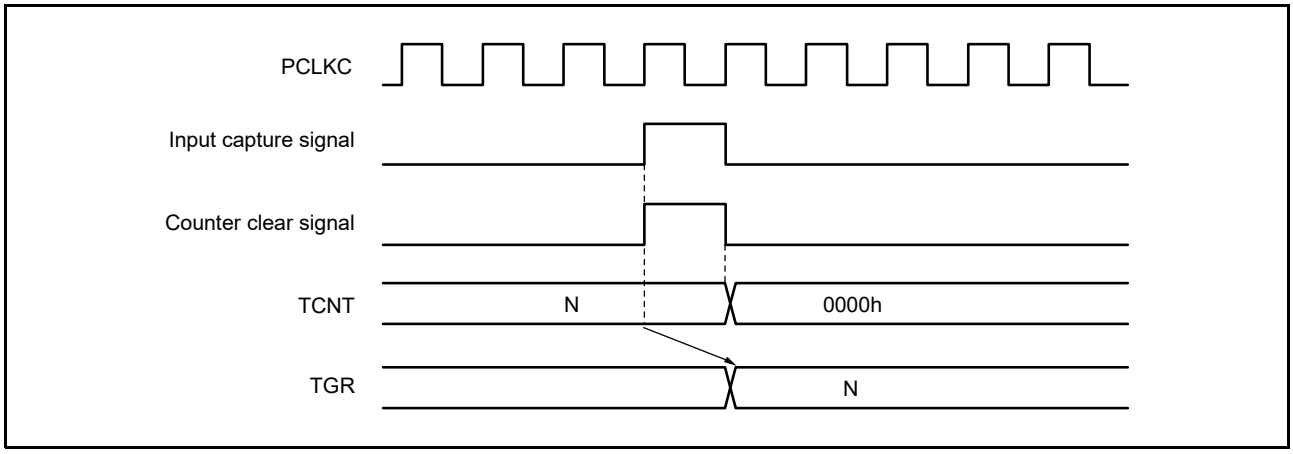


Figure 22.120 Counter Clear Timing (Input Capture) (MTU0 to MTU7, and MTU9)

(5) Buffer Operation Timing

Figure 22.121 to Figure 22.123 show the timing in buffer operation.

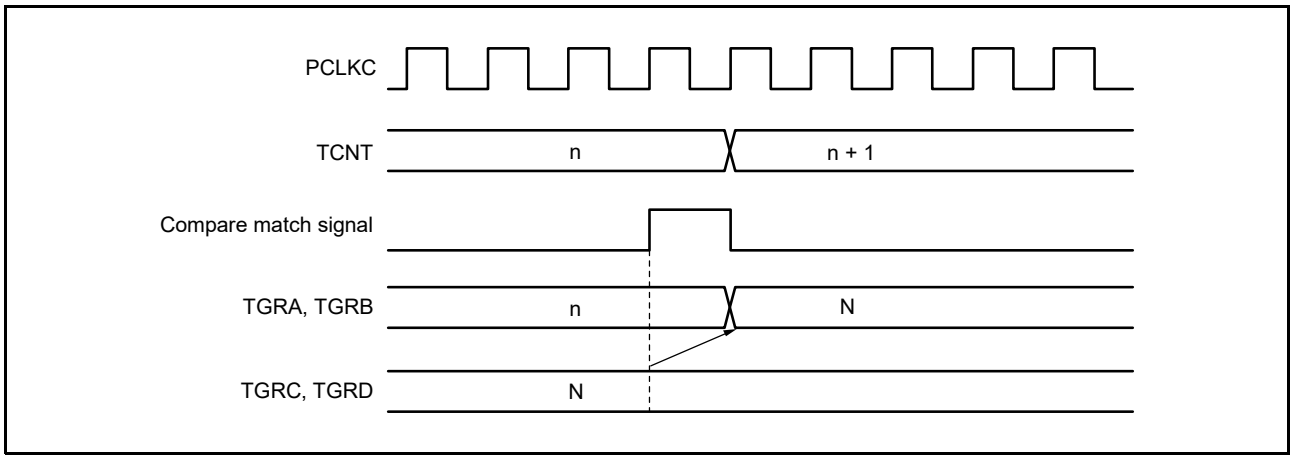


Figure 22.121 Buffer Operation Timing (Compare Match)

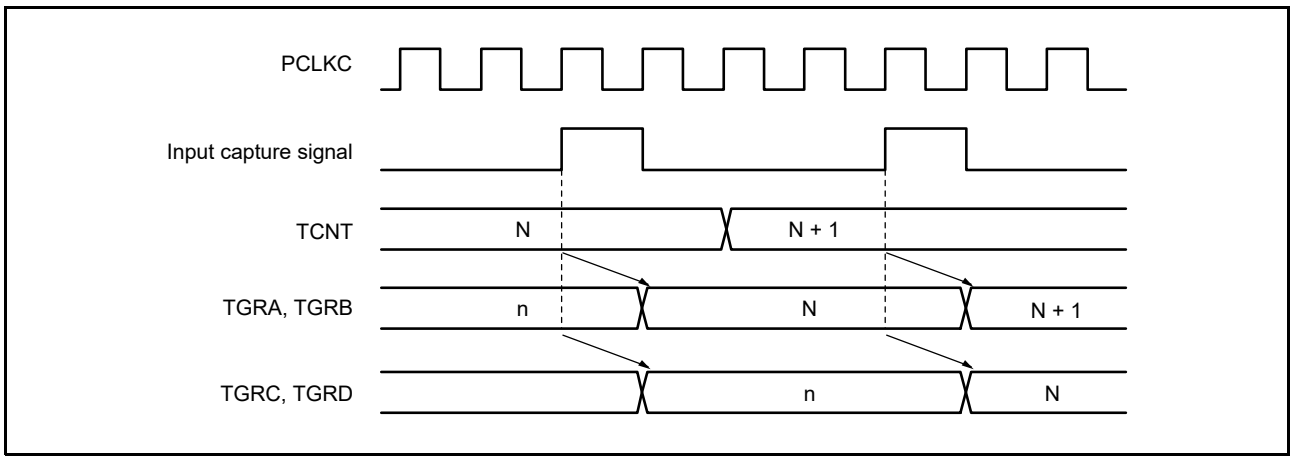


Figure 22.122 Buffer Operation Timing (Input Capture)

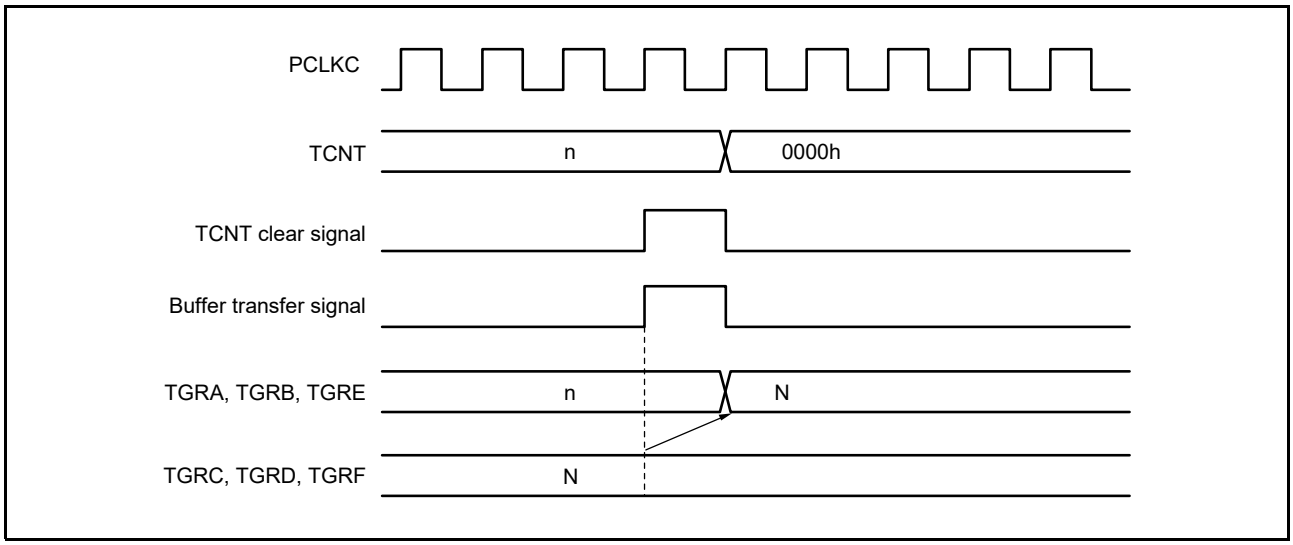


Figure 22.123 Buffer Operation Timing (When TCNT Cleared)

(6) Buffer Transfer Timing (Complementary PWM Mode)

Figure 22.124 to Figure 22.126 show the buffer transfer timing in complementary PWM mode.

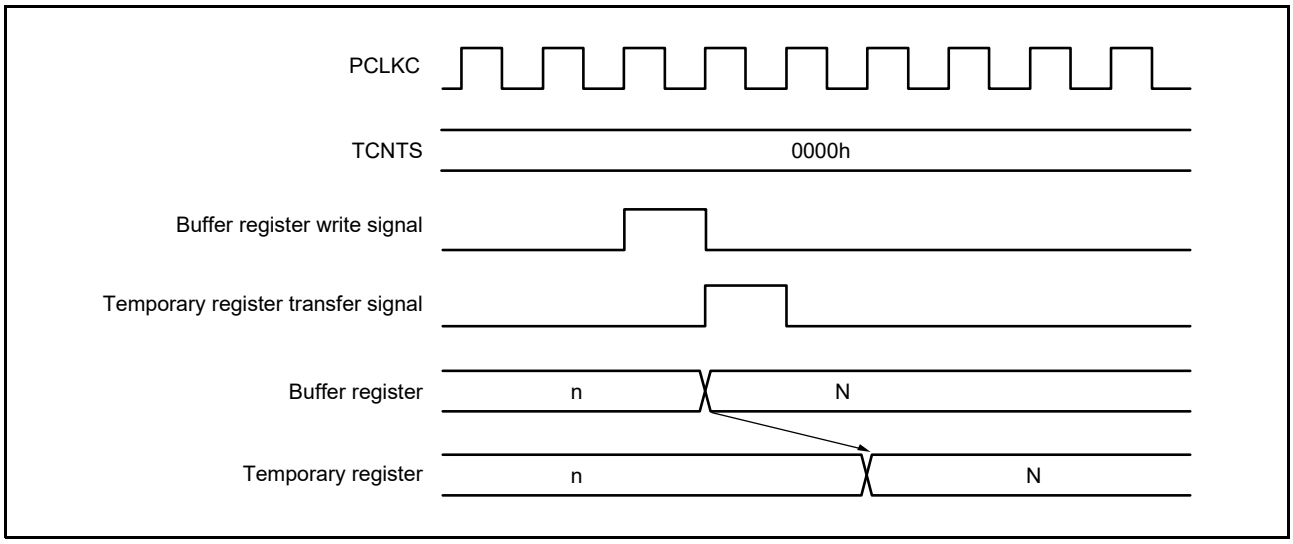


Figure 22.124 Transfer Timing from Buffer Register to Temporary Register (TCNTSA Stopped)

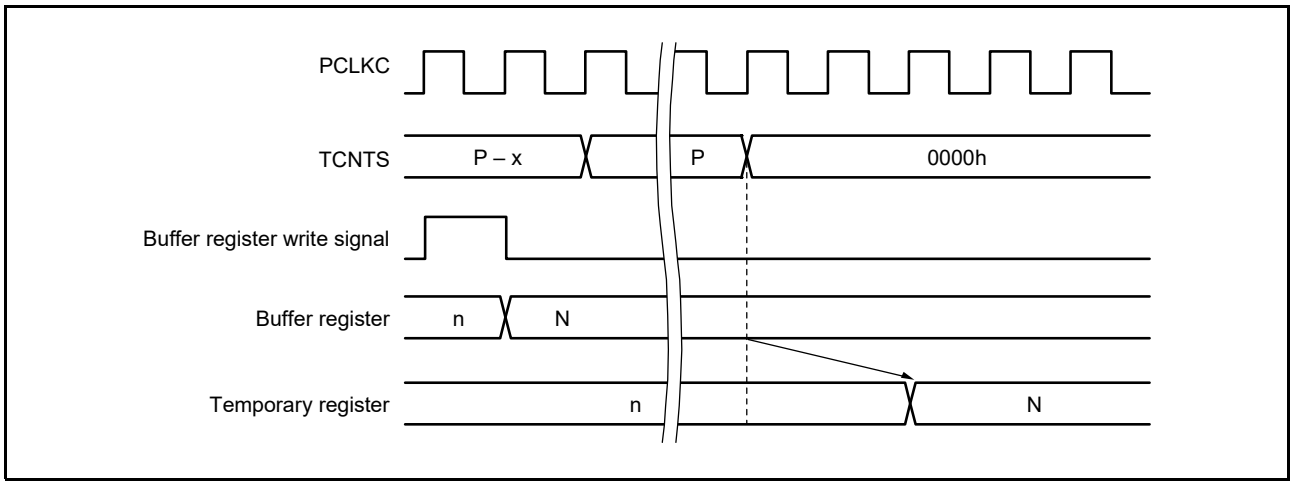


Figure 22.125 Transfer Timing from Buffer Register to Temporary Register (TCNTSA Operating)

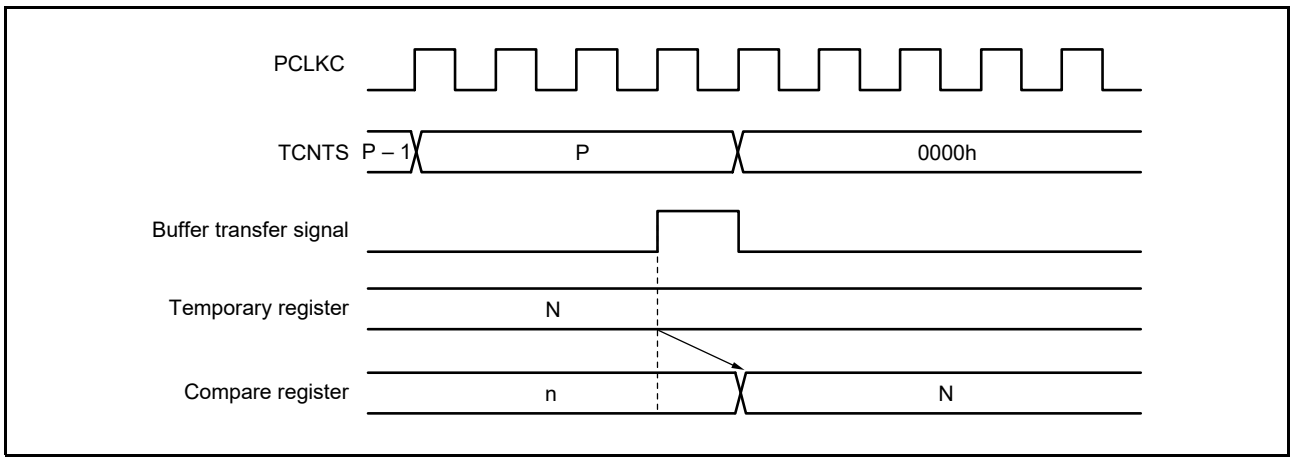


Figure 22.126 Transfer Timing from Temporary Register to Compare Register

22.5.2 Interrupt Signal Timing

(1) TGI Interrupt Timing by Compare Match

Figure 22.127 and Figure 22.128 show the TGI interrupt request signal timing when a compare match occurs.

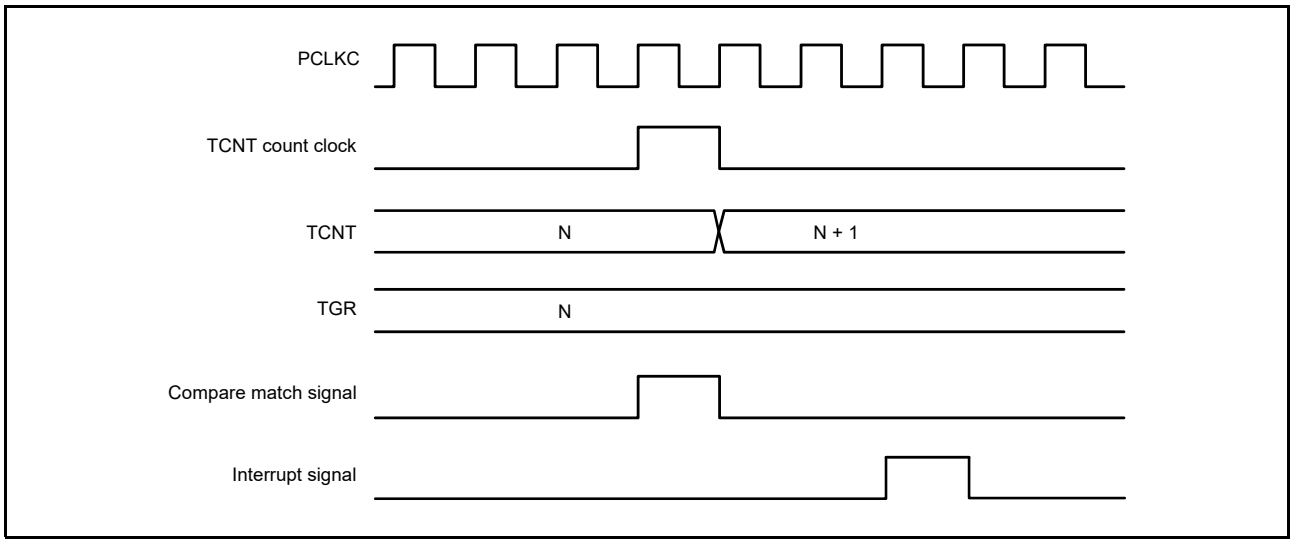


Figure 22.127 TGI Interrupt Timing (Compare Match) (MTU0 to MTU4, MTU6, MTU7, and MTU9)

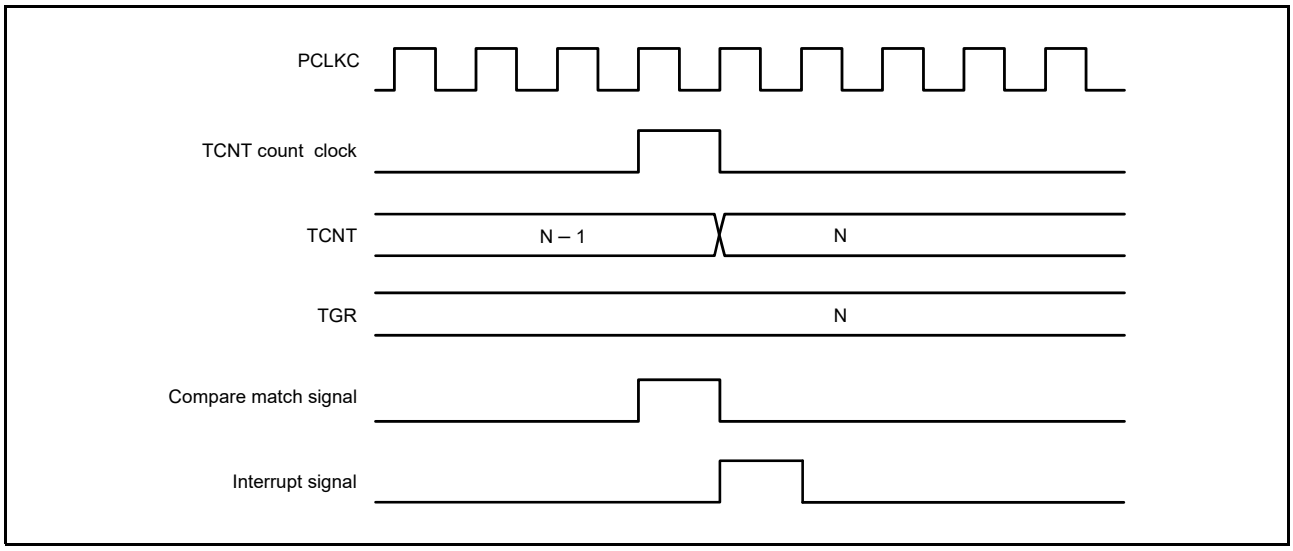


Figure 22.128 TGI Interrupt Timing (Compare Match) (MTU5)

(2) TGI Interrupt Timing by Input Capture

Figure 22.129 and Figure 22.130 show the TGI interrupt request signal timing when an input capture occurs.

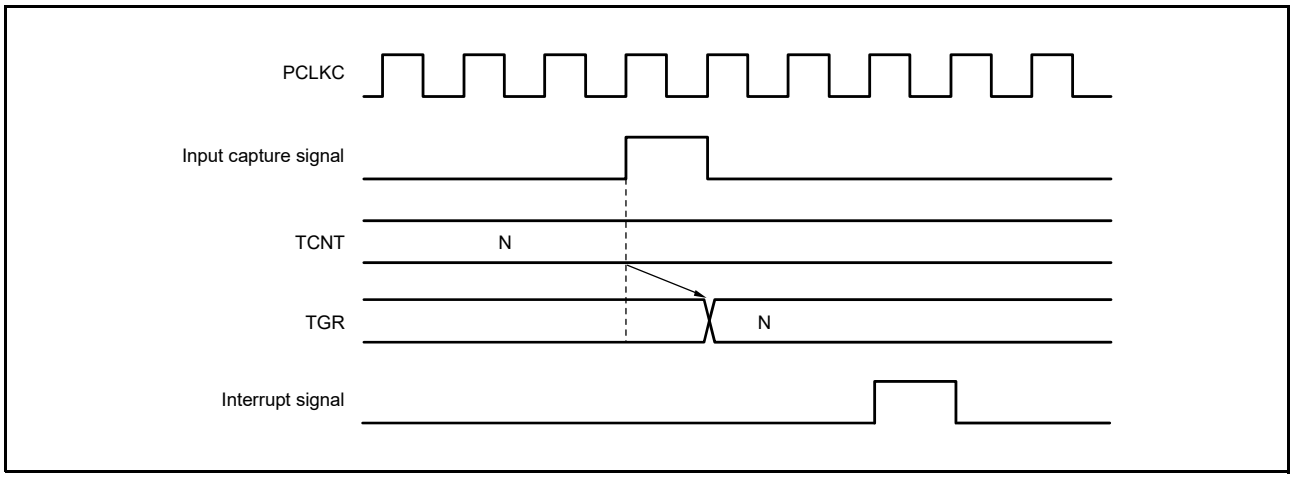


Figure 22.129 TGI Interrupt Timing (Input Capture) (MTU0 to MTU4, MTU6, MTU7, and MTU9)

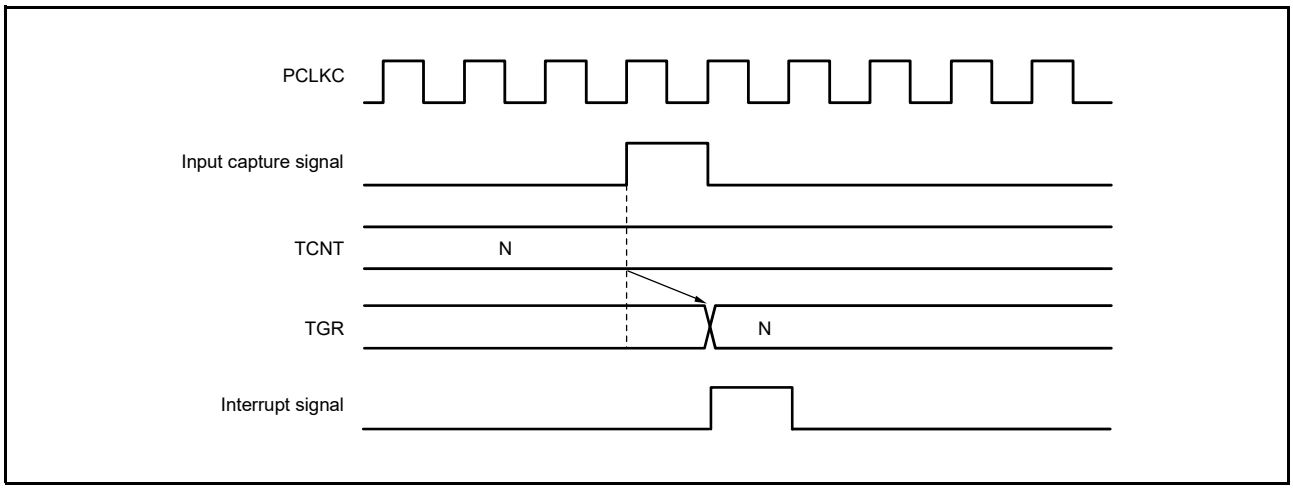


Figure 22.130 TGI Interrupt Timing (Input Capture) (MTU5)

(3) TCIV and TCIU Interrupt Timing

Figure 22.131 shows the TCIV interrupt request signal timing when an overflow is generated.
Figure 22.132 shows the TCIU interrupt request signal timing when an underflow is generated.

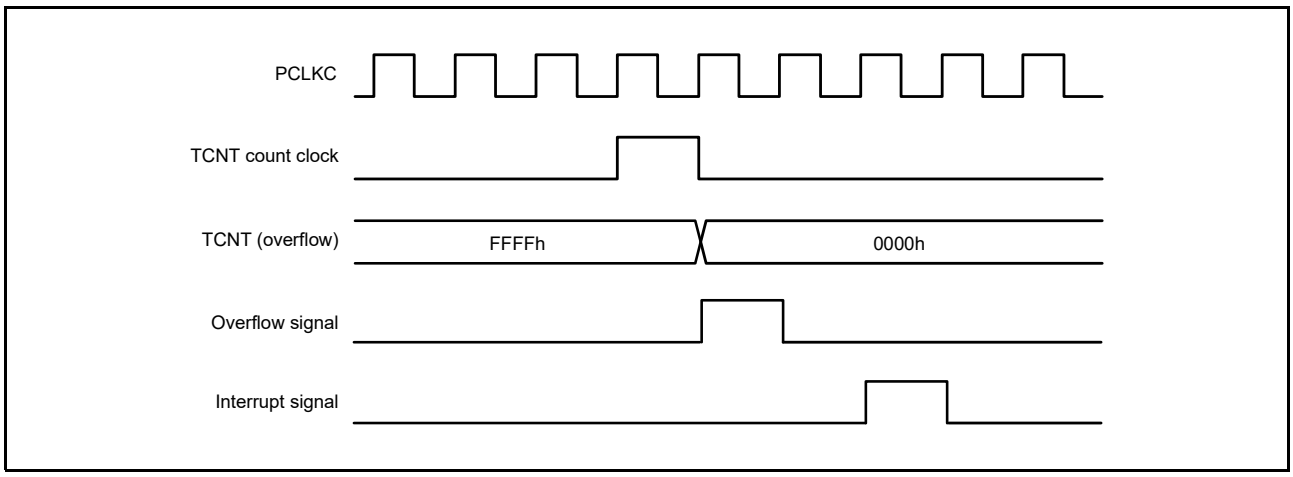


Figure 22.131 TCIV Interrupt Timing

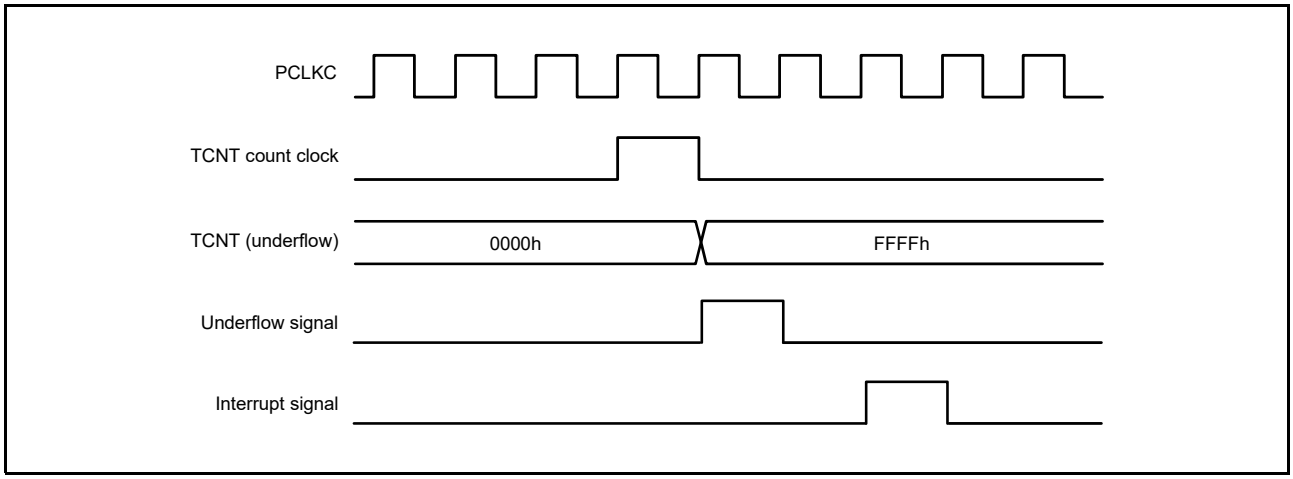


Figure 22.132 TCIU Interrupt Timing

22.6 Usage Notes

22.6.1 Module Stop Function Setting

MTU operation can be disabled or enabled using the module stop control register. MTU operation is stopped with the initial setting. Register access is enabled by releasing the module clock stop state. For details, refer to section 11, Low Power Consumption.

22.6.2 Count Clock Restrictions

The count clock source pulse width must be at least 1.5 PCLKC cycles for single-edge detection, and at least 2.5 PCLKC cycles for both-edge detection. The MTU will not operate properly at narrower pulse widths.
In phase counting mode, the phase difference and overlap between the two input clocks must be at least 1.5 PCLKC cycles, and the pulse width must be at least 2.5 PCLKC cycles. Figure 22.133 shows the input clock conditions in phase counting mode.

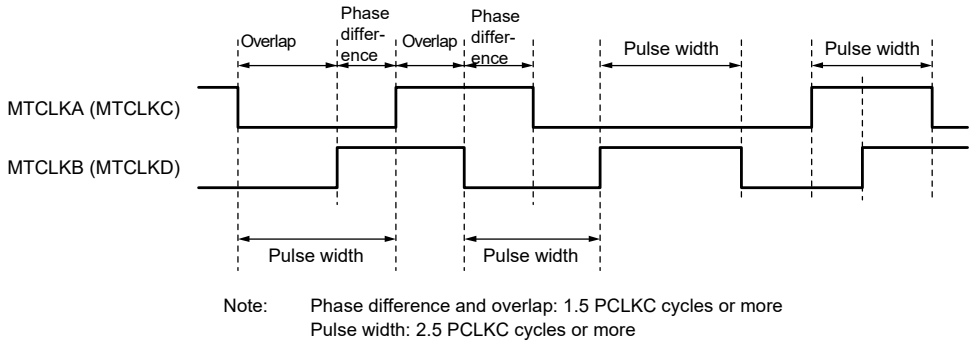


Figure 22.133 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

22.6.3 Note on Period Setting

When counter clearing on compare match is set, TCNT is cleared in the final state in which it matches the TGR value (the point at which TCNT updates the matched count value). Consequently, the actual counter frequency is given by the following formula:

- MTU0 to MTU4, MTU6, MTU7, and MTU9

$$f = \frac{CNTCLK}{N + 1}$$

- MTU5

$$f = \frac{CNTCLK}{N}$$

f: Counter frequency
CNTCLK: The count clock frequency set by TCR.TPSC[2:0] and TCR2.TPSC2[2:0]
N: TGR setting

22.6.4 Contention between TCNT Write and Clear Operations

If the counter clear signal is generated in the TCNT write cycle, TCNT clearing takes precedence and TCNT write operation is not performed.
Figure 22.134 shows the timing in this case.

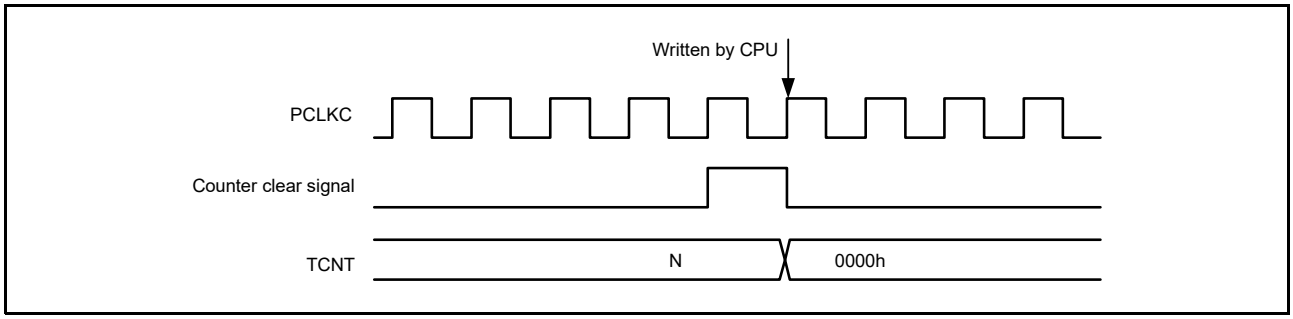


Figure 22.134 Contention between TCNT Write and Clear Operations

22.6.5 Contention between TCNT Write and Increment Operations

If incrementing occurs in a TCNT write cycle, TCNT write operation takes precedence and TCNT is not incremented.
Figure 22.135 shows the timing in this case.

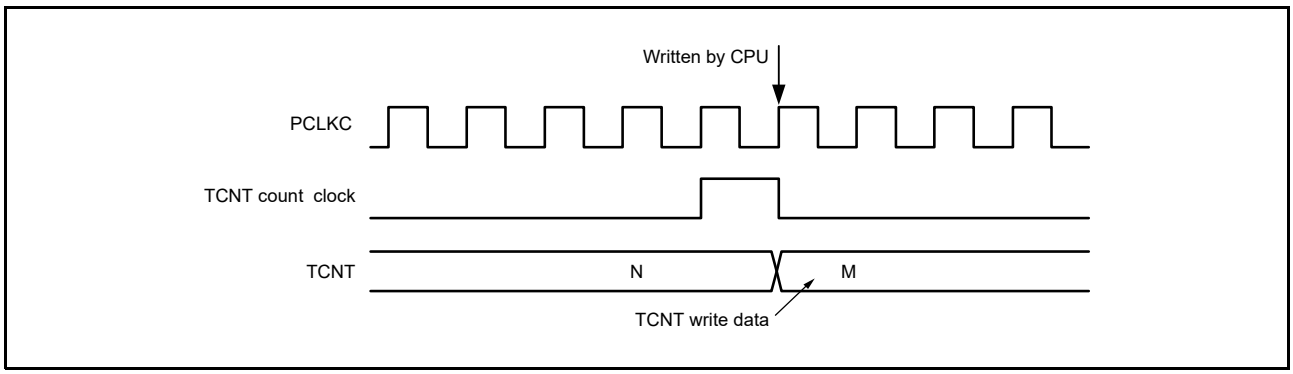


Figure 22.135 Contention between TCNT Write and Increment Operations

22.6.6 Contention between TGR Write Operation and Compare Match

If a compare match occurs in a TGR write cycle, TGR write operation is executed and the compare match signal is also generated.

Figure 22.136 shows the timing in this case.

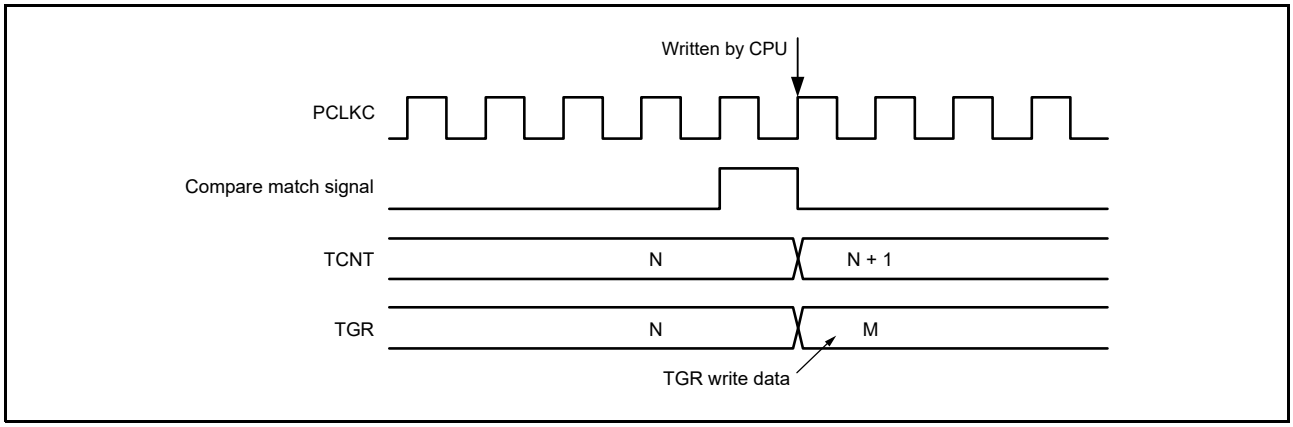


Figure 22.136 Contention between TGR Write Operation and Compare Match

22.6.7 Contention between Buffer Register Write Operation and Compare Match

If a compare match occurs in the T2 state in a TGR write cycle, the data before write operation is transferred to TGR by the buffer operation.

Figure 22.137 shows the timing in this case.

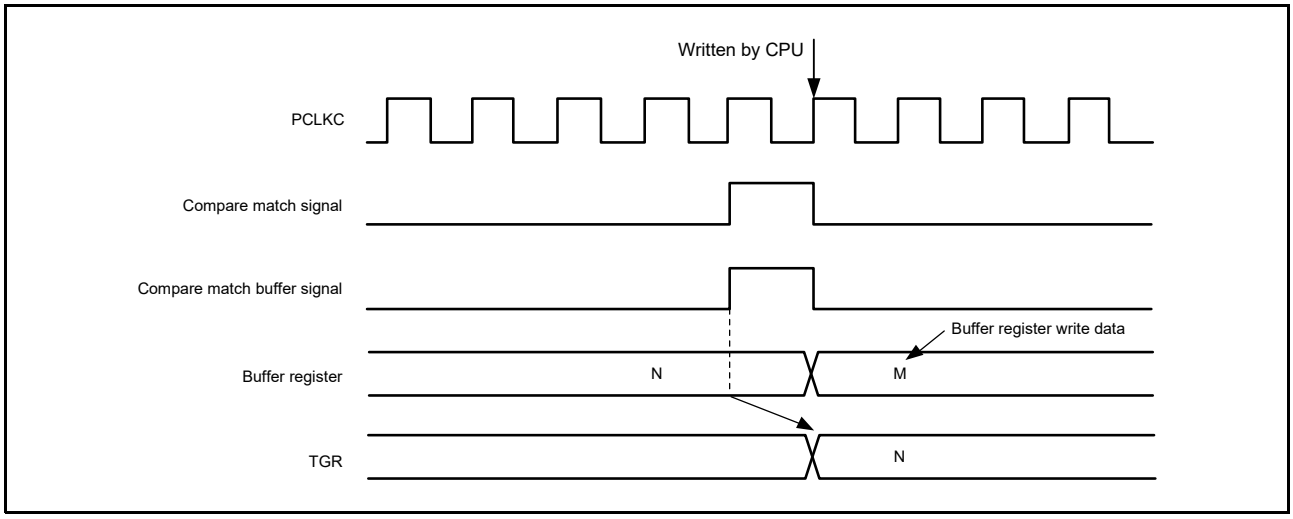


Figure 22.137 Contention between Buffer Register Write Operation and Compare Match

22.6.8 Contention between Buffer Register Write and TCNT Clear Operations

When the buffer transfer timing is set at the TCNT clear timing by the timer buffer transfer mode register (TBTM), if TCNT clearing occurs in the TGR write cycle, the data before write operation is transferred to TGR by the buffer operation.
Figure 22.138 shows the timing in this case.

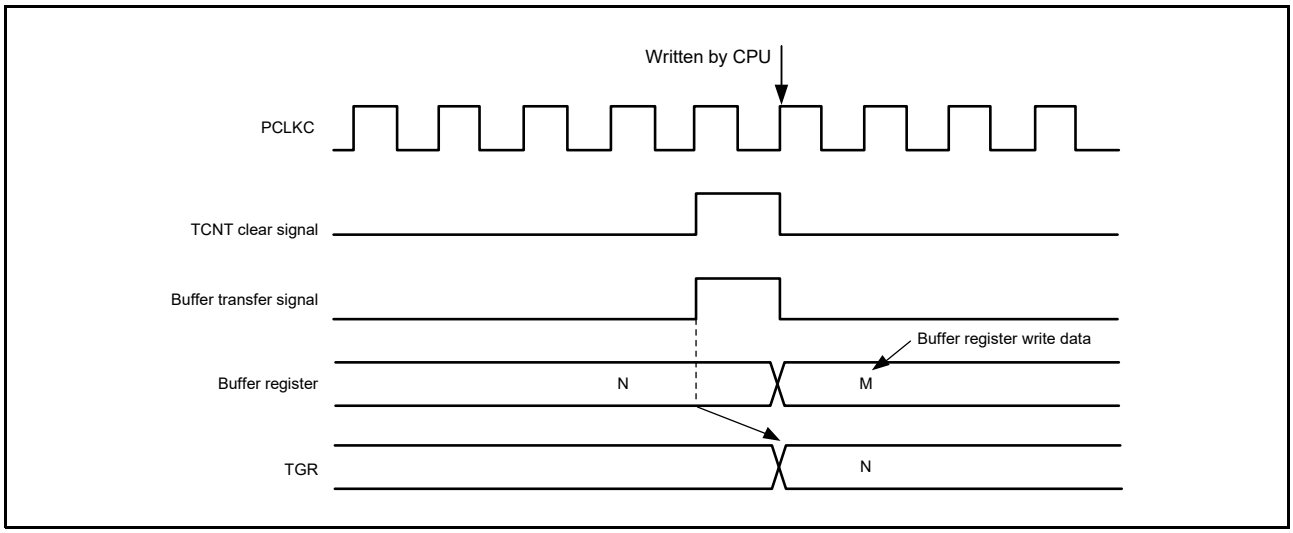


Figure 22.138 Contention between Buffer Register Write and TCNT Clear Operations

22.6.9 Contention between TGR Read Operation and Input Capture

If an input capture signal is generated in a TGR read cycle, the data before input capture transfer is read.
Figure 22.139 shows the timing in this case.

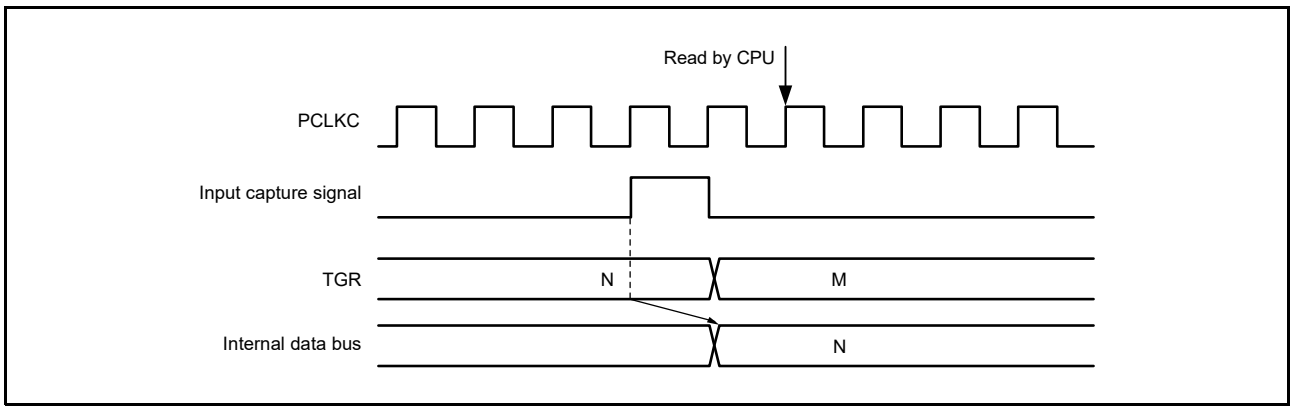


Figure 22.139 Contention between TGR Read Operation and Input Capture (MTU0 to MTU7, and MTU9)

22.6.10 Contention between TGR Write Operation and Input Capture

If an input capture signal is generated in the TGR write cycle, the input capture operation takes precedence and the TGR write operation is not performed in MTU0 to MTU4, MTU6, MTU7, and MTU9. In MTU5, the TGR write operation is performed and the input capture signal is generated.
Figure 22.140 and Figure 22.141 show the timing in this case.

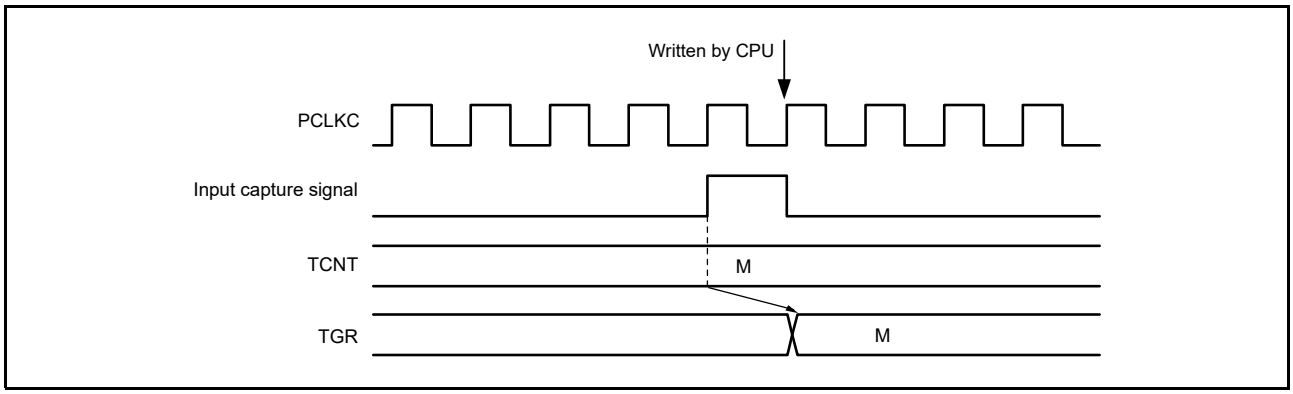


Figure 22.140 Contention between TGR Write Operation and Input Capture (MTU0 to MTU4, MTU6, MTU7, and MTU9)

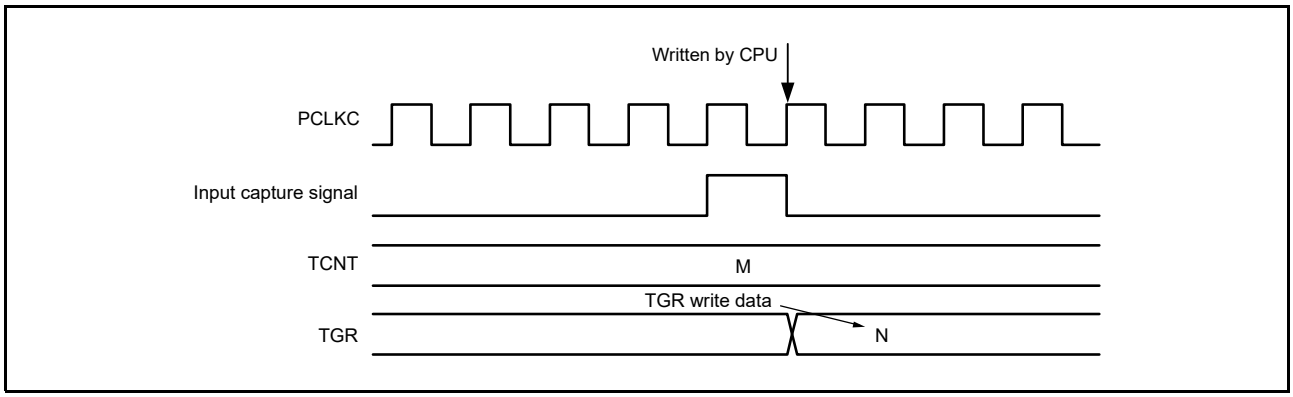


Figure 22.141 Contention between TGR Write Operation and Input Capture (MTU5)

22.6.11 Contention between Buffer Register Write Operation and Input Capture

If an input capture signal is generated in the buffer register write cycle, the buffer operation takes precedence and the buffer register write operation is not performed.
Figure 22.142 shows the timing in this case.

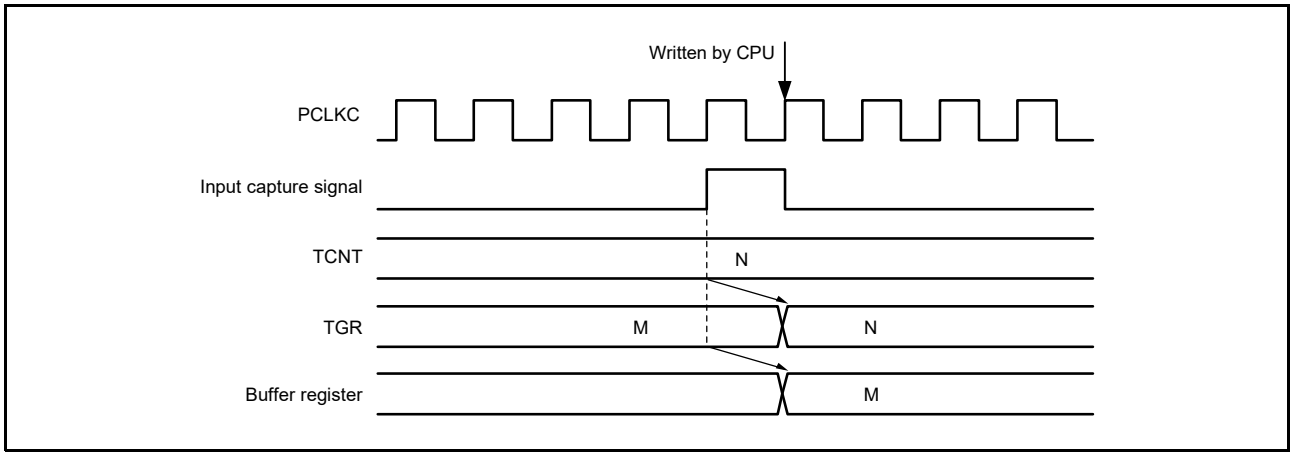


Figure 22.142 Contention between Buffer Register Write Operation and Input Capture

22.6.12 Contention between MTU2.TCNT Write Operation and Overflow/Underflow in Cascaded Operation

With timer counters MTU1.TCNT and MTU2.TCNT in a cascade, when a contention occurs between MTU1.TCNT counting (an MTU2.TCNT overflow/underflow) and the MTU2.TCNT write operation, the MTU2.TCNT write operation is performed and the MTU1.TCNT count signal is disabled. In this case, if MTU1.TGRA works as a compare match register and there is a match between the MTU1.TGRA and MTU1.TCNT values, a compare match signal is issued.

Furthermore, when the MTU1.TCNT (MTU2.TCNT) count clock is selected as the input capture source of MTU0 (MTU9), MTU0.TGRA to MTU0.TGRD (MTU9.TGRA to MTU9.TGRD) work in input capture mode. In addition, when the MTU0.TGRC compare match/input capture is selected as the input capture source of MTU1.TGRB, MTU1.TGRB works in input capture mode.

Figure 22.143 shows the timing in this case.

When setting the TCNT clearing function in cascaded operation, be sure to synchronize MTU1 and MTU2.

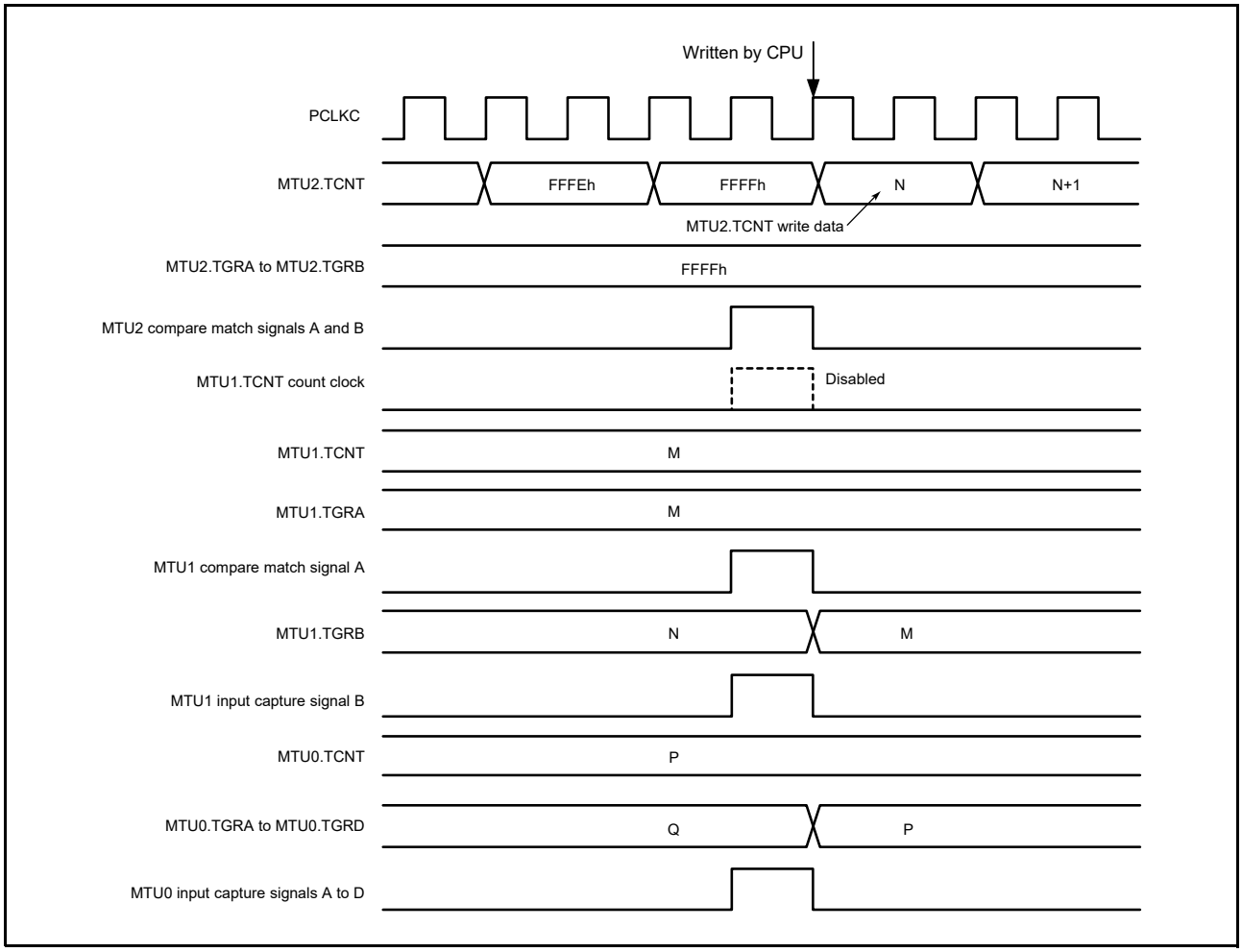


Figure 22.143 Contention between MTU2.TCNT Write Operation and Overflow/Underflow in Cascaded Operation

22.6.13 Counter Value When Count Operation is Stopped in Complementary PWM Mode

When counting operation in MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT) is stopped in complementary PWM mode, the MTU3.TCNT (MTU6.TCNT) value is set to the timer dead time register (TDDRA (TDDRB)) value and MTU4.TCNT (MTU7.TCNT) is set to 0000h.
When operation is restarted in complementary PWM mode, counting begins automatically from the initial setting state. Figure 22.144 shows this operation.
When counting begins in another operating mode, be sure to make initial settings in MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT).

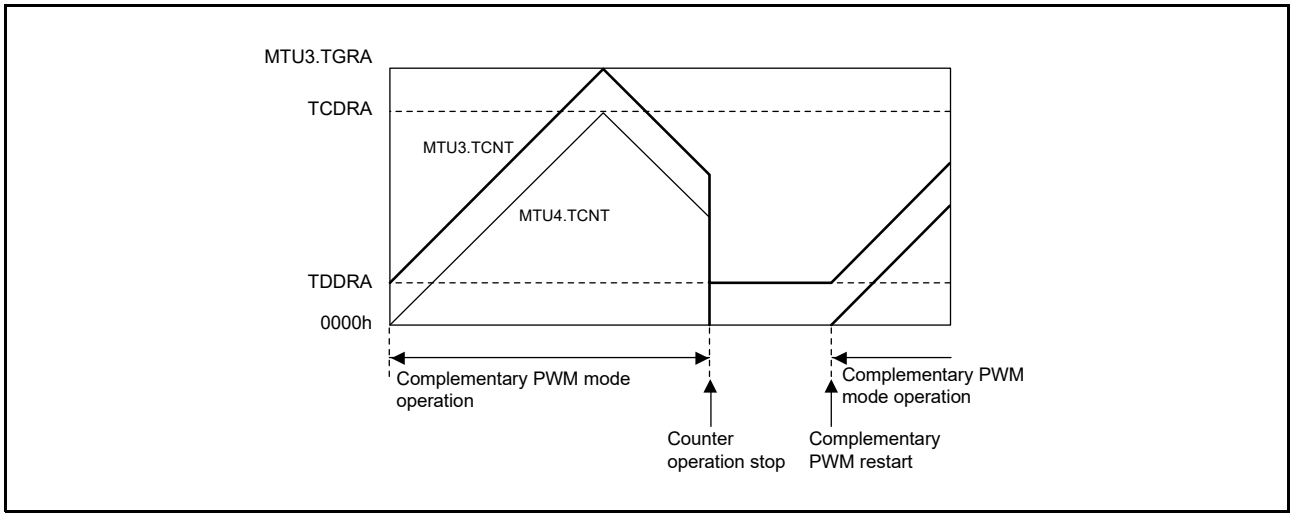


Figure 22.144 Counter Value When Stopped in Complementary PWM Mode

22.6.14 Buffer Operation Setting in Complementary PWM Mode

When modifying the PWM period set register (MTU3.TGRA or MTU6.TGRA), timer period data register (TCDRA or TCDRB), and duty set registers (MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB (MTU6.TGRB, MTU7.TGRA, and MTU7.TGRB)) in complementary PWM mode, be sure to use buffer operation. In addition, set the BFA and BFB bits in MTU4.TMDR1 (MTU7.TMDR1) to 0. The MTIOC4C (MTIOC7C) pin cannot output waveforms if the BFA bit in MTU4.TMDR1 (MTU7.TMDR1) is set to 1. Likewise, the MTIOC4D (MTIOC7D) pin cannot output waveforms if the BFB bit in MTU4.TMDR1 (MTU7.TMDR1) is set to 1.
In complementary PWM mode, buffer operation in MTU3 and MTU4 (or MTU6 and MTU7) depends on the settings in bits BFA and BFB of MTU3.TMDR1 (MTU6.TMDR1). When the BFA bit in MTU3.TMDR1 (MTU6.TMDR1) is set to 1, MTU3.TGRC (MTU6.TGRC) functions as a buffer register for MTU3.TGRA (MTU6.TGRA). At the same time, MTU4.TGRC (MTU7.TGRC) functions as a buffer register for MTU4.TGRA (MTU7.TGRA), and TCBRA (TCBRB) functions as a buffer register for TCDRA (TCDRB).

22.6.15 Buffer Operation and Compare Match in Reset-Synchronized PWM Mode

When setting buffer operation in reset-synchronized PWM mode, set the BFA and BFB bits in MTU4.TMDR1 (MTU7.TMDR1) to 0. The MTIOC4C (MTIOC7C) pin cannot output waveforms if the BFA bit in MTU4.TMDR1 (MTU7.TMDR1) is set to 1. Likewise, the MTIOC4D (MTIOC7D) pin cannot output waveforms if the BFB bit in MTU4.TMDR1 (MTU7.TMDR1) is set to 1.

In reset-synchronized PWM mode, buffer operation in MTU3 and MTU4 (or MTU 6 and MTU7) depends on the settings in the BFA and BFB bits of MTU3.TMDR1 (MTU6.TMDR1). For example, if the BFA bit in MTU3.TMDR1 (MTU6.TMDR1) is set to 1, MTU3.TGRC (MTU6.TGRC) functions as a buffer register for MTU3.TGRA (MTU6.TGRA). At the same time, MTU4.TGRC (MTU7.TGRC) functions as a buffer register for MTU4.TGRA (MTU7.TGRA).

While the MTU3.TGRC (MTU6.TGRC) and MTU3.TGRD (MTU6.TGRD) are operating as buffer registers, a TGImn interrupt (m = C, D; n = 3, 4 or 6, 7) is not generated.

Figure 22.145 shows an example of MTU3.TGR (MTU6.TGR), MTU4.TGR (MTU7.TGR), MTIOC3 (MTIOC6), and MTIOC4 (MTIOC7) operation with the BFA and BFB bits in MTU3.TMDR1 (MTU6.TMDR1) set to 1 and the BFA and BFB bits in MTU4.TMDR1 (MTU7.TMDR1) set to 0.

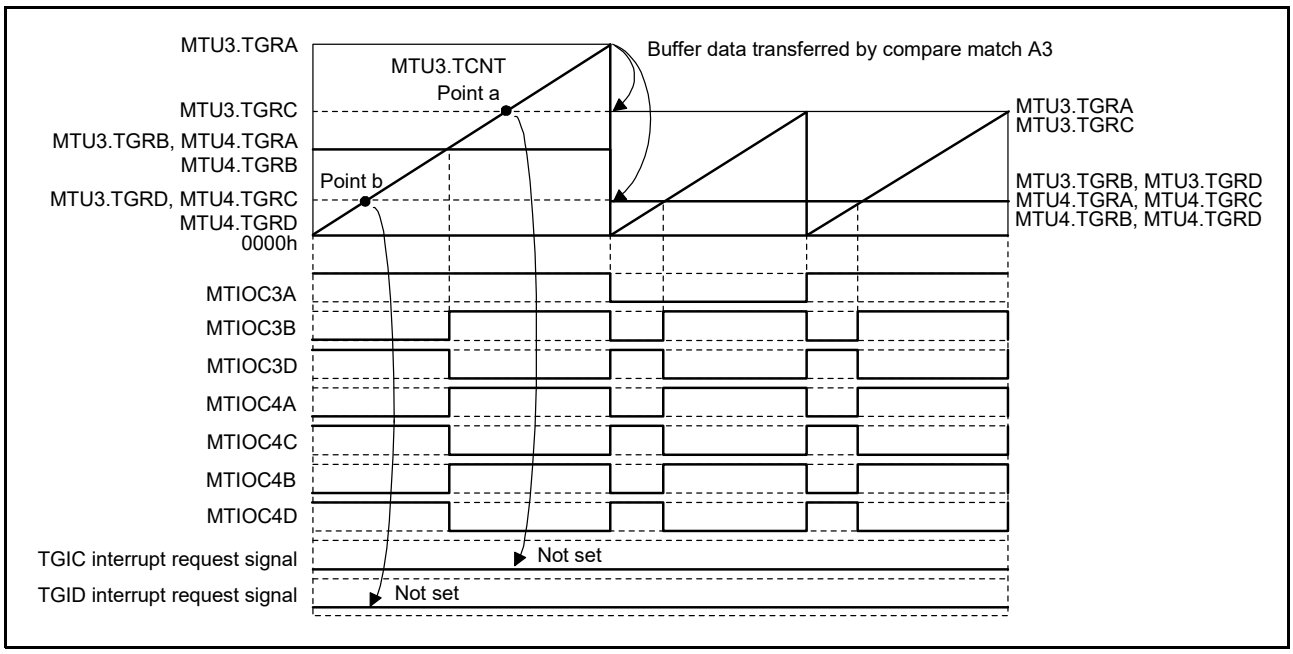


Figure 22.145 Buffer Operation and Compare Match in Reset-Synchronized PWM Mode

22.6.16 Overflow in Reset-Synchronized PWM Mode

After reset-synchronized PWM mode is selected, MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT) start counting when the CST3 (CST6) bit of TSTRA (TSTRB) is set to 1. In this state, the MTU4.TCNT (MTU7.TCNT) count clock source and count edge are determined by the MTU3.TCR (MTU6.TCR) setting.

In reset-synchronized PWM mode, with period register MTU3.TGRA (MTU6.TGRA) set to FFFFh and the MTU3.TGRA (MTU6.TGRA) compare match selected as the counter clearing source, MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT) count up to FFFFh, then a compare match occurs with MTU3.TGRA (MTU6.TGRA), and MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT) are both cleared. In this case, a TCIVn interrupt (n = 3, 4 or 6, 7) is not generated.

Figure 22.146 shows an example of operation in reset-synchronized PWM mode with period register MTU3.TGRA (MTU6.TGRA) set to FFFFh and the MTU3.TGRA (MTU6.TGRA) compare match specified for the counter clearing source.

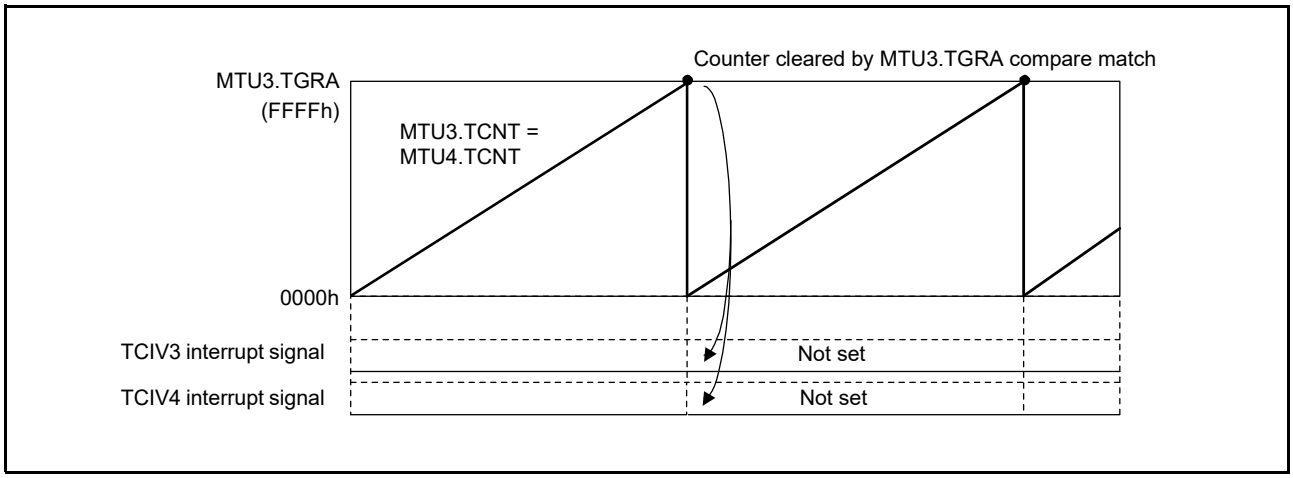


Figure 22.146 Overflow in Reset-Synchronized PWM Mode

22.6.17 Contention between Overflow/Underflow and Counter Clearing

If an overflow/underflow and counter clearing occur simultaneously, a TCIVn interrupt (n = 0 to 4, 6, 7, 9) nor a TCIUn interrupt (n = 1, 2) is not generated and TCNT clearing takes precedence.
Figure 22.147 shows the operation timing when a TGR compare match is specified as the clearing source and TGR is set to FFFFh.

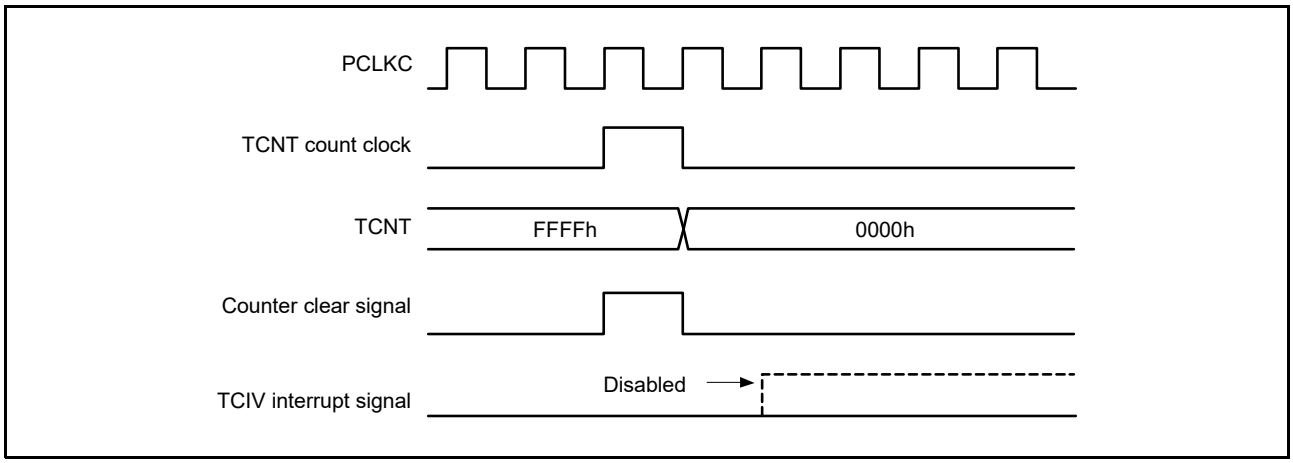


Figure 22.147 Contention between Overflow and Counter Clearing

22.6.18 Contention between TCNT Write Operation and Overflow/Underflow

If TCNT counts up or down in a TCNT write cycle and an overflow or an underflow occurs, the TCNT write operation takes precedence. A TCIVn interrupt (n = 0 to 4, 6, 7, 9) nor a TCIUn interrupt (n = 1, 2) is not generated.
Figure 22.148 shows the operation timing when there is contention between TCNT write operation and overflow.

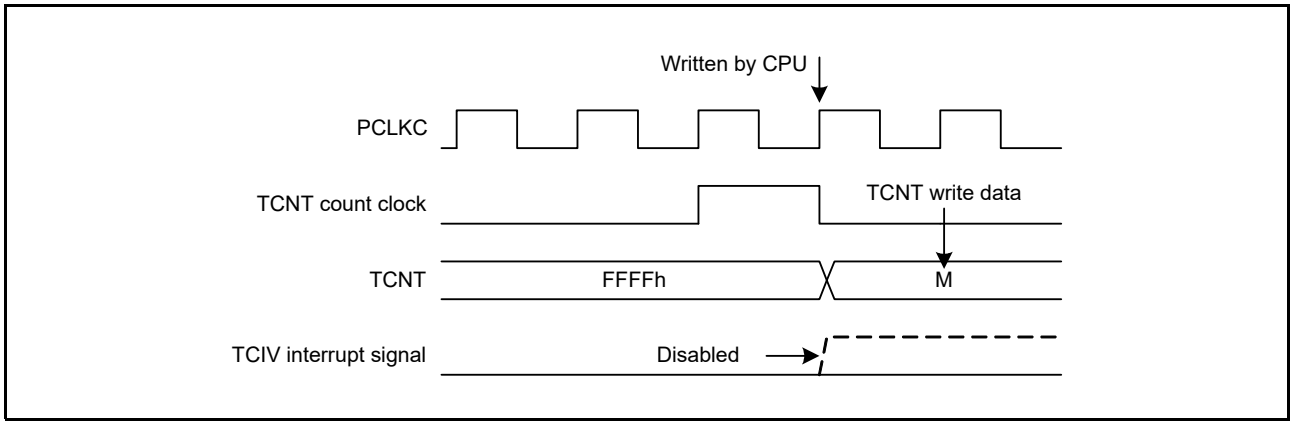


Figure 22.148 Contention between TCNT Write Operation and Overflow

22.6.19 Note on Transition from Normal Mode or PWM Mode 1 to Reset-Synchronized PWM Mode

When making a transition from normal mode or PWM mode 1 to reset-synchronized PWM mode in MTU3 and MTU4 (or MTU6 and MTU7), if the counter is stopped while the output pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D, MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, and MTIOC7D) are held at a high level and then operation is started after a transition to reset-synchronized PWM mode, the initial pin output will not be correct.

When making a transition from normal mode to reset-synchronized PWM mode, write 11h to MTU3.TIORH, MTU3.TIORL, MTU4.TIORH, and MTU4.TIORL (MTU6.TIORH, MTU6.TIORL, MTU7.TIORH, and MTU7.TIORL) to initialize the output pin state to a low level, then set the registers to the initial value (00h) before making the mode transition.

When making a transition from PWM mode 1 to reset-synchronized PWM mode, switch to normal mode, initialize the output pin state to a low level, and then set the registers to the initial value (00h) before making the transition to reset-synchronized PWM mode.

22.6.20 Output Level in Complementary PWM Mode and Reset-Synchronized PWM Mode

When MTU3 and MTU4 (or MTU6 and MTU7) are in complementary PWM mode or reset-synchronized PWM mode, the PWM waveform output level is determined by the OLSP and OLSN bits in the timer output control register (TOCR1A or TOCR1B). In complementary PWM mode or reset-synchronized PWM mode, TIOR should be set to 00h. The output level in negative phase when the TDERA.TDER (TDERB.TDER) bit is set to 0 in complementary PWM mode (the dead time is not generated) does not depend on the setting of the TOCR1A.OLSN (TOCR1B.OLSN) bit. It is equivalent to the inverted level of positive phase output based on the setting of the TOCR1A.OLSP (TOCR1B.OLSP) bit.

22.6.21 Simultaneous Input Capture in MTU1.TCNT and MTU2.TCNT in Cascade Connection

When timer counters 1 and 2 (MTU1.TCNT and MTU2.TCNT) operate as a 32-bit counter in cascade connection, the cascaded counter value cannot be captured successfully in some cases even if input-capture input is simultaneously done to MTIOC1A and MTIOC2A or to MTIOC1B and MTIOC2B. This is because the input timing of MTIOC1A and MTIOC2A or to MTIOC1B and MTIOC2B may not be the same when external input-capture signals input into MTU1.TCNT and MTU2.TCNT are taken in synchronization with the internal clock.

For example, MTU1.TCNT (the counter for upper 16 bits) does not capture the count-up value by an overflow from MTU2.TCNT (the counter for lower 16 bits) but captures the count value before the up-counting. In this case, the values of MTU1.TCNT = FFF1h and MTU2.TCNT = 0000h should be transferred to MTU1.TGRA and MTU2.TGRA or to MTU1.TGRB and MTU2.TGRB, but the values of MTU1.TCNT = FFF0h and MTU2.TCNT = 0000h are erroneously transferred.

The MTU has a function that allows simultaneous capture of MTU1.TCNT and MTU2.TCNT with a single input capture input. This function can be used to read the 32-bit counter such that MTU1.TCNT and MTU2.TCNT are captured at the same time. For details, refer to section 22.2.11, Timer Input Capture Control Register (TICCR).

22.6.22 Interrupt Skipping Function 2

When interrupt skipping function 2 is in use and the difference between the values in MTU4.TADCORA and MTU4.TADCORB is small, correct counting of the number skipped may not be possible, in which case requests for A/D conversion will not be generated with the expected timing. The conditions listed below thus apply to these settings.
For MTU6 and MTU7, the same conditions apply to the settings of MTU7.TADCORA and MTU7.TADCORB.

(1) When the number skipped is zero for skipping function 2

- The difference between the values in MTU4.TADCORA and MTU4.TADCORB must be at least four.
- The interval of comparison for MTU4.TADCORA must be at least four cycles of PCLKC (the updated value of MTU4.TADCORA is set to the previous value plus or minus at least four).
- The interval of comparison for MTU4.TADCORB must be at least four cycles of PCLKC (the updated value of MTU4.TADCORB is set to the previous value plus or minus at least four).

(2) When the number skipped is one or more for skipping function 2

- The difference between the values in MTU4.TADCORA and MTU4.TADCORB must be at least two.
- The interval of comparison for MTU4.TADCORB must be at least two cycles of PCLKC (the updated value of MTU4.TADCORB is set to the previous value plus or minus at least two).

22.6.23 Notes When Complementary PWM Mode Output Protection Function is Not Used

The complementary PWM mode output protection function is initially enabled. For details, refer to section 23, Port Output Enable 3 (POE3D).

22.6.24 Notes Regarding Timer Counter (MTU5.TCNT) and Timer General Register (MTU5.TGR)

Do not set an MTU5.TGRm (m = U, V, W) bit to the value of the corresponding MTU5.TCNTm (m = U, V, W) plus one while counting by the MTU5.TCNTm (m = U, V, W) register is stopped. If an MTU5.TGRm (m = U, V, W) bit is set to the value of the corresponding MTU5.TCNTm (m = U, V, W) plus one while counting by the MTU5.TCNTm (m = U, V, W) is stopped, a compare-match will be generated even though counting is stopped.
In this case, if the compare match enable bit (MTU5.TIER.TGIE5m (m = U, V, W) bit is set to 1 (enabling interrupts), a compare-match interrupt will also be generated. If the value of the timer compare match clear register is 1 (enabled), the timer is automatically cleared to 0000h when the compare-match is generated, regardless of whether interrupts from the MTU5.TCNTm (m = U, V, W) are enabled or disabled.

22.6.25 Notes to Prevent Malfunctions in Synchronous Clearing for Complementary PWM Mode

If control of the output waveform is enabled (TWCRA.WRE bit = 1 or TWCRB.WRE bit = 1) at the time of synchronous counter clearing in complementary PWM mode, satisfaction of either condition 1 or 2 below has the following effects.

- Dead time on the PWM output pins is shortened (or disappears).
- The active level is output on the negative phase PWM output pins beyond the period for active-level output.

Condition 1: In portion (10) of the initial output inhibition period in Figure 22.149, synchronous clearing occurs within the dead-time period for PWM output.

Condition 2: In portions (10) and (11) of the initial output inhibition period in Figure 22.150, synchronous clearing occurs when any condition from among MTU3.TGRB (MTU6.TGRB) ≤ TDDRA (TDDRB), MTU4.TGRA (MTU7.TGRA) ≤ TDDRA (TDDRB), or MTU4.TGRB (MTU7.TGRB) ≤ TDDRA (TDDRB) is satisfied.

The following method avoids the above phenomena.
Ensure that synchronous clearing proceeds with the value of each comparison register (MTU3.TGRB (MTU6.TGRB), MTU4.TGRA (MTU7.TGRA), and MTU4.TGRB (MTU7.TGRB)) set to at least double the value of the TDDRA register (TDDRB register).

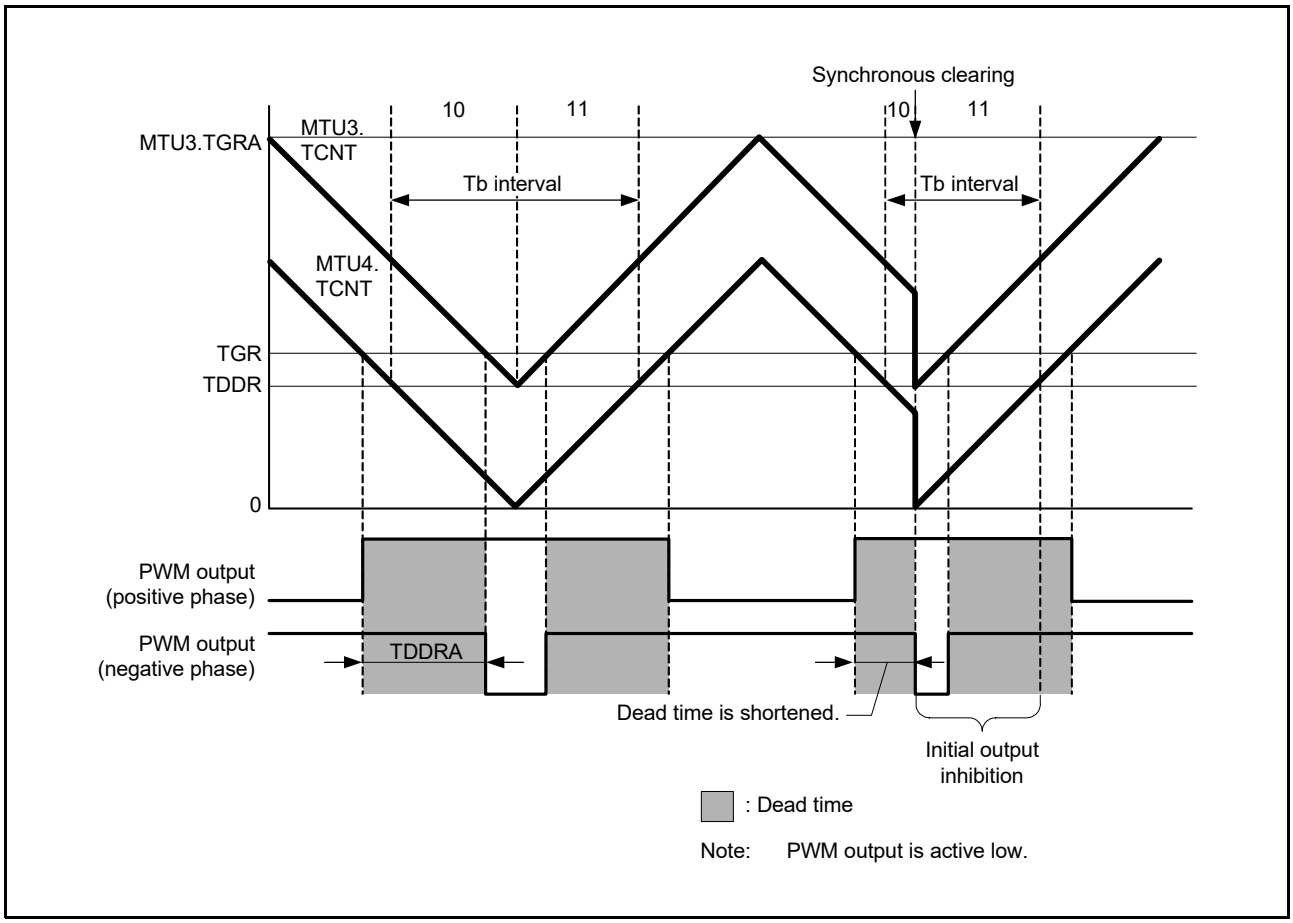


Figure 22.149 Example of Synchronous Clearing (When Condition 1 Applies)

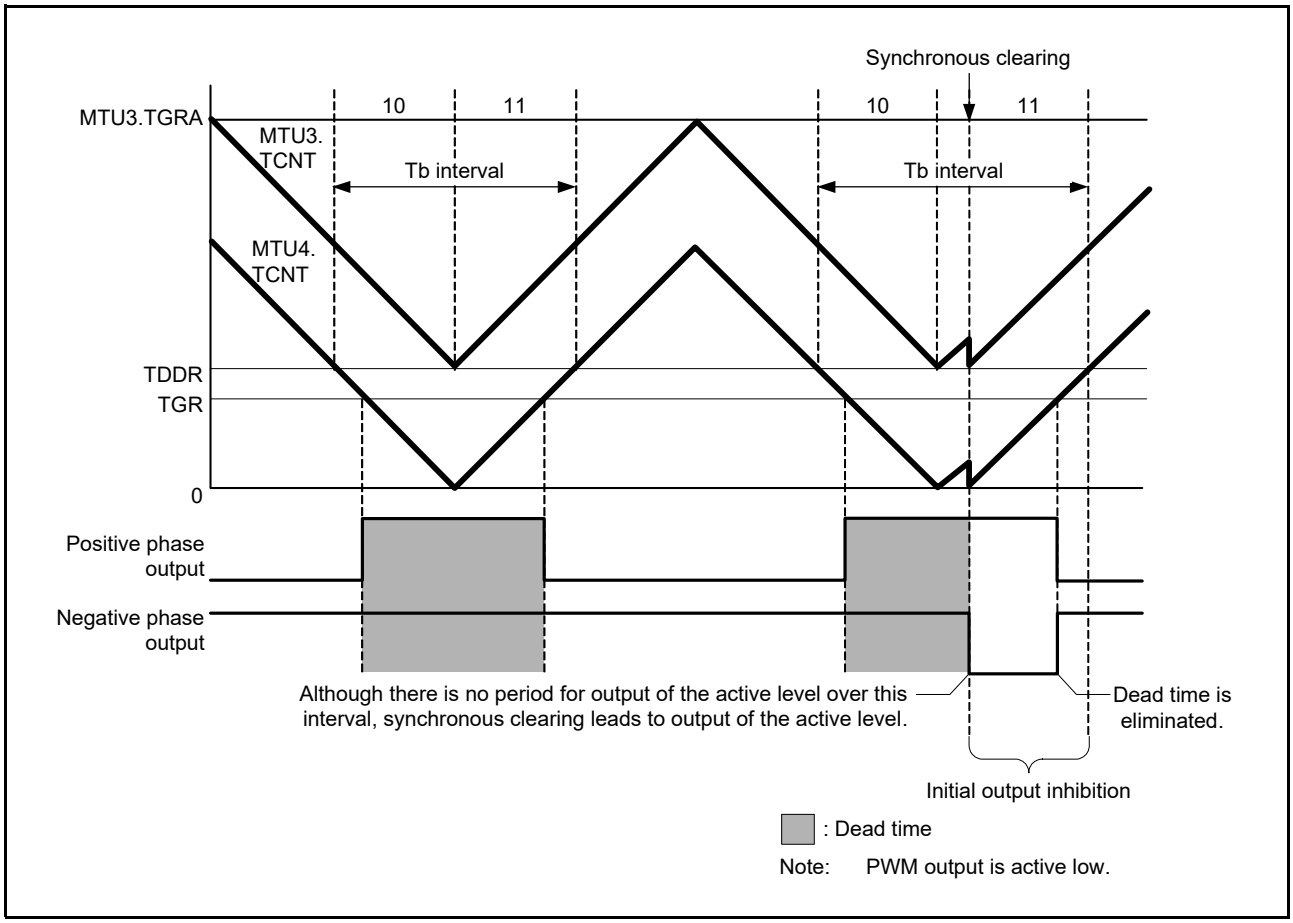


Figure 22.150 Example of Synchronous Clearing (When Condition 2 Applies)

22.6.26 Notes on Timer Mode Register Setting for ELC Event Input

When MTU is used in ELC operation, set the timer mode register (TMDR) of the relevant channel to its initial value (00h).

22.6.27 Continuous Output of Interrupt Signal in Response to a Compare Match

When the TGR register is set to 0000h, the PCLKC/1 is set as the count clock, and compare match is set as the trigger for clearing of the count clock, the value of the TCNT counter remains 0000h, and the interrupt signal will be output continuously (i.e. its level will be flat) rather than output over a single cycle. Consequently, interrupts will not be detected in response to second and subsequent compare matches.
Figure 22.151 shows the timing for continuous output of the interrupt signal in response to a compare match.

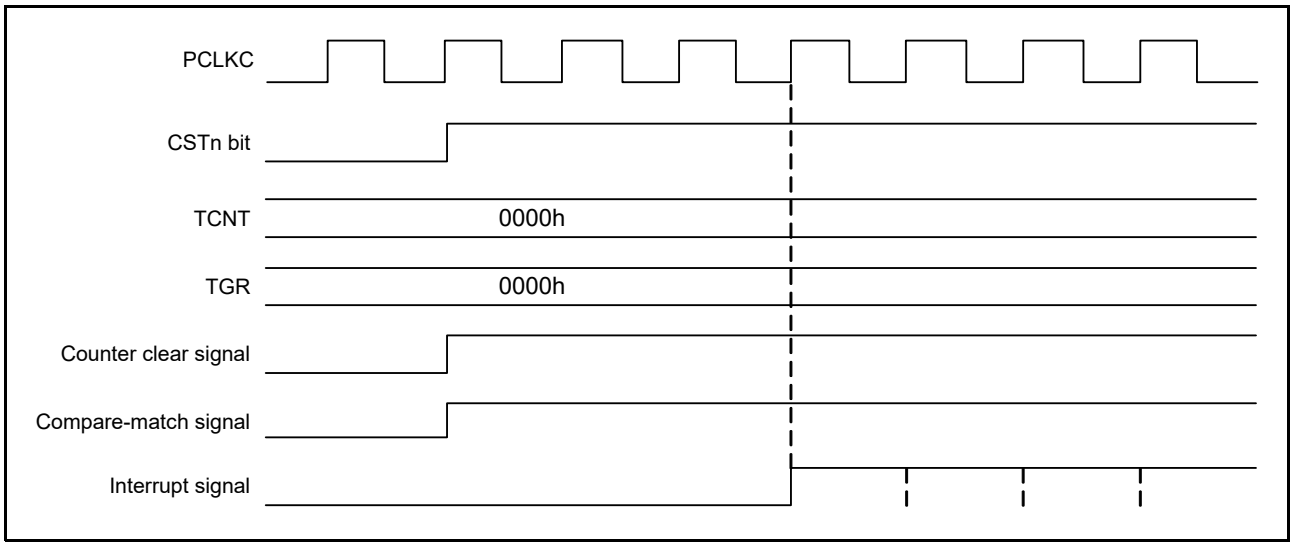


Figure 22.151 Continuous Output of Interrupt Signal in Response to a Compare Match

22.6.28 Usage Notes on A/D Conversion Start Request Delaying Function in Complementary PWM Mode

- When data is transferred from a buffer register at the trough of the MTU4.TCNT (MTU7.TCNT) counter while the MTU4.TADCOBRA and MTU4.TADCOBRB (MTU7.TADCOBRA and MTU7.TADCOBRB) registers are set to 0 and the UT4AE and UT4BE (UT7AE and UT7BE) bits in the MTU4.TADCR (MTU7.TADCR) register are set to 1, no A/D conversion start request is issued during up-counting immediately after transfer. Refer to Figure 22.152.
- When data is transferred from a buffer register at the crest of the MTU4.TCNT (MTU7.TCNT) counter while the MTU4.TADCOBRA and MTU4.TADCOBRB (MTU7.TADCOBRA and MTU7.TADCOBRB) registers are set to the same value as the TCDR and the UT4AE and UT4BE (UT7AE and UT7BE) bits in the MTU4.TADCR (MTU7.TADCR) register are set to 1, no A/D conversion start request is issued during down-counting immediately after transfer. Refer to Figure 22.153.
- To issue an A/D conversion start request linked with the interrupt skipping function, set the MTU4.TADCORA and MTU4.TADCORB (MTU7.TADCORA and MTU7.TADCORB) registers so that $2 \leq$ MTUn.TADCORA/TADCORB $\leq$ TCDR $- 2$ is satisfied (n = 4, 7).

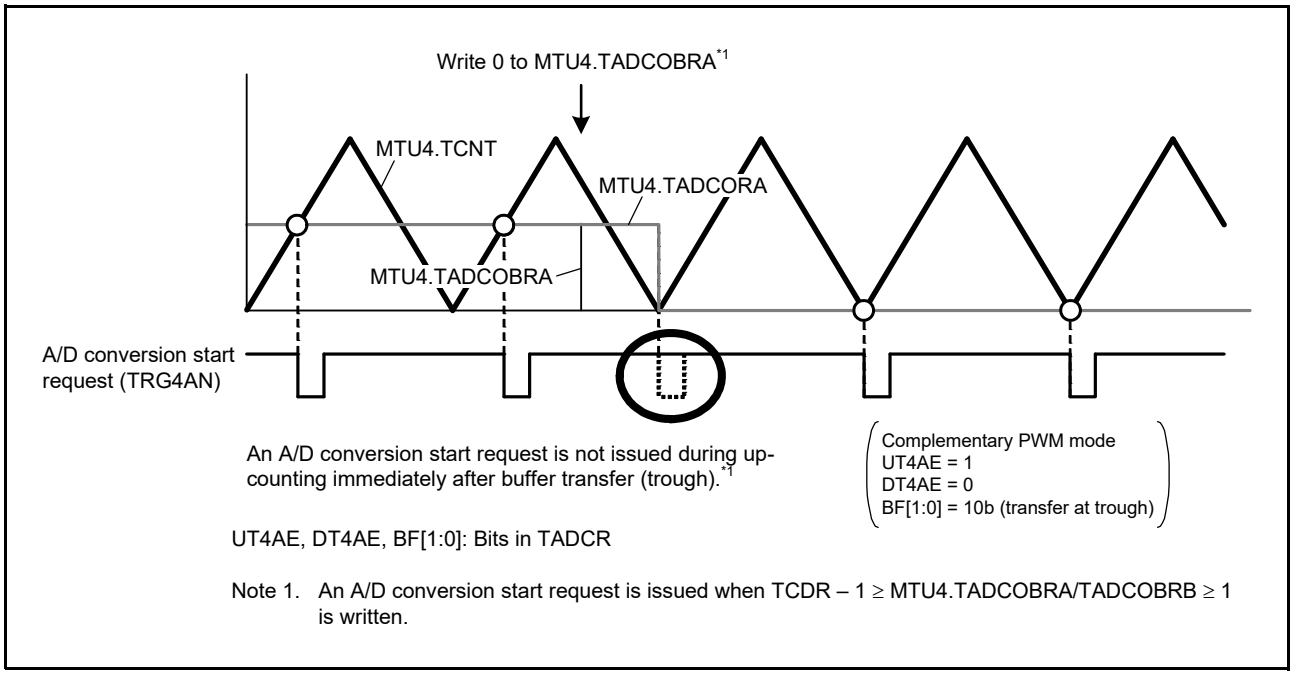


Figure 22.152 A/D Conversion Start Request When 0 is Written to MTU4.TADCOBRA

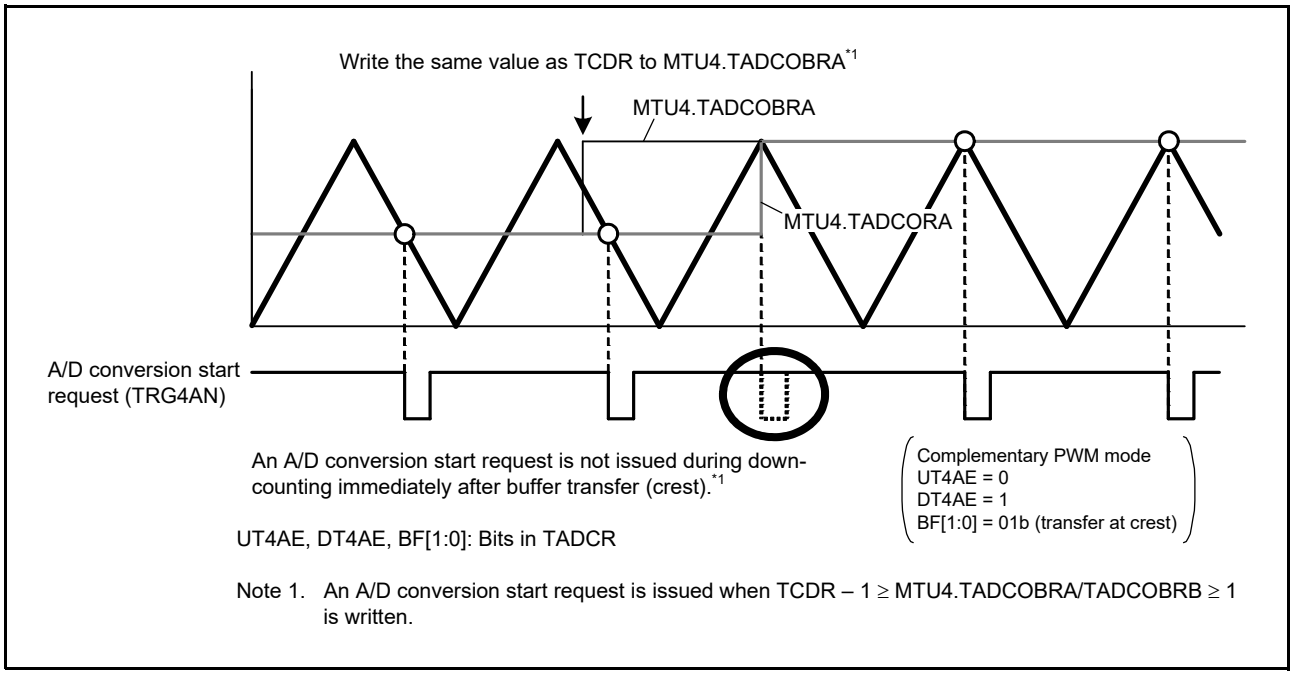


Figure 22.153 A/D Conversion Start Request When the Same Value as TCDR is Written to MTU4.TADCOBRA

22.7 MTU Output Pin Initialization

22.7.1 Operating Modes

The MTU has the following six operating modes. Waveforms can be output in any of these modes.

- Normal mode (MTU0 to MTU4, MTU6, MTU7, and MTU9)
- PWM mode 1 (MTU0 to MTU4, MTU6, MTU7, and MTU9)
- PWM mode 2 (MTU0 to MTU2, and MTU9)
- Phase counting modes 1 to 5 (MTU1 and MTU2)
- Complementary PWM mode (MTU3, MTU4, MTU6, and MTU7)
- Reset-synchronized PWM mode (MTU3, MTU4, MTU6, and MTU7)

This section describes how to initialize the MTU output pins in each of these modes.

22.7.2 Operation in Case of Re-Setting Due to Error during Operation

If an error occurs during MTU operation, MTU output should be cut off by the system. The output can be cut off by allowing non-active level output from the pins by setting the pins as general output ports using the port direction registers (PDR), port output data registers (PODR), and port mode registers (PMR) of the I/O ports. MTU output can be disabled through TIOR settings. Complementary PWM output (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, MTIOC4D, MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7B, MTIOC7C, and MTIOC7D) should be specified through TOERA and TOERB settings. For PWM output pins, output can also be cut by hardware, using port output enable 3 (POE3). The pin initialization procedures for re-setting due to an error during operation and the procedures for restarting in a different mode after re-setting are described below.

The MTU has six operating modes, as stated above. There are thus 36 mode transition combinations, but some transitions are not available with certain channel and mode combinations. Available mode transition combinations are shown in Table 22.80.

Table 22.80 Mode Transition Combinations

	Normal	PWM1	PWM2	PCM	CPWM	RPWM
Normal	(1)	(2)	(3)	(4)	(5)	(6)
PWM1	(7)	(8)	(9)	(10)	(11)	(12)
PWM2	(13)	(14)	(15)	(16)	Not available	Not available
PCM	(17)	(18)	(19)	(20)	Not available	Not available
CPWM	(21)	(22)	Not available	Not available	(23) (24)	(25)
RPWM	(26)	(27)	Not available	Not available	(28)	(29)

Normal: Normal mode
PWM1: PWM mode 1
PWM2: PWM mode 2
PCM: Phase counting modes 1 to 5
CPWM: Complementary PWM mode
RPWM: Reset-synchronized PWM mode

22.7.3 Overview of Initialization Procedures and Mode Transitions in Case of Error during Operation

- When making a transition to a mode (Normal, PWM1, PWM2, or PCM) in which the pin output level is selected by the timer I/O control register (TIOR) setting, initialize the pins by means of TIOR setting.
- In PWM mode 1, waveforms are not output to the MTIOCnB and MTIOCnD (n = 3, 4, 6, 7) pins. When a pin is configured for MTIOCnB or MTIOCnD, it enters high-impedance state. To output a specified level, set the pin to general output port.
- In PWM mode 2, a waveform is not output to the pin that corresponds to the TGR register used as a period register. When a pin is configured for MTIOCnm (n = 0 to 2, 9; m = A to D), it enters high-impedance state. To output a specified level, set the pin to general output port.
- In normal mode or PWM mode 2, if TGRC and TGRD operate as buffer registers, waveforms are not output to the corresponding pins (MTIOCnC or MTIOCnD (n = 0, 3, 4, 6, 7, 9)). When a pin is configured for MTIOCnC or MTIOCnD, it enters high-impedance state. To output a specified level, set the pin to general output port.
- In PWM mode 1, if either TGRC or TGRD operates as a buffer register, waveforms are not output to the corresponding pins (MTIOCnC or MTIOCnD (n = 0, 3, 4, 6, 7, 9)). When a pin is configured for MTIOCnC or MTIOCnD, it enters high-impedance state. To output a specified level, set the pin to general output port.
- When making a transition to a mode (CPWM or RPWM) in which the pin output level is selected by the timer output control register (TOCR1A, TOCR2A, TOCR1B, or TOCR2B) setting, temporarily disable output in MTU3 and MTU4 (or MTU6 and MTU7) with the timer output master enable register (TOERA or TOERB). At this time, when a pin is configured for MTIOCnm (n = 3, 4, 6, 7; m = A to D), it enters high-impedance state. To output a specified level, set the pin to general output port. Switch to normal mode, perform initialization with TIOR, restore TIOR to its initial value, then operate the MTU in accordance with the mode setting procedure (TOCR1A setting, TOCR2A setting, TMDR1 setting, and TOERA setting (TOCR1B setting, TOCR2B setting, TMDR1 setting, and TOERB setting)).

Note: Channel number is substituted for “n” indicated in this section.

Pin initialization procedures are described below for the numbered combinations in Table 22.80. The active level is assumed to be low.

(1) Operation When Error Occurs in Normal Mode and Operation is Restarted in Normal Mode

Figure 22.154 shows a case in which an error occurs in normal mode and operation is restarted in normal mode after re-setting.

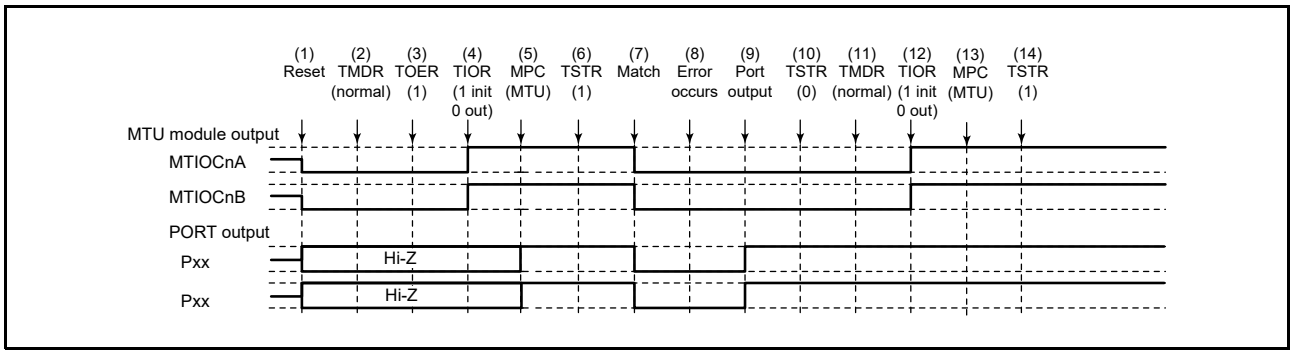


Figure 22.154 Error Occurrence in Normal Mode, Recovery in Normal Mode

(1) After a reset, the MTU output goes low and the ports enter high-impedance state.
(2) After a reset, the TMDR1 setting is for normal mode.
(3) For MTU3 and MTU4 (MTU6 and MTU7), enable output with the TOERA (TOERB) register before initializing the pins with the TIOR register.
(4) Initialize the pins with the TIOR register. (In the example, the initial output is a high level, and a low level is output on compare match occurrence.)
(5) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(6) Start count operation by setting the TSTRA (TSTRB) register.
(7) Output goes low on compare match occurrence.
(8) An error occurs.
(9) Allow non-active level output by setting the pins as general output ports using the port direction registers (PDR), port output data registers (PODR), and port mode registers (PMR) of the I/O ports.
(10) Stop count operation by setting the TSTRA (TSTRB) register.
(11) This step is not necessary when restarting in normal mode.
(12) Initialize the pins with the TIOR register.
(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting the TSTRA (TSTRB) register.

(2) Operation When Error Occurs in Normal Mode and Operation is Restarted in PWM Mode 1

Figure 22.155 shows a case in which an error occurs in normal mode and operation is restarted in PWM mode 1 after re-setting.

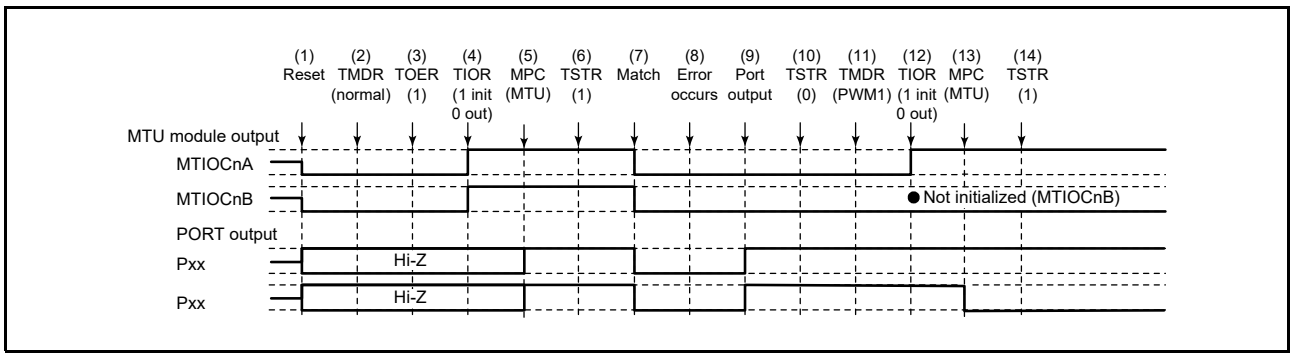


Figure 22.155 Error Occurrence in Normal Mode, Recovery in PWM Mode 1

(1) to (10) are the same as in Figure 22.154.
(11) Set PWM mode 1.
(12) Initialize the pins with the TIOR register. (In PWM mode 1, waveforms are not output to the MTIOCnB (MTIOCnD) pins. To output a specified level, make necessary settings for general output ports in the port direction registers (PDR) and port output data registers (PODR) of the I/O ports.)
(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting the TSTRA (TSTRB) register.

(3) Operation When Error Occurs in Normal Mode and Operation is Restarted in PWM mode 2

Figure 22.156 shows a case in which an error occurs in normal mode and operation is restarted in PWM mode 2 after re-setting.

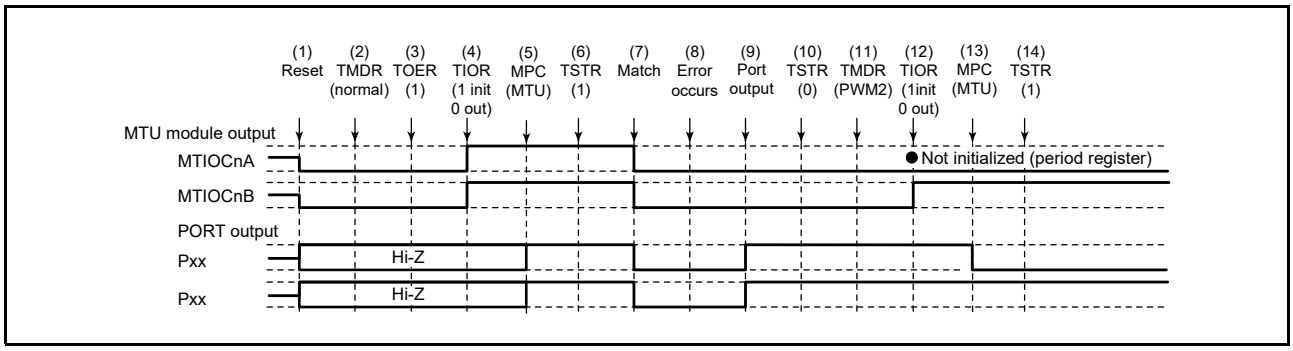


Figure 22.156 Error Occurrence in Normal Mode, Recovery in PWM Mode 2

(1) to (10) are the same as in Figure 22.154.
(11) Set PWM mode 2.
(12) Initialize the pins with the TIOR register. (In PWM mode 2, a waveform is not output to the pin that corresponds to the TGR register used as a period register. To output a specified level, make necessary settings for general output ports in the port direction registers (PDR) and port output data registers (PODR) of the I/O ports.)
(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting the TSTRA register.

Note: PWM mode 2 can only be selected for MTU0 to MTU2 and MTU9, and therefore the TOERA register setting is not necessary.

(4) Operation When Error Occurs in Normal Mode and Operation is Restarted in Phase Counting Mode

Figure 22.157 shows a case in which an error occurs in normal mode and operation is restarted in phase counting mode after re-setting.

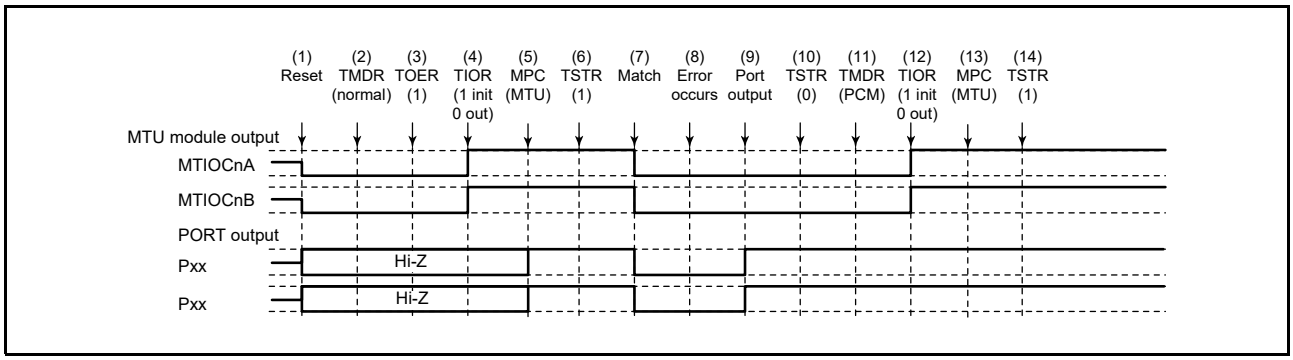


Figure 22.157 Error Occurrence in Normal Mode, Recovery in Phase Counting Mode

(1) to (10) are the same as in Figure 22.154.
(11) Set the phase counting mode.
(12) Initialize the pins with the TIOR register.
(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting the TSTRA register.

Note: The phase counting mode can only be selected for MTU1 and MTU2, and therefore the TOERA register setting is not necessary.

(5) Operation When Error Occurs in Normal Mode and Operation is Restarted in Complementary PWM Mode

Figure 22.158 shows a case in which an error occurs in normal mode and operation is restarted in complementary PWM mode after re-setting.

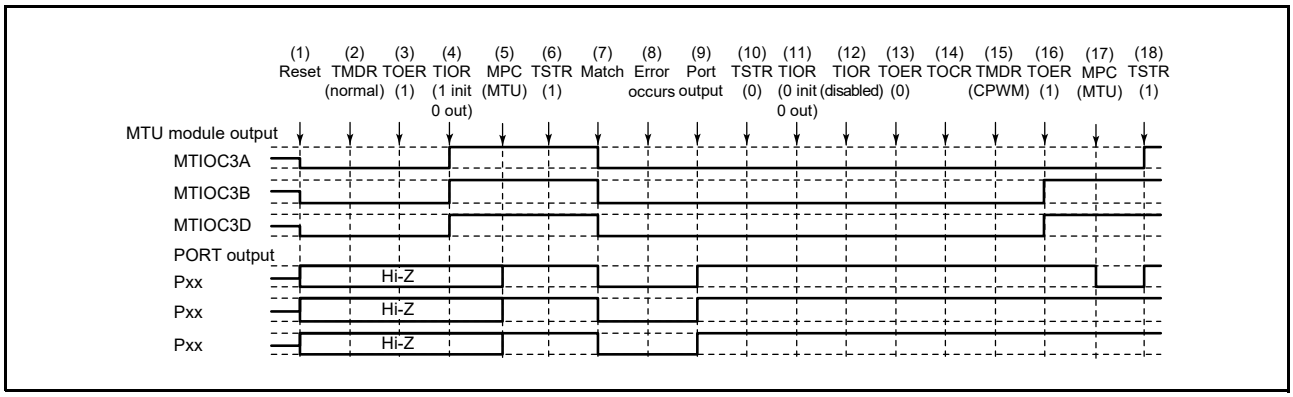


Figure 22.158 Error Occurrence in Normal Mode, Recovery in Complementary PWM Mode

(1) to (10) are the same as in Figure 22.154.
(11) Initialize the normal mode waveform generation block with the TIOR register.
(12) Disable operation of the normal mode waveform generation block with the TIOR register.
(13) Disable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.
(14) Select the complementary PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A (TOCR1B and TOCR2B).
(15) Set complementary PWM mode.
(16) Enable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.
(17) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(18) Restart operation by setting the TSTRA (TSTRB) register.

(6) Operation When Error Occurs in Normal Mode and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 22.159 shows a case in which an error occurs in normal mode and operation is restarted in reset-synchronized PWM mode after re-setting.

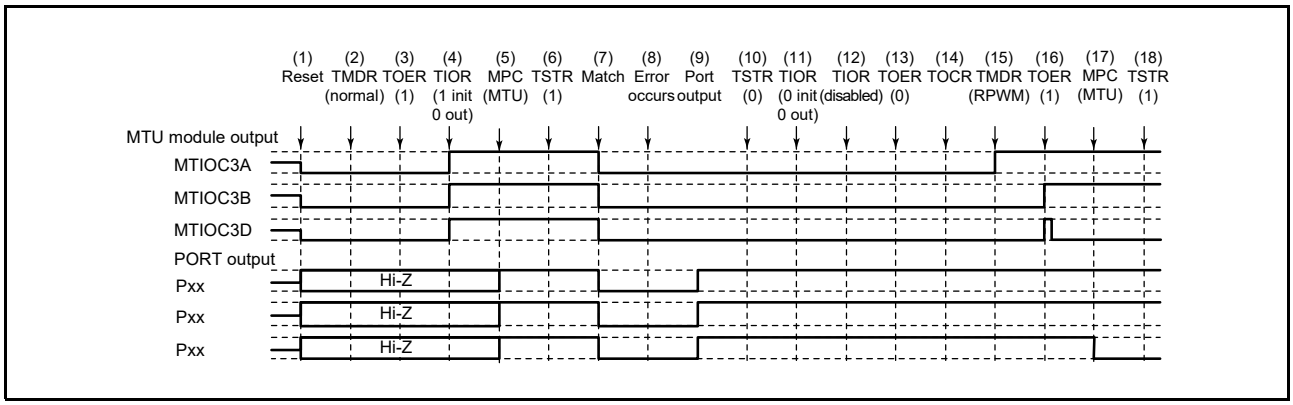


Figure 22.159 Error Occurrence in Normal Mode, Recovery in Reset-Synchronized PWM Mode

(1) to (13) are the same as in Figure 22.158.

(14) Select the reset-synchronized PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A (TOCR1B and TOCR2B).

(15) Set reset-synchronized PWM mode.

(16) Enable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.

(17) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.

(18) Restart operation by setting the TSTRA (TSTRB) register.

(7) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in Normal Mode

Figure 22.160 shows a case in which an error occurs in PWM mode 1 and operation is restarted in normal mode after re-setting.

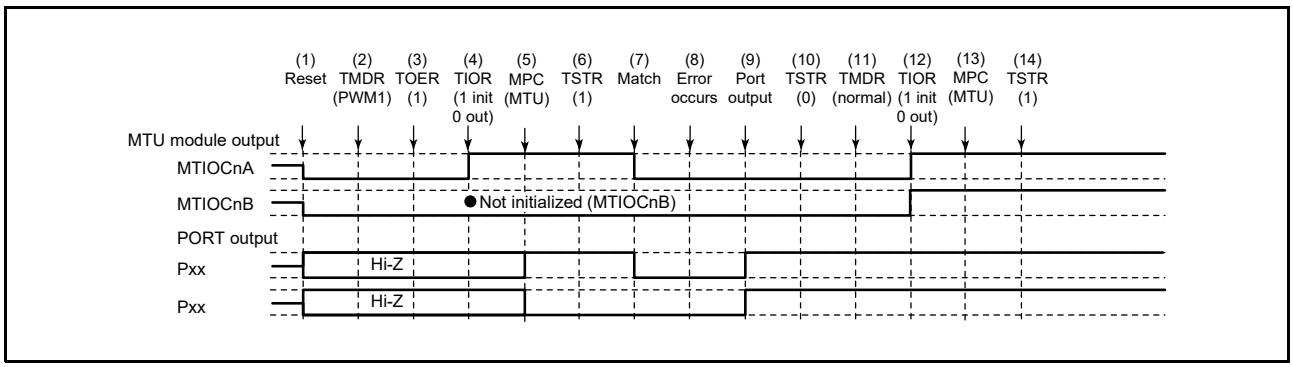


Figure 22.160 Error Occurrence in PWM Mode 1, Recovery in Normal Mode

(1) After a reset, the MTU output goes low and the ports enter high-impedance state.
(2) Set PWM mode 1.
(3) For MTU3 and MTU4 (MTU6 and MTU7), enable output with the TOERA (TOERB) register before initializing the pins with the TIOR register.
(4) Initialize the pins with the TIOR register. (In the example, the initial output is a high level, and a low level is output on compare match occurrence. In PWM mode 1, the MTIOCnB side is not initialized.)
(5) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(6) Start count operation by setting the TSTRA (TSTRB) register.
(7) Output goes low on compare match occurrence.
(8) An error occurs.
(9) Allow non-active level output by setting the pins as general output ports using the port direction registers (PDR), port output data registers (PODR), and port mode registers (PMR) of the I/O ports.
(10) Stop count operation by setting the TSTRA (TSTRB) register.
(11) Set normal mode.
(12) Initialize the pins with the TIOR register.
(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting the TSTRA (TSTRB) register.

(8) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in PWM mode 1

Figure 22.161 shows a case in which an error occurs in PWM mode 1 and operation is restarted in PWM mode 1 after re-setting.

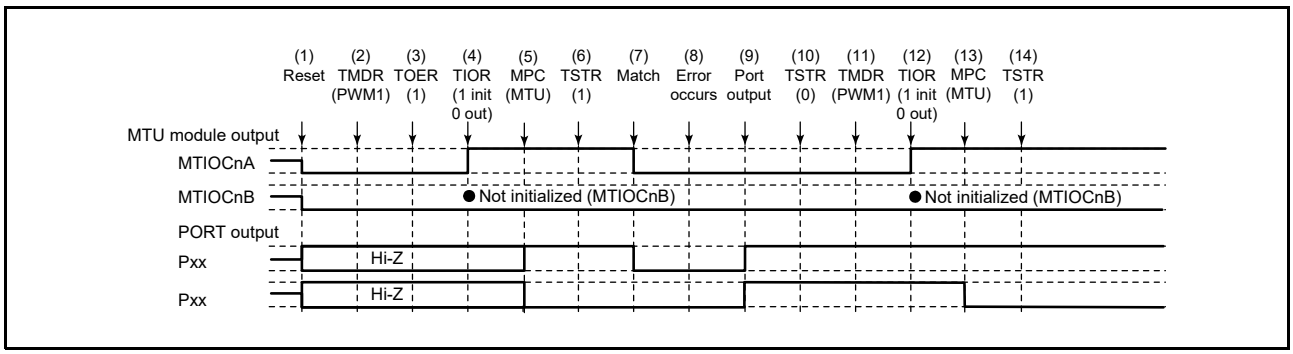


Figure 22.161 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 1

(1) to (10) are the same as in Figure 22.160.
(11) This step is not necessary when restarting in PWM mode 1.
(12) Initialize the pins with the TIOR register. (In PWM mode 1, waveforms are not output to the MTIOCnB (MTIOCnD) pins. To output a specified level, make necessary settings for general output ports in the port direction registers (PDR) and port output data registers (PODR) of the I/O ports.)
(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting the TSTRA (TSTRB) register.

(9) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in PWM mode 2

Figure 22.162 shows a case in which an error occurs in PWM mode 1 and operation is restarted in PWM mode 2 after re-setting.

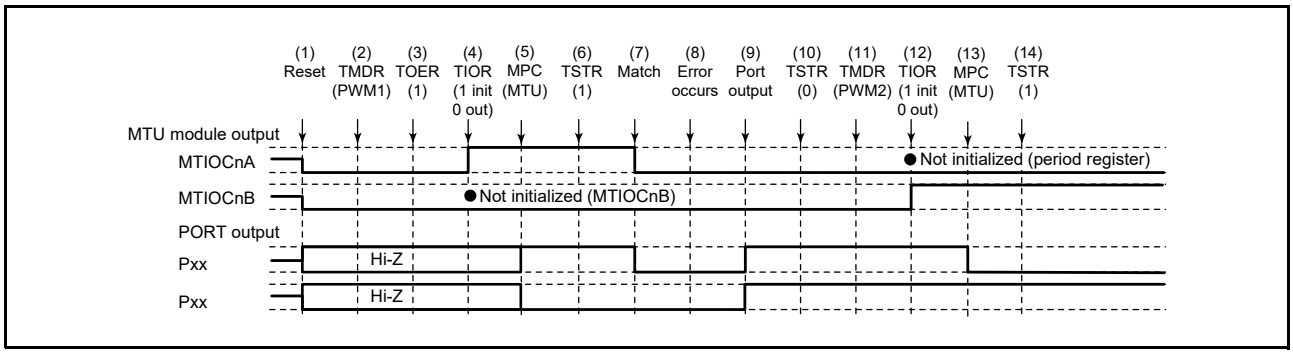


Figure 22.162 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 2

(1) to (10) are the same as in Figure 22.160.
(11) Set PWM mode 2.
(12) Initialize the pins with the TIOR register. (In PWM mode 2, a waveform is not output to the pin that corresponds to the TGR register used as a period register. To output a specified level, make necessary settings for general output ports in the port direction registers (PDR) and port output data registers (PODR) of the I/O ports.)
(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting the TSTRA register.

Note: PWM mode 2 can only be selected for MTU0 to MTU2 and MTU9, and therefore the TOERA register setting is not necessary.

(10) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in Phase Counting Mode

Figure 22.163 shows a case in which an error occurs in PWM mode 1 and operation is restarted in phase counting mode after re-setting.

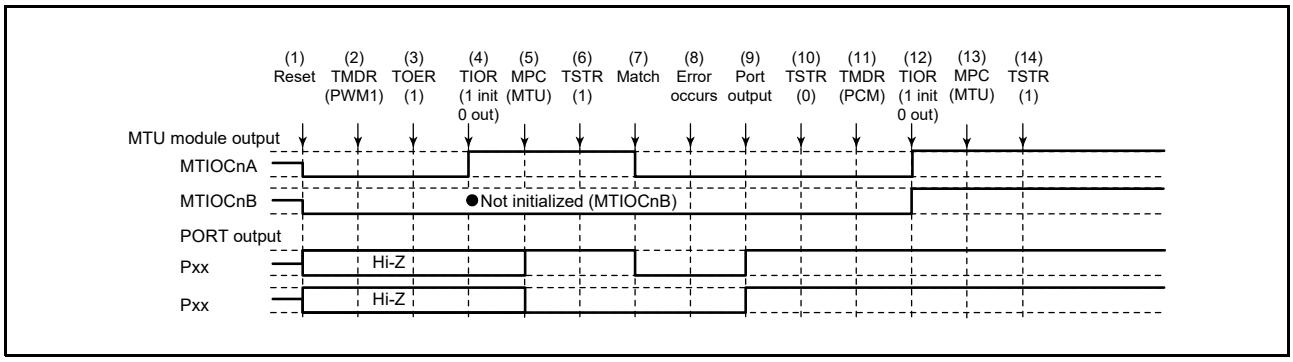


Figure 22.163 Error Occurrence in PWM Mode 1, Recovery in Phase Counting Mode

(1) to (10) are the same as in Figure 22.160.
(11) Set the phase counting mode.
(12) Initialize the pins with the TIOR register.
(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting the TSTRA register.

Note: The phase counting mode can only be selected for MTU1 and MTU2, and therefore the TOERA register setting is not necessary.

(11) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in Complementary PWM Mode

Figure 22.164 shows a case in which an error occurs in PWM mode 1 and operation is restarted in complementary PWM mode after re-setting.

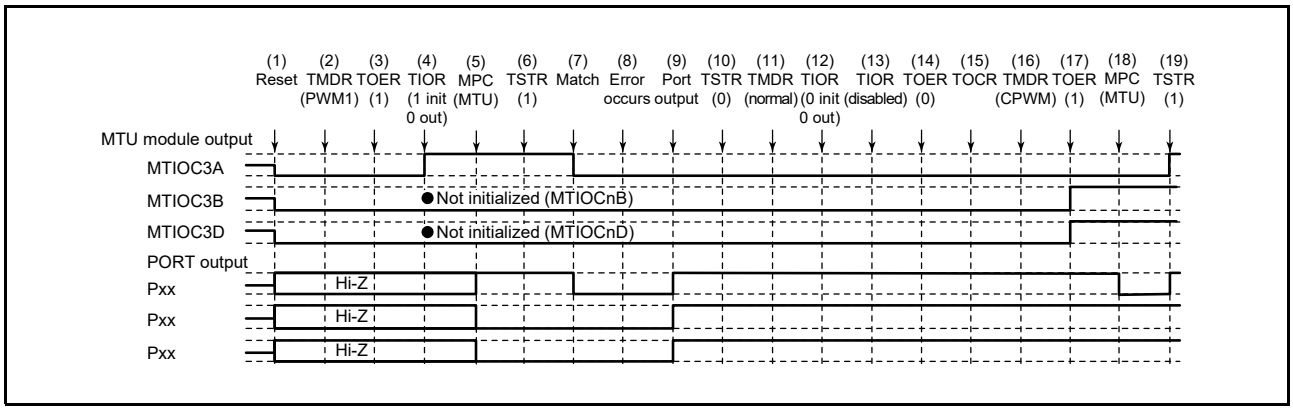


Figure 22.164 Error Occurrence in PWM Mode 1, Recovery in Complementary PWM Mode

(1) to (10) are the same as in Figure 22.160.
(11) Set normal mode to initialize the normal mode waveform generation block.
(12) Initialize the PWM mode 1 waveform generation block with the TIOR register.
(13) Disable operation of the PWM mode 1 waveform generation block with the TIOR register.
(14) Disable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.
(15) Select the complementary PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A (TSTRB).
(16) Set complementary PWM mode.
(17) Enable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.
(18) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(19) Restart operation by setting the TSTRA (TSTRB) register.

(12) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 22.165 shows a case in which an error occurs in PWM mode 1 and operation is restarted in reset-synchronized PWM mode after re-setting.

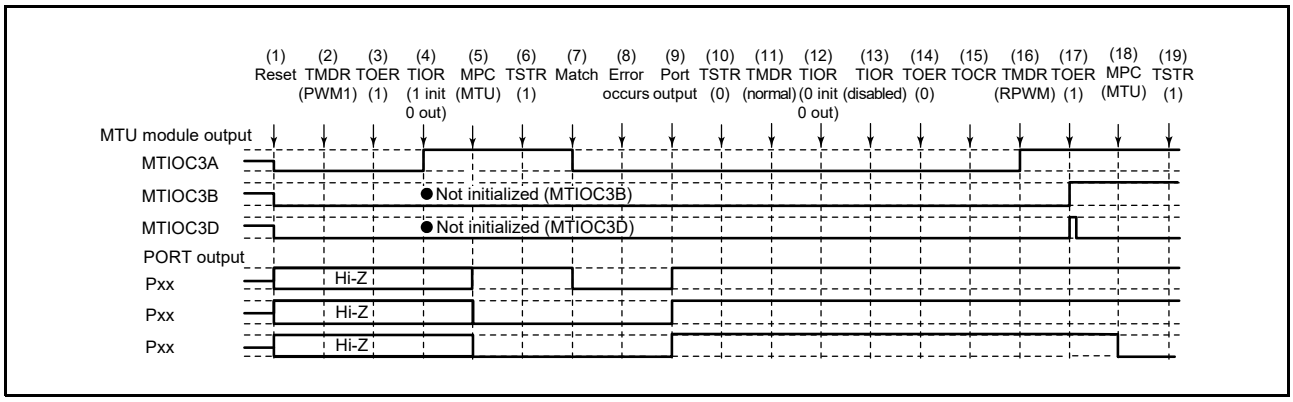


Figure 22.165 Error Occurrence in PWM Mode 1, Recovery in Reset-Synchronized PWM Mode

(1) to (14) are the same as in Figure 22.164.

(15) Select the reset-synchronized PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A (TOCR1B and TOCR2B).

(16) Set reset-synchronized PWM mode.

(17) Enable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.

(18) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.

(19) Restart operation by setting the TSTRA (TSTRB) register.

(13) Operation When Error Occurs in PWM Mode 2 and Operation is Restarted in Normal Mode

Figure 22.166 shows a case in which an error occurs in PWM mode 2 and operation is restarted in normal mode after re-setting.

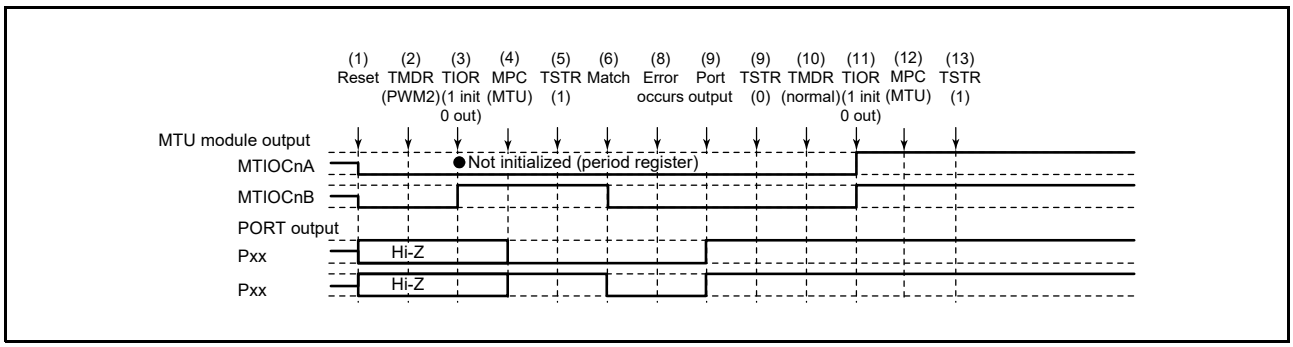


Figure 22.166 Error Occurrence in PWM Mode 2, Recovery in Normal Mode

(1) After a reset, the MTU output goes low and the ports enter high-impedance state.
(2) Set PWM mode 2.
(3) Initialize the pins with the TIOR register. (In the example, the initial output is a high level, and a low level is output on compare match occurrence. In PWM mode 2, the pin that corresponds to the TGR register used as a period register is not initialized. In the example, the MTUn.TGRA register is used as a period register.)
(4) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(5) Start count operation by setting the TSTRA register.
(6) Output goes low on compare match occurrence.
(7) An error occurs.
(8) Allow non-active level output by setting the pins as general output ports using the port direction registers (PDR), port output data registers (PODR), and port mode registers (PMR) of the I/O ports.
(9) Stop count operation by setting the TSTRA register.
(10) Set normal mode.
(11) Initialize the pins with the TIOR register.
(12) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(13) Restart operation by setting the TSTRA register.

(14) Operation When Error Occurs in PWM Mode 2 and Operation is Restarted in PWM Mode 1

Figure 22.167 shows a case in which an error occurs in PWM mode 2 and operation is restarted in PWM mode 1 after re-setting.

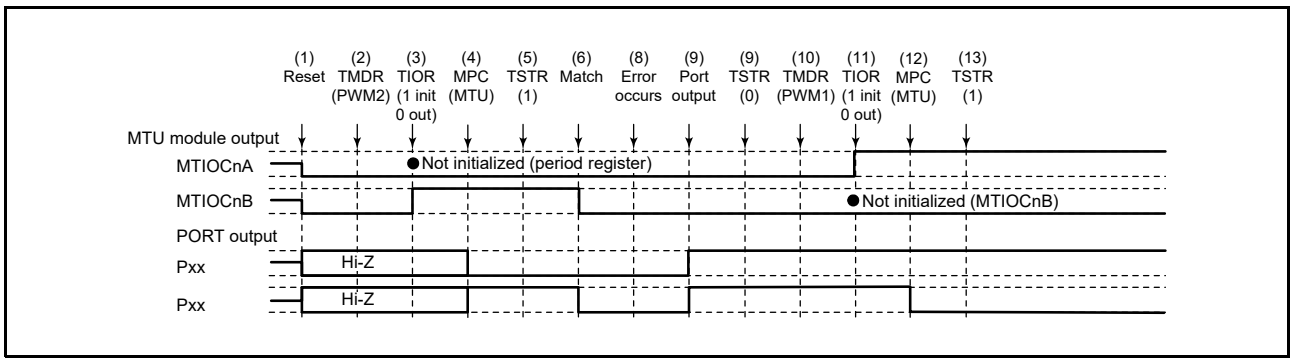


Figure 22.167 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 1

(1) to (9) are the same as in Figure 22.166.
(10) Set PWM mode 1.
(11) Initialize the pins with the TIOR register. (In PWM mode 1, waveforms are not output to the MTIOCnB (MTIOCnD) pins. To output a specified level, make necessary settings for general output ports in the port direction registers (PDR) and port output data registers (PODR) of the I/O ports.)
(12) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(13) Restart operation by setting the TSTRA register.

(15) Operation When Error Occurs in PWM Mode 2 and Operation is Restarted in PWM Mode 2

Figure 22.168 shows a case in which an error occurs in PWM mode 2 and operation is restarted in PWM mode 2 after re-setting.

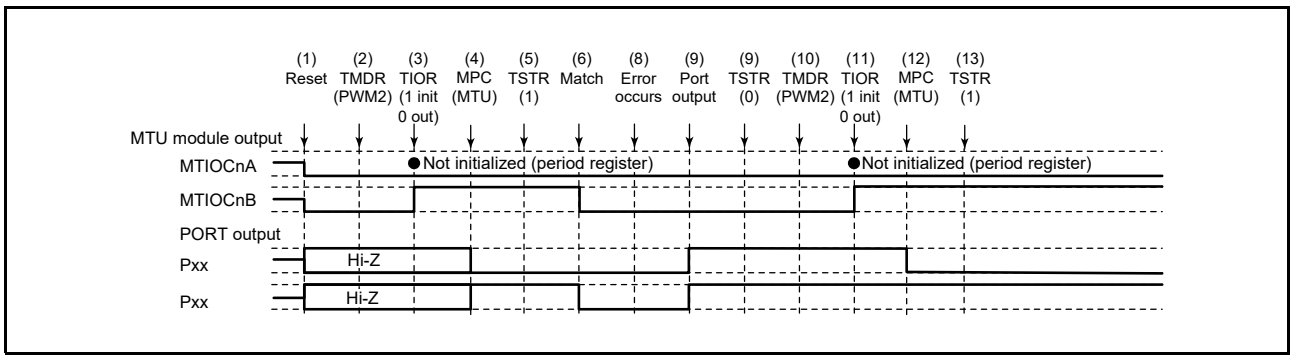


Figure 22.168 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 2

(1) to (9) are the same as in Figure 22.166.
(10) This step is not necessary when restarting in PWM mode 2.
(11) Initialize the pins with the TIOR register. (In PWM mode 2, a waveform is not output to the pin that corresponds to the TGR register used as a period register. To output a specified level, make necessary settings for general output ports in the port direction registers (PDR) and port output data registers (PODR) of the I/O ports.)
(12) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(13) Restart operation by setting the TSTRA register.

(16) Operation When Error Occurs in PWM Mode 2 and Operation is Restarted in Phase Counting Mode

Figure 22.169 shows a case in which an error occurs in PWM mode 2 and operation is restarted in phase counting mode after re-setting.

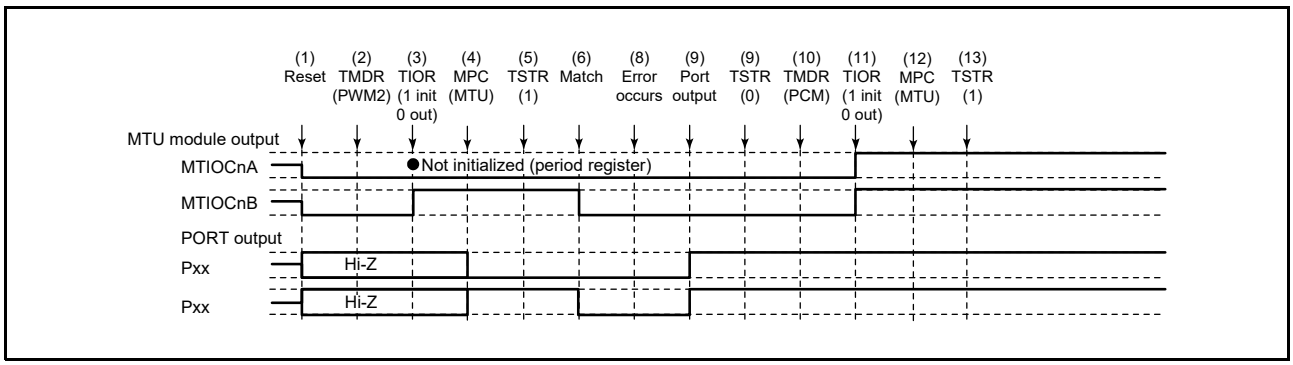


Figure 22.169 Error Occurrence in PWM Mode 2, Recovery in Phase Counting Mode

(1) to (9) are the same as in Figure 22.166.
(10) Set the phase counting mode.
(11) Initialize the pins with the TIOR register.
(12) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(13) Restart operation by setting the TSTRA register.

(17) Operation When Error Occurs in Phase Counting Mode and Operation is Restarted in Normal Mode

Figure 22.170 shows a case in which an error occurs in phase counting mode and operation is restarted in normal mode after re-setting.

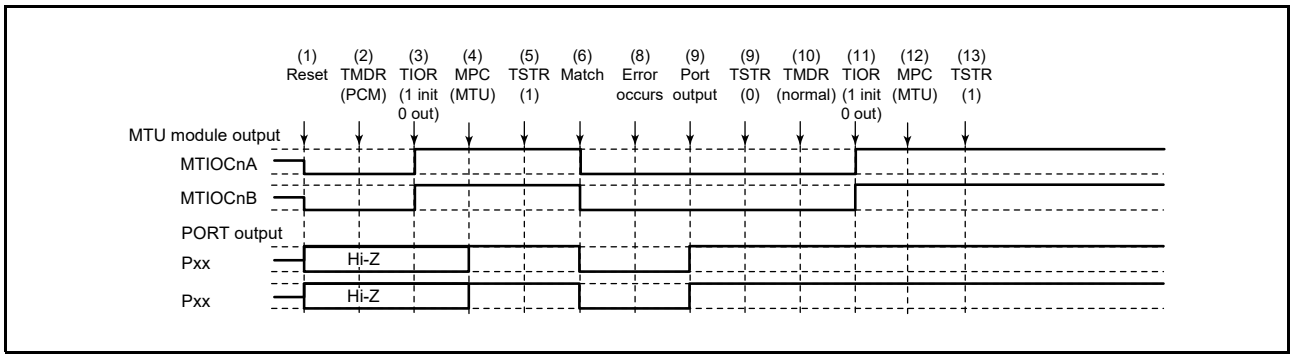


Figure 22.170 Error Occurrence in Phase Counting Mode, Recovery in Normal Mode

(1) After a reset, the MTU output goes low and the ports enter high-impedance state.
(2) Set phase counting mode.
(3) Initialize the pins with the TIOR register. (In the example, the initial output is a high level, and a low level is output on compare match occurrence.)
(4) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(5) Start count operation by setting the TSTRA register.
(6) Output goes low on compare match occurrence.
(7) An error occurs.
(8) Allow non-active level output by setting the pins as general output ports using the port direction registers (PDR), port output data registers (PODR), and port mode registers (PMR) of the I/O ports.
(9) Stop count operation by setting the TSTRA register.
(10) Set normal mode.
(11) Initialize the pins with the TIOR register.
(12) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(13) Restart operation by setting the TSTRA register.

(18) Operation When Error Occurs in Phase Counting Mode and Operation is Restarted in PWM Mode 1

Figure 22.171 shows a case in which an error occurs in phase counting mode and operation is restarted in PWM mode 1 after re-setting.

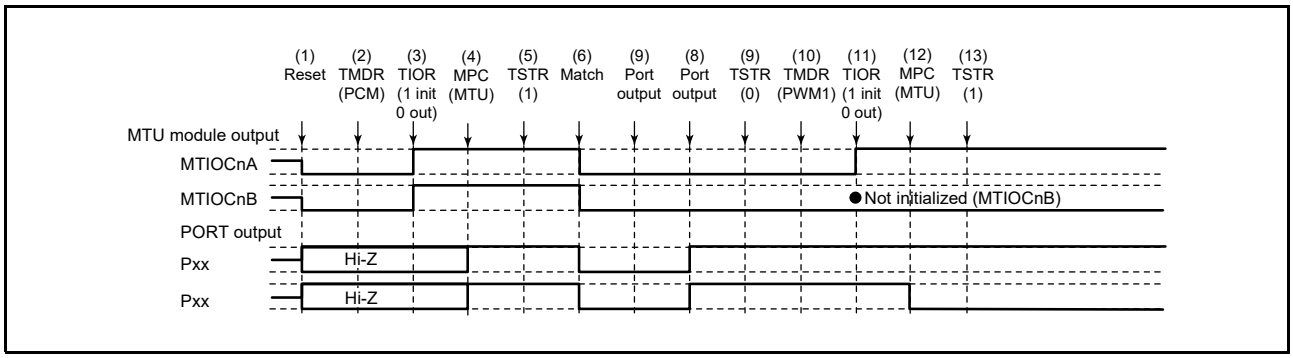


Figure 22.171 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 1

(1) to (9) are the same as in Figure 22.170.
(10) Set PWM mode 1.
(11) Initialize the pins with the TIOR register. (In PWM mode 1, waveforms are not output to the MTIOCnB (MTIOCnD) pins. To output a specified level, make necessary settings for general output ports in the port direction registers (PDR) and port output data registers (PODR) of the I/O ports.)
(12) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(13) Restart operation by setting the TSTRA register.

(19) Operation When Error Occurs in Phase Counting Mode and Operation is Restarted in PWM Mode 2

Figure 22.172 shows a case in which an error occurs in phase counting mode and operation is restarted in PWM mode 2 after re-setting.

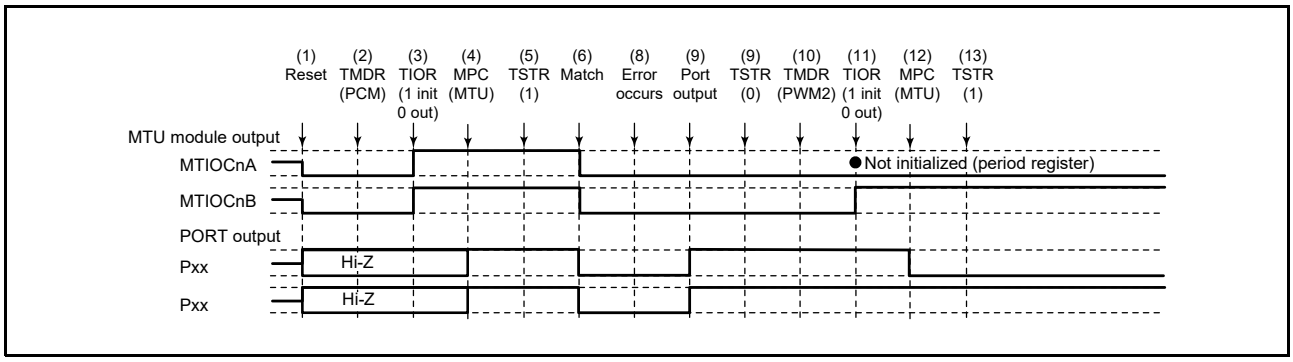


Figure 22.172 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 2

(1) to (9) are the same as in Figure 22.170.
(10) Set PWM mode 2.
(11) Initialize the pins with the TIOR register. (In PWM mode 2, a waveform is not output to the pin that corresponds to the TGR register used as a period register. To output a specified level, make necessary settings for general output ports in the port direction registers (PDR) and port output data registers (PODR) of the I/O ports.)
(12) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(13) Restart operation by setting the TSTRA register.

(20) Operation When Error Occurs in Phase Counting Mode and Operation is Restarted in Phase Counting Mode

Figure 22.173 shows a case in which an error occurs in phase counting mode and operation is restarted in phase counting mode after re-setting.

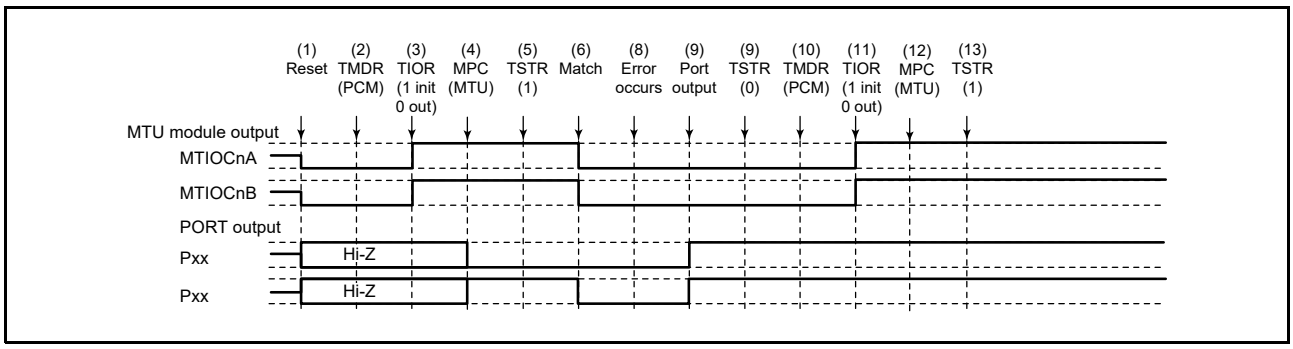


Figure 22.173 Error Occurrence in Phase Counting Mode, Recovery in Phase Counting Mode

(1) to (9) are the same as in Figure 22.170.
(10) This step is not necessary when restarting in phase counting mode.
(11) Initialize the pins with the TIOR register.
(12) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(13) Restart operation by setting the TSTRA register.

(21) Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in Normal Mode

Figure 22.174 shows a case in which an error occurs in complementary PWM mode and operation is restarted in normal mode after re-setting.

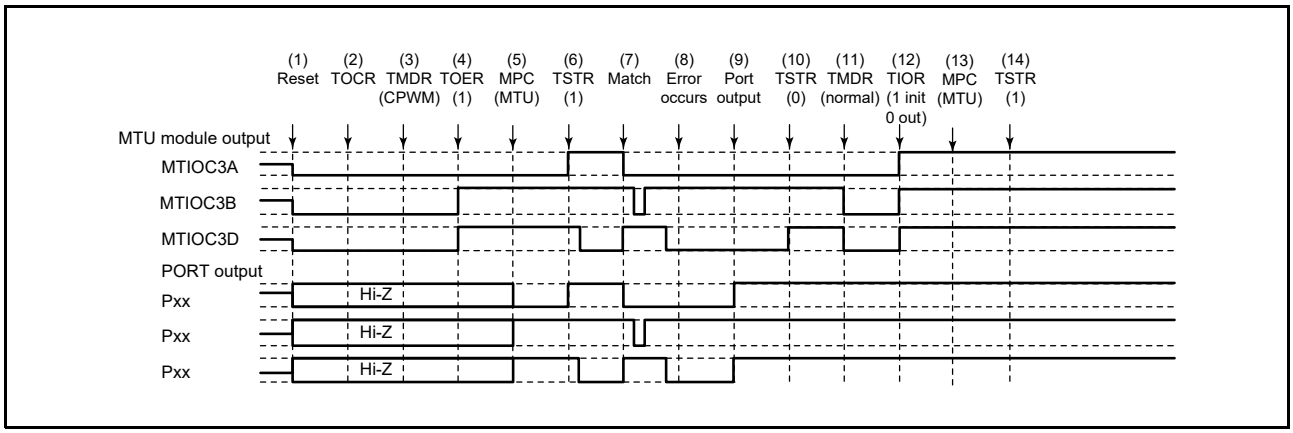


Figure 22.174 Error Occurrence in Complementary PWM Mode, Recovery in Normal Mode

(1) After a reset, the MTU3 output goes low and the ports enter high-impedance state.
(2) Select the complementary PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A (TOCR1B and TOCR2B).
(3) Set complementary PWM mode.
(4) Enable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.
(5) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(6) Start count operation by setting the TSTRA (TSTRB) register.
(7) The complementary PWM waveform is output on compare match occurrence.
(8) An error occurs.
(9) Allow non-active level output by setting the pins as general output ports using the port direction registers (PDR), port output data registers (PODR), and port mode registers (PMR) of the I/O ports.
(10) Stop count operation by setting the TSTRA (TSTRB) register. (MTU output becomes the initial complementary PWM output value).
(11) Set normal mode (MTU output goes low).
(12) Initialize the pins with the TIOR register.
(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting the TSTRA (TSTRB) register.

(22) Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in PWM Mode 1

Figure 22.175 shows a case in which an error occurs in complementary PWM mode and operation is restarted in PWM mode 1 after re-setting.

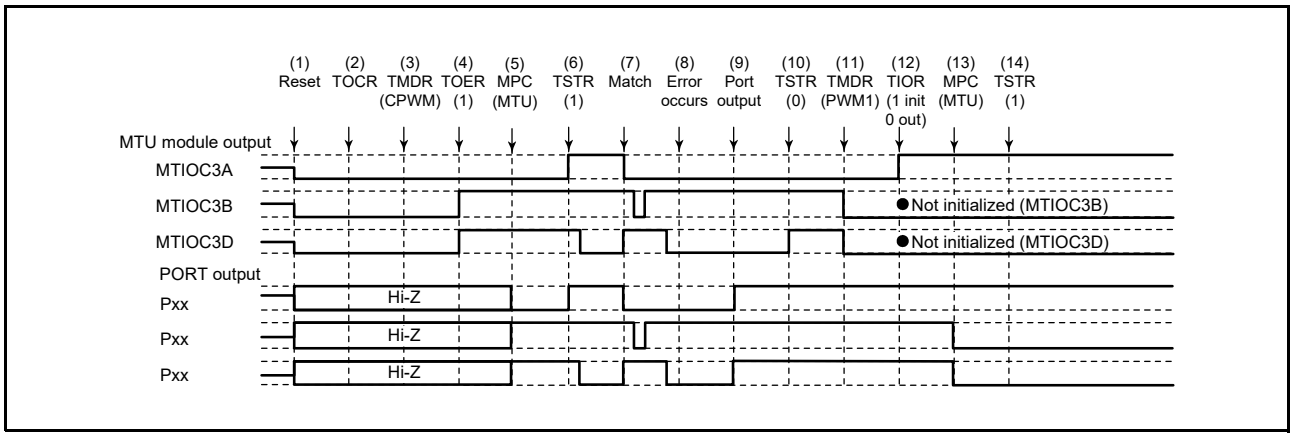


Figure 22.175 Error Occurrence in Complementary PWM Mode, Recovery in PWM Mode 1

(1) to (10) are the same as in Figure 22.174.
(11) Set PWM mode 1 (MTU output goes low).
(12) Initialize the pins with the TIOR register. (In PWM mode 1, waveforms are not output to the MTIOCnB (MTIOCnD) pins. To output a specified level, make necessary settings for general output ports in the port direction registers (PDR) and port output data registers (PODR) of the I/O ports.)
(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting the TSTRA (TSTRB) register.

(23) Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in Complementary PWM Mode

Figure 22.176 shows a case in which an error occurs in complementary PWM mode and operation is restarted in complementary PWM mode after re-setting (when operation is restarted using the period and duty settings at the time of stopping the counter).

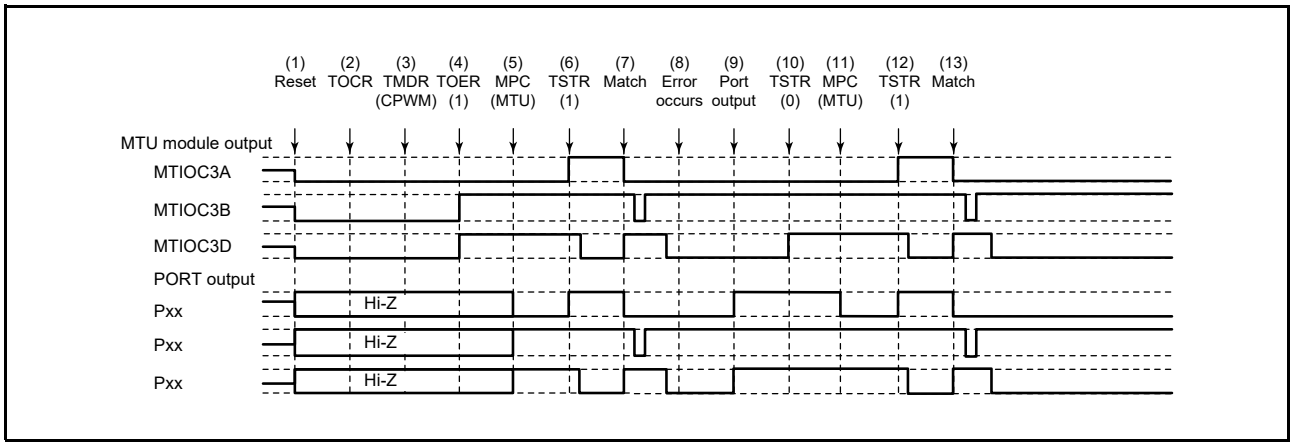


Figure 22.176 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode (1)

(1) to (10) are the same as in Figure 22.174.
(11) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(12) Restart operation by setting the TSTRA (TSTRB) register.
(13) The complementary PWM waveform is output on compare match occurrence.

(24) Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in Complementary PWM Mode with New Settings

Figure 22.177 shows a case in which an error occurs in complementary PWM mode and operation is restarted in complementary PWM mode after re-setting (operation is restarted using new period and duty ratio settings).

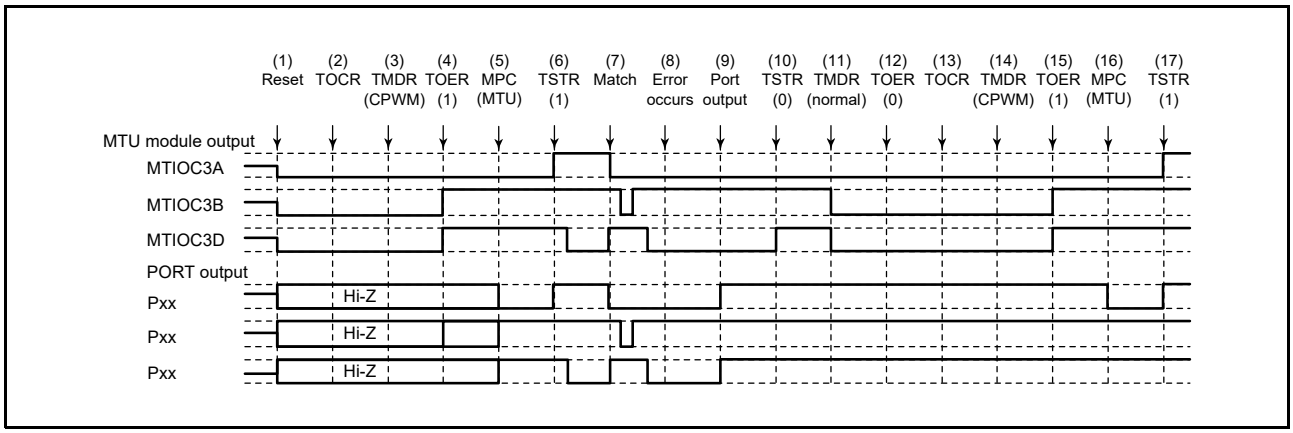


Figure 22.177 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode (2)

(1) to (10) are the same as in Figure 22.174.
(11) Set normal mode and make new settings (MTU output goes low).
(12) Disable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.
(13) Select the complementary PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A (TOCR1B and TOCR2B).
(14) Set complementary PWM mode.
(15) Enable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.
(16) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(17) Restart operation by setting the TSTRA (TSTRB) register.

(25) Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 22.178 shows a case in which an error occurs in complementary PWM mode and operation is restarted in reset-synchronized PWM mode after re-setting.

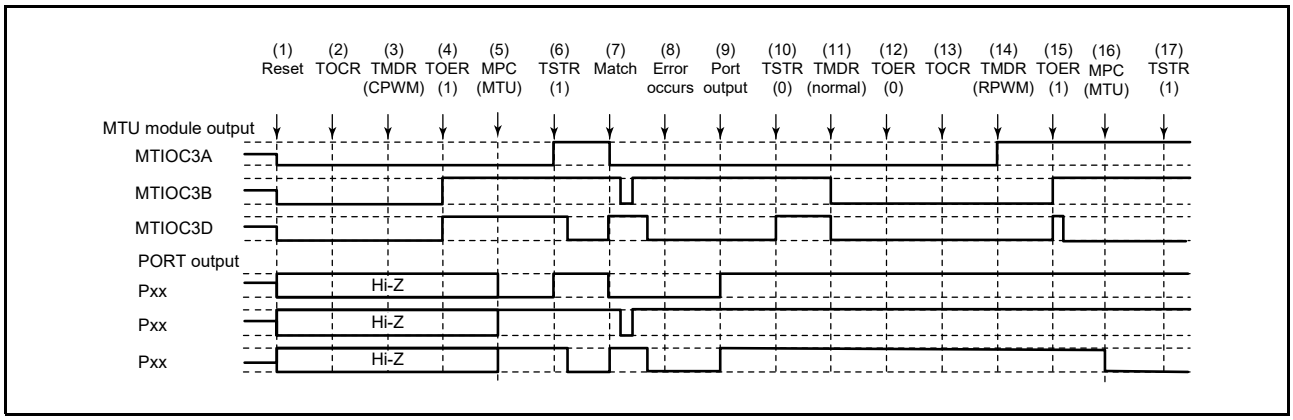


Figure 22.178 Error Occurrence in Complementary PWM Mode, Recovery in Reset-Synchronized PWM Mode

(1) to (10) are the same as in Figure 22.174.
(11) Set normal mode (MTU output goes low).
(12) Disable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.
(13) Select the reset-synchronized PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A (TOCR1B and TOCR2B).
(14) Set reset-synchronized PWM mode.
(15) Enable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.
(16) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(17) Restart operation by setting the TSTRA (TSTRB) register.

(26) Operation When Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in Normal Mode

Figure 22.179 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in normal mode after re-setting.

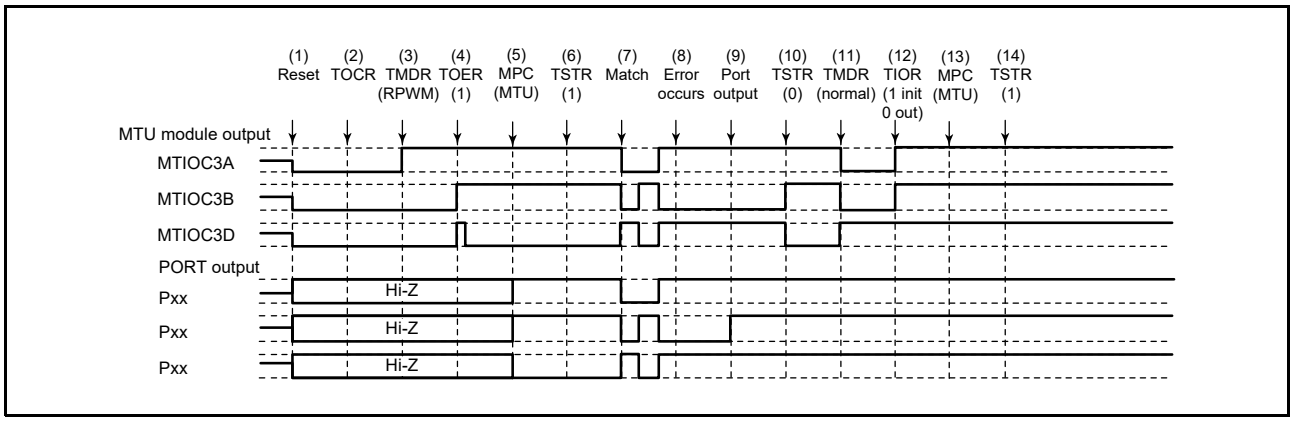


Figure 22.179 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Normal Mode

(1) After a reset, the MTU output goes low and the ports enter high-impedance state.
(2) Select the reset-synchronized PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A (TOCR1B and TOCR2B).
(3) Set reset-synchronized PWM mode.
(4) Enable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.
(5) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(6) Start count operation by setting the TSTRA (TSTRB) register.
(7) The reset-synchronized PWM waveform is output on compare match occurrence.
(8) An error occurs.
(9) Allow non-active level output by setting the pins as general output ports using the port direction registers (PDR), port output data registers (PODR), and port mode registers (PMR) of the I/O ports.
(10) Stop count operation by setting the TSTRA (TSTRB) register. (MTU output becomes the initial reset-synchronized PWM output value.)
(11) Set normal mode (positive-phase MTU output goes low, and negative-phase output goes high).
(12) Initialize the pins with the TIOR register.
(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting the TSTRA (TSTRB) register.

(27) Operation When Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in PWM Mode 1

Figure 22.180 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in PWM mode 1 after re-setting.

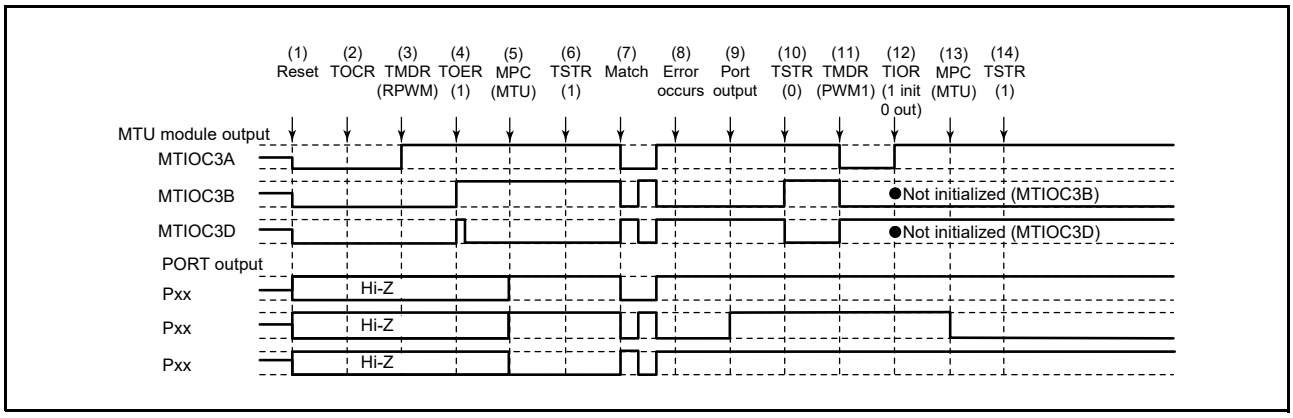


Figure 22.180 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in PWM Mode 1

(1) to (10) are the same as in Figure 22.179.

(11) Set PWM mode 1 (positive-phase MTU output goes low, and negative-phase output goes high).

(12) Initialize the pins with the TIOR register. (In PWM mode 1, waveforms are not output to the MTIOCnB (MTIOCnD) pins. To output a specified level, make necessary settings for general output ports in the port direction registers (PDR) and port output data registers (PODR) of the I/O ports.)

(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.

(14) Restart operation by setting the TSTRA (TSTRB) register.

(28) Operation When Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in Complementary PWM Mode

Figure 22.181 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in complementary PWM mode after re-setting.

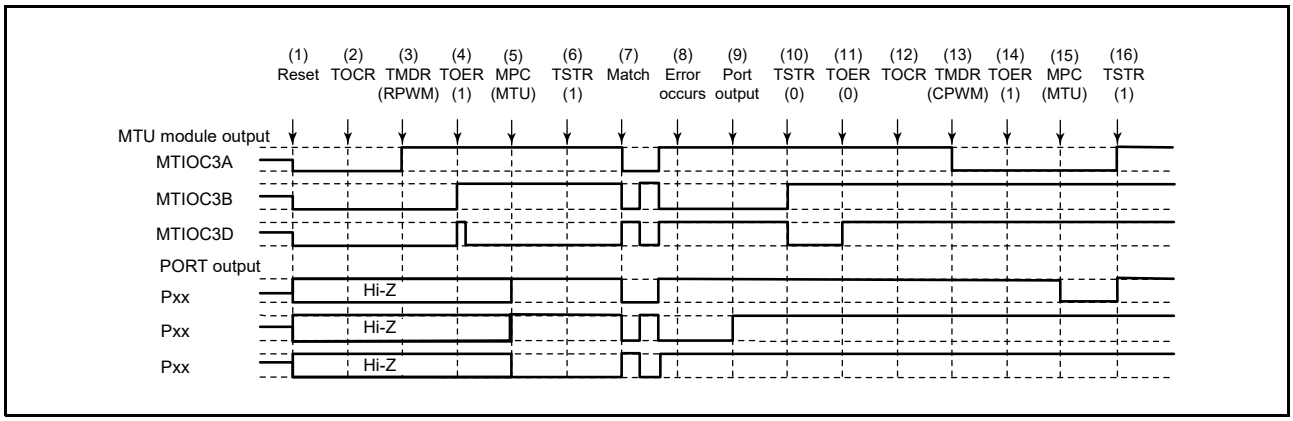


Figure 22.181 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Complementary PWM Mode

(1) to (10) are the same as in Figure 22.179.
(11) Disable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.
(12) Select the complementary PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A (TOCR1B and TOCR2B).
(13) Set complementary PWM mode (MTU cyclic output pin goes low).
(14) Enable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.
(15) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(16) Restart operation by setting the TSTRA (TSTRB) register.

(29) Operation When Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 22.182 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in reset-synchronized PWM mode after re-setting.

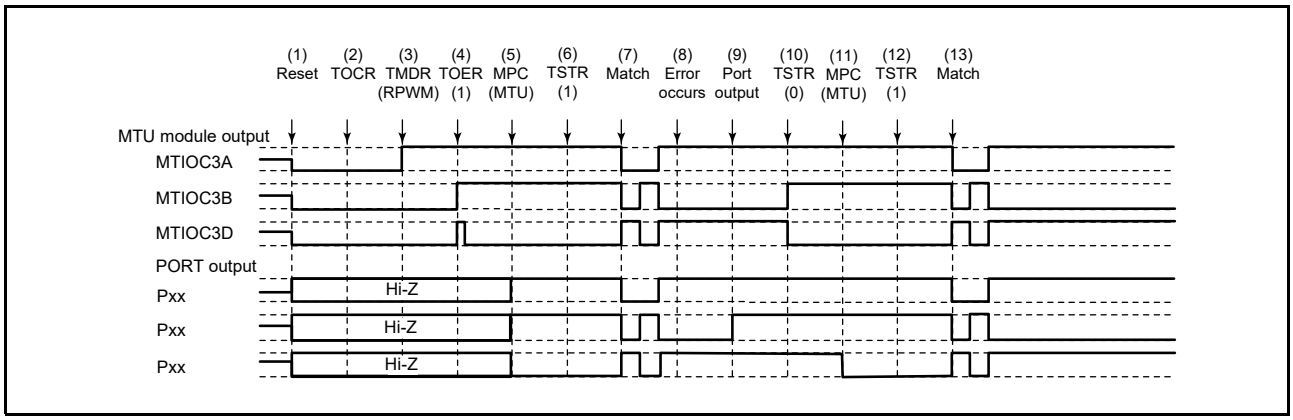


Figure 22.182 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Reset-Synchronized PWM Mode

(1) to (10) are the same as in Figure 22.179.
(11) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(12) Restart operation by setting the TSTRA (TSTRB) register.
(13) The reset-synchronized PWM waveform is output on compare match occurrence.

22.8 Operations Linked by the ELC

22.8.1 Event Signal Output to the ELC

The MTU is capable of operation linked with another module set in advance when its interrupt request signal is used as an event signal by the event link controller (ELC).
The MTU outputs the event signal regardless of the setting of the corresponding interrupt request enable bit.

22.8.2 MTU Operations in Response to Receiving Event Signals from the ELC

The MTU can perform the following operations in response to the event set in advance in the ELSRn register of the event link controller (ELC).

(1) Start Counting Operation

Select “counting is started” as the operation of the MTU by setting the ELOPA, ELOPB, or ELOPE register in the ELC. The ELOPA register controls the operation of MTU0 and MTU3, the ELOPB register controls the operation of MTU4, and the ELOPE register controls the operation of MTU6, MTU7, and MTU9. When the event specified in the ELSRn register occurs, the CSTn bit in the TSTRA/TSTRB register shown in Table 22.81 is set to 1, and the MTU counter starts.
However, when the specified event is generated while the CSTn bit in the TSTRA/TSTRB register has already been set to 1, the event has no effect. Table 22.81 lists the TSTRA/TSTRB register bits used for each channel.

Table 22.81 Counter Start Bit Set by the ELC

Channel No.	Counter Start Bit
MTU0	TSTRA.CST0 bit
MTU3	TSTRA.CST3 bit
MTU4	TSTRA.CST4 bit
MTU6	TSTRB.CST6 bit
MTU7	TSTRB.CST7 bit
MTU9	TSTRA.CST9 bit

(2) Input Capture Operation

Select “input capture” as the operation of the MTU by setting the ELOPA, ELOPB, or ELOPE register in the ELC. The ELOPA register controls the operation of MTU0 and MTU3, the ELOPB register controls the operation of MTU4, and the ELOPE register controls the operation of MTU6, MTU7, and MTU9. When the event specified in the ELSRn register occurs, the value of the TCNT register is captured in the TGR register. When using input capture in response to an event, the corresponding bit of the TIOR register in the MTU should be set for input capture and the CSTn bit of TSTRA/TSTRB register should be set to 1 to start counting by the counter.
In this case, the TIOCnA pin (input capture pin) input has no effect.
Table 22.82 lists the timer general register and I/O control bit used for each channel in input capture operations in response to the ELC.

Table 22.82 Timer General Register and I/O Control Bit Used in the Input Capture Operation

Channel No.	Timer General Register	I/O Control Bit
MTU0	MTU0.TGRA	MTU0.TIORH.IOA[3:0] bits
MTU3	MTU3.TGRA	MTU3.TIORH.IOA[3:0] bits
MTU4	MTU4.TGRA	MTU4.TIORH.IOA[3:0] bits
MTU6	MTU6.TGRA	MTU6.TIORH.IOA[3:0] bits
MTU7	MTU7.TGRA	MTU7.TIORH.IOA[3:0] bits
MTU9	MTU9.TGRA	MTU9.TIORH.IOA[3:0] bits

(3) Restart Counting (Clear Counter) Operation

Select “counting is restarted” as the operation of the MTU by setting the ELOPA, ELOPB, or ELOPE register in the ELC. The ELOPA register controls the operation of MTU0 and MTU3, the ELOPB register controls the operation of MTU4, and the ELOPE register controls the operation of MTU6, MTU7, and MTU9. When the event specified in the ELSRn register occurs, the TCNT register is cleared. If the corresponding CSTn bit in the TSTRA/TSTRB register is set to 1, counting continues. For the CSTn bits in the TSTRA/TSTRB register, refer to Table 22.81.

22.8.3 Usage Notes on MTU Operation by Event Signal Reception from the ELC

The following notes on usage apply when the MTU is used in event link operation.

(1) Start Counting

If the event specified in the ELSRn register occurs during a cycle of writing to a CSTn bit in the TSTRA/TSTRB register, writing to the CSTn bit in the TSTRA/TSTRB register does not proceed because setting of the bit to 1 due to the event takes priority.

(2) Restart Counting (Clear Counter)

If the event specified in the ELSRn register occurs during a cycle of writing to the TCNT counter, writing to the TCNT counter does not proceed because clearing of the counter due to the event takes priority.
In addition, for MTU3 and MTU4 or MTU6 and MTU7 in complementary PWM mode, do not use the counter restarting by the ELC.

23. Port Output Enable 3 (POE3D)

This MCU incorporates a port output enable 3 (POE3D) which can be used to, under various conditions, disable output signals for the MTU and the GPTW. The output status can be selected from high-impedance or general I/O port while the output is disabled.
In this section, “PCLK” is used to refer to PCLKB.

23.1 Overview

Table 23.1 lists the specifications of the POE. Figure 23.1 and Figure 23.2 show a system block diagram around POE and a block diagram of POE, respectively.

Table 23.1 POE Specifications (1/2)

Item	Description
Pin status while output is disabled	• High-impedance • General I/O port
Target pins for switching to disabling of signal output	• MTU output pins MTU0 pins (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D) MTU3 pins (MTIOC3B, MTIOC3D) MTU4 pins (MTIOC4A, MTIOC4B, MTIOC4C, MTIOC4D) MTU6 pins (MTIOC6B, MTIOC6D) MTU7 pins (MTIOC7A, MTIOC7B, MTIOC7C, MTIOC7D) MTU9 pins (MTIOC9A, MTIOC9B, MTIOC9C, MTIOC9D) • GPTW output pins GPTW0 pins (GTIOC0A, GTIOC0B) GPTW1 pins (GTIOC1A, GTIOC1B) GPTW2 pins (GTIOC2A, GTIOC2B) GPTW3 pins (GTIOC3A, GTIOC3B) GPTW4 pins (GTIOC4A, GTIOC4B) GPTW5 pins (GTIOC5A, GTIOC5B) GPTW6 pins (GTIOC6A, GTIOC6B) GPTW7 pins (GTIOC7A, GTIOC7B)
Generating conditions of request for switching to disable output	• Input signal detection: Detection of the POE0#, POE4#, POE8#, POE10#, POE11#, POE12#, and POE9# signal level. • Simultaneous conduction between output signals: When the output signal levels (active levels) of the following combinations match for one cycle or more (simultaneous conduction).

	MTU Complementary PWM Output Signals
1	MTIOC3B and MTIOC3D
2	MTIOC4A and MTIOC4C
3	MTIOC4B and MTIOC4D
4	MTIOC6B and MTIOC6D
5	MTIOC7A and MTIOC7C
6	MTIOC7B and MTIOC7D

	GPTW Complementary PWM Output Signals
1	GTIOC0A and GTIOC0B
2	GTIOC1A and GTIOC1B
3	GTIOC2A and GTIOC2B
4	GTIOC4A and GTIOC4B
5	GTIOC5A and GTIOC5B
6	GTIOC6A and GTIOC6B
7	GTIOC7A and GTIOC7B

• SPOER register setting being made
• Detection that the main clock oscillator had stopped oscillating
• Detection of the comparator C (CMPC) outputs

Table 23.1 POE Specifications (2/2)

Item	Description
Function	• Each of the POE0#, POE4#, POE8#, POE10#, POE11#, POE12#, and POE9# pins can be set for falling-edge or low-level detection. When setting a low-level detection, a sampling clock can be selected from PCLK/1, PCLK/2, PCLK/4, PCLK/8, PCLK/16, and PCLK/128, while the number of samples can be selected from four, eight, or 16. • The outputs of the target pins can be disabled by detecting falling-edge or low-level of input to the POE0#, POE4#, POE8#, POE10#, POE11#, POE12#, and POE9# pins. • The outputs of the target pins can be disabled when oscillation stop is detected by the oscillation stop detection function of the clock generator. • The MTU complementary PWM outputs can be disabled when output levels of the MTU complementary PWM output signals are compared and simultaneous active-level output continues for one cycle or more. • The GPTW outputs can be disabled when output levels of the GPTW complementary PWM output signals (GPTW0 to GPTW2, GPTW4 to GPTW6, and GPTW7) are compared and simultaneous active-level output continues for one cycle or more. • The outputs of the target pins can be disabled in response to comparator C (CMPC) output detection. • The outputs of the target pins can be disabled by modifying the settings of the POE registers. • Interrupts can be generated by input-level sampling or output-level comparison results. • Output disabling requests triggered by the POE0#, POE4#, POE8#, POE10#, POE11#, POE12#, and POE9# pins and the COMP0 to COMP5 level detection signals can be masked by the MTU (MTU0 to MTU4, MTU9, MTU7, and MTU9) and GPTW (GPTW0 to GPTW7) output signals.

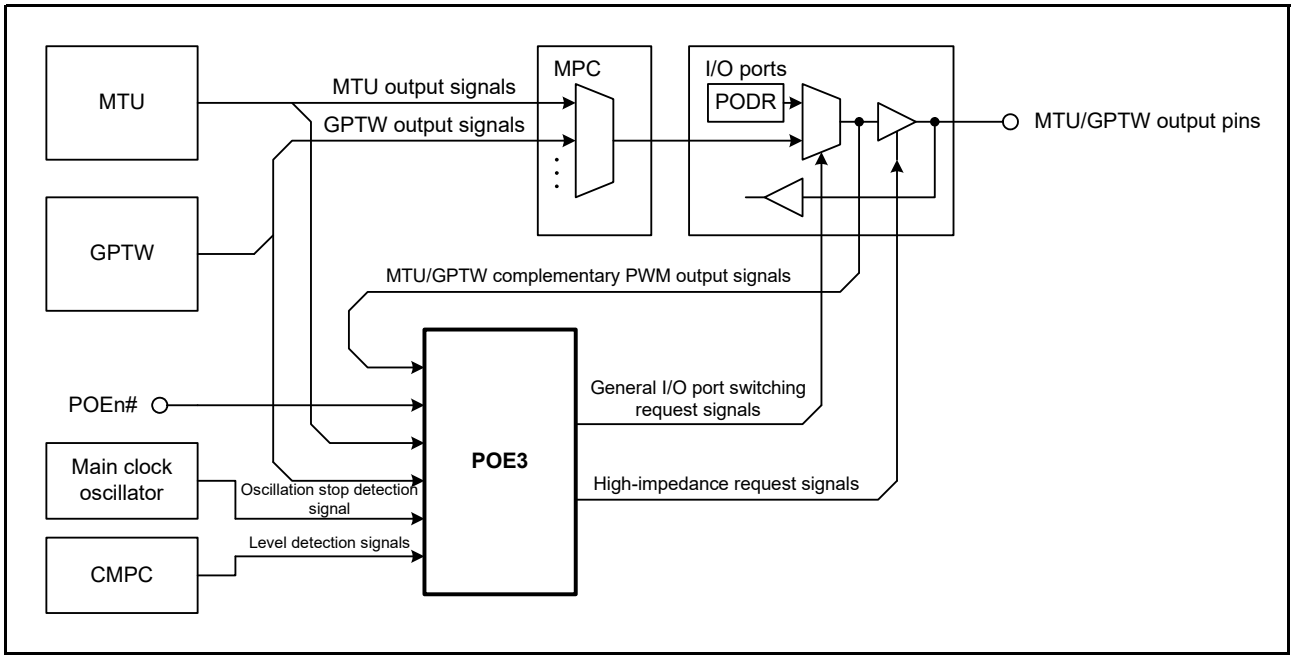


Figure 23.1 POE System Block Diagram

The POE has input-level detector, pin selector, output-level comparator, and a high-impedance request/port switching request/interrupt request generator as shown in Figure 23.2.

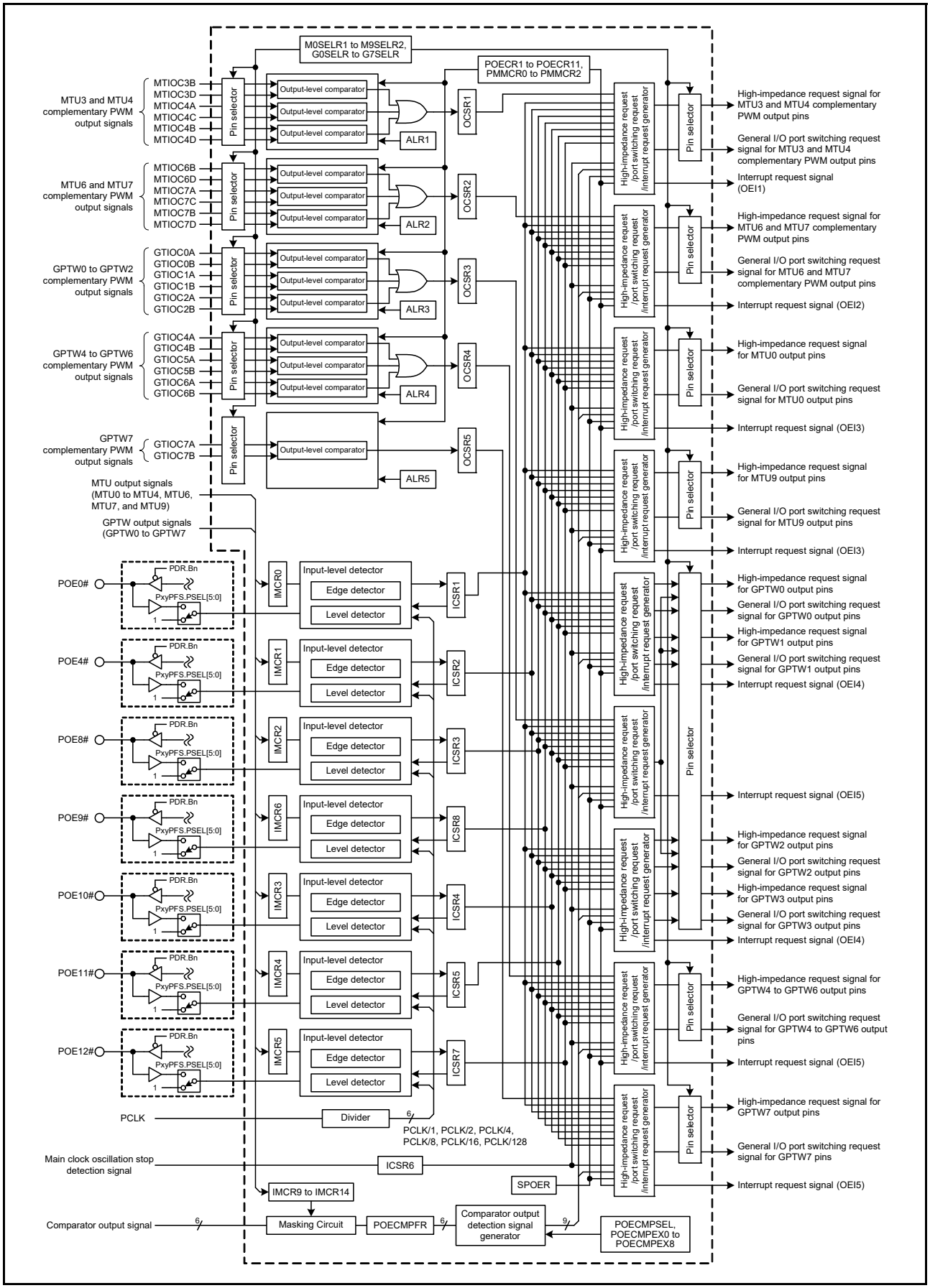


Figure 23.2 POE Block Diagram

Table 23.2 shows I/O pins to be used by the POE.

Table 23.2 POE I/O Pins

Pin Name	I/O	Description
POE0#	Input	Controls all the target pins by register settings. Assigned as a request signal to disable outputs of the MTU complementary PWM output pins (MTU3 and MTU4 pins) after reset.
POE4#	Input	Controls all the target pins by register settings. Assigned as a request signal to disable outputs of the MTU complementary PWM output pins (MTU6 and MTU7 pins) after reset.
POE8#	Input	Controls all the target pins by register settings. Assigned as a request signal to disable outputs of the MTU0 pins after reset.
POE9#	Input	Controls all the target pins by register settings. Assigned as a request signal to disable outputs of the MTU9 pins after reset.
POE10#	Input	Controls all the target pins by register settings. Assigned as a request signal to disable outputs of the GPTW0 and GPTW1 pins after reset.
POE11#	Input	Controls all the target pins by register settings. Assigned as a request signal to disable outputs of the GPTW2 and GPTW3 pins after reset.
POE12#	Input	Controls all the target pins by register settings. Assigned as a request signal to disable outputs of the GPTW0 to GPTW2 pins after reset.

Table 23.3 shows output-level comparisons with signal combinations.

Table 23.3 Output Signal Combinations

Signal Combination	I/O	Description
MTIOC3B and MTIOC3D	Output	The MTU complementary PWM output pins (MTU3 and MTU4 pins) are disabled when two signals in a pair simultaneously output the active level (low level when the MTU.TOCR1A.OLSP bit is 0 or high level when the OLSP bit is 1 while the OLSEN bit in the ALR1 register is 0 and the MTU.TOCR1A.TOCS bit is 0, low level when the OLS1P, OLS1N, OLS2P, OLS2N, OLS3P, and OLS3N bits in the MTU.TOCR2A register are 0 or high level when these bits are 1 while the OLSEN bit in the ALR1 register is 0 and the MTU.TOCR1A.TOCS bit is 1, or low level when the OLSG0A, OLSG0B, OLSG1A, OLSG1B, OLSG2A, and OLSG2B bits in the ALR1 register are 0 and high level when these bits are 1 while the OLSEN bit in the ALR1 register is 1) for one or more cycles of the peripheral module clock (PCLK). Pin combinations for output comparison can be selected by registers of POE.
MTIOC4A and MTIOC4C	Output	
MTIOC4B and MTIOC4D	Output	
MTIOC6B and MTIOC6D	Output	The MTU complementary PWM output pins (MTU6 and MTU7 pins) are disabled when two signals in a pair simultaneously output the active level (low level when the MTU.TOCR1B.OLSP bit is 0 or high level when the OLSP bit is 1 while the ALR2.OLSEN bit is 0 and the MTU.TOCR1B.TOCS bit is 0, or low level when the OLS1P, OLS1N, OLS2P, OLS2N, OLS3P, and OLS3N bits in the MTU.TOCR2B register are 0 or high level when these bits are 1 while the ALR2.OLSEN bit is 0 and the MTU.TOCR1B.TOCS bit is 1, or low level when the OLSG4A, OLSG4B, OLSG5A, OLSG5B, OLSG6A, and OLSG6B bits in the ALR2 register is 0 or high level when these bits are 1 while the ALR2.OLSEN bit is 1) for one or more cycles of the peripheral module clock (PCLK). Pin combinations for output comparison can be selected by registers of POE.
MTIOC7A and MTIOC7C	Output	
MTIOC7B and MTIOC7D	Output	
GTIOC0A and GTIOC0B	Output	The GPTW output pins (GPTW0 to GPTW2 pins) are disabled when two signals in a pair simultaneously output the active level (low level when the OLSG0A, OLSG0B, OLSG1A, OLSG1B, OLSG2A, and OLSG2B bits in the ALR3 register are 0, or high level when these bits are 1) for one or more cycles of the peripheral module clock (PCLK). Pin combinations for output comparison can be selected by registers of POE.
GTIOC1A and GTIOC1B	Output	
GTIOC2A and GTIOC2B	Output	
GTIOC4A and GTIOC4B	Output	The GPTW output pins (GPTW4 to GPTW6 pins) are disabled when two signals in a pair simultaneously output the active level (low level when the OLSG0A, OLSG0B, OLSG1A, OLSG1B, OLSG2A, and OLSG2B bits in the ALR4 register are 0, or high level when these bits are 1) for one or more cycles of the peripheral module clock (PCLK). Pin combinations for output comparison can be selected by registers of POE.
GTIOC5A and GTIOC5B	Output	
GTIOC6A and GTIOC6B	Output	
GTIOC7A and GTIOC7B	Output	The GPTW7 output pins are disabled when two signals in a pair simultaneously output the active level (low level when the OLSG0A and OLSG0B bits in the ALR5 register are 0, or high level when these bits are 1) for one or more cycles of the peripheral module clock (PCLK). Pin combinations for output comparison can be selected by registers of POE.

23.2 Register Descriptions

The POE registers are initialized by a reset.

23.2.1 Input Level Control/Status Register 1 (ICSR1)

Address(es): POE.ICSR1 0009 E400h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	INV	—	—	POE0F	—	—	—	PIE1	POE0M2[3:0]				POE0M[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	POE0M[3:0]	POE0 Mode Select	b3 b0 0 0 0 0: Accepts the request on the falling or rising edge of the POE0# pin input. 0 0 0 1: Samples the input from the POE0# pin with PCLK/8 and accepts the request when the low or high level is detected consecutively a specified number of times. 0 0 1 0: Samples the input from the POE0# pin with PCLK/16 and accepts the request when the low or high level is detected consecutively a specified number of times. 0 0 1 1: Samples the input from the POE0# pin with PCLK/128 and accepts the request when the low or high level is detected consecutively a specified number of times. 0 1 0 0: Samples the input from the POE0# pin with PCLK and accepts the request when the low or high level is detected consecutively a specified number of times. 0 1 0 1: Samples the input from the POE0# pin with PCLK/2 and accepts the request when the low or high level is detected consecutively a specified number of times. 0 1 1 0: Samples the input from the POE0# pin with PCLK/4 and accepts the request when the low or high level is detected consecutively a specified number of times. Settings other than above are prohibited.	R/W*1
b7 to b4	POE0M2[3:0]	POE0 Sampling Count Select	b7 b4 0 0 0 0: 16 times 0 0 0 1: 4 times 0 0 1 0: 8 times 0 0 1 1: 9 times 0 1 0 0: 10 times 0 1 0 1: 11 times 0 1 1 0: 12 times 0 1 1 1: 13 times 1 0 0 0: 14 times 1 0 0 1: 15 times Settings other than above are prohibited.	R/W*1
b8	PIE1	Port Interrupt Enable 1	0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
b11 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	POE0F	POE0 Flag	0: Indicates that an output disabling request has not been input to the POE0# pin. 1: Indicates that an output disabling request has been input to the POE0# pin.	R/(W)*2
b14, b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	INV	POE0# Pin Input Invert	0: Does not invert the input signal from the POE0# pin (falling edge or low-level detection) 1: Inverts the input signal from the POE0# pin (rising edge or high-level detection)	R/W

Note 1. Can be modified only once after a reset.

Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

The ICSR1 register selects the POE0# pin input modes, controls the enable/disable of interrupts, and indicates status.

POE0M[3:0] Bits (POE0 Mode Select)

These bits select detection mode for the POE0# pin input. Either of edge detection or level detection can be selected. The INV bit specifies which edge to use or which level to use.
When level detection is selected, a sampling interval can be specified.

POE0M2[3:0] Bits (POE0 Sampling Count Select)

These bits specify sampling count when level detection is selected.
When edge detection is selected, setting value for these bits is ignored.

PIE1 Bit (Port Interrupt Enable 1)

This bit enables or disables interrupt requests when the POE0F flag is set to 1.

POE0F Flag (POE0 Flag)

This flag indicates that an output disabling request has been input to the POE0# pin.
[Setting condition]

- When the input set by the POE0M[3:0] and POE0M2[3:0] bits occurs at the POE0# pin

[Clearing condition]

- By writing 0 to the POE0F flag after reading POE0F = 1
  When low-level detection is set by the POE0M[3:0] bits, the high level needs to be input to the POE0# pin to write 0 to this flag. Similarly, when high-level detection is set, the low level needs to be input to the POE0# pin to write 0 to this flag. For details, refer to section 23.3.10, Recover from Output Disabled State.

INV Bit (POE0# Pin Input Invert)

This bit logically inverts the input signal from the POE0# pin.
When this bit is 0, falling-edge detection or low-level detection can be selected. When it is 1, rising-edge detection or high-level detection can be selected.

23.2.2 Input Level Control/Status Register 2 (ICSR2)

Address(es): POE.ICSR2 0009 E404h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	INV	—	—	POE4F	—	—	—	PIE2	POE4M2[3:0]				POE4M[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	POE4M[3:0]	POE4 Mode Select	b3 b0 0 0 0 0: Accepts the request on the falling or rising edge of the POE4# pin input. 0 0 0 1: Samples the input from the POE4# pin with PCLK/8 and accepts the request when the low or high level is detected consecutively a specified number of times. 0 0 1 0: Samples the input from the POE4# pin with PCLK/16 and accepts the request when the low or high level is detected consecutively a specified number of times. 0 0 1 1: Samples the input from the POE4# pin with PCLK/128 and accepts the request when the low or high level is detected consecutively a specified number of times. 0 1 0 0: Samples the input from the POE4# pin with PCLK and accepts the request when the low or high level is detected consecutively a specified number of times. 0 1 0 1: Samples the input from the POE4# pin with PCLK/2 and accepts the request when the low or high level is detected consecutively a specified number of times. 0 1 1 0: Samples the input from the POE4# pin with PCLK/4 and accepts the request when the low or high level is detected consecutively a specified number of times. Settings other than above are prohibited.	R/W*1
b7 to b4	POE4M2[3:0]	POE4 Sampling Count Select	b7 b4 0 0 0 0: 16 times 0 0 0 1: 4 times 0 0 1 0: 8 times 0 0 1 1: 9 times 0 1 0 0: 10 times 0 1 0 1: 11 times 0 1 1 0: 12 times 0 1 1 1: 13 times 1 0 0 0: 14 times 1 0 0 1: 15 times Settings other than above are prohibited.	R/W*1
b8	PIE2	Port Interrupt Enable 2	0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
b11 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	POE4F	POE4 Flag	0: Indicates that an output disabling request has not been input to the POE4# pin. 1: Indicates that an output disabling request has been input to the POE4# pin.	R/(W)*2
b14, b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	INV	POE4# Pin Input Invert	0: Does not invert the input signal from the POE4# pin (falling edge or low-level detection) 1: Inverts the input signal from the POE4# pin (rising edge or high-level detection)	R/W

Note 1. Can be modified only once after a reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

The ICSR2 register selects the POE4# pin input mode, controls the enable/disable of interrupts, and indicates status.

POE4M[3:0] Bits (POE4 Mode Select)

These bits select detection mode for the POE4# pin input. Either of edge detection or level detection can be selected. The INV bit specifies which edge to use or which level to use.
When level detection is selected, a sampling interval can be specified.

POE4M2[3:0] Bits (POE4 Sampling Count Select)

These bits specify sampling count when level detection is selected.
When edge detection is selected, setting value for these bits is ignored.

PIE2 Bit (Port Interrupt Enable 2)

This bit enables or disables interrupt requests when the POE4F flag is set to 1.

POE4F Flag (POE4 Flag)

This flag indicates that an output disabling request has been input to the POE4# pin.
[Setting condition]

- When the input set by the POE4M[3:0] and POE4M2[3:0] bits occurs at the POE4# pin

[Clearing condition]

- By writing 0 to POE4F after reading POE4F = 1
  When low-level detection is set by the POE4M[3:0] bits, the high level needs to be input to the POE4# pin to write 0 to this flag. Similarly, when high-level detection is set, the low level needs to be input to the POE4# pin to write 0 to this flag. For details, refer to section 23.3.10, Recover from Output Disabled State.

INV Bit (POE4# Pin Input Invert)

This bit logically inverts the input signal from the POE4# pin.
When this bit is 0, falling-edge detection or low-level detection can be selected. When it is 1, rising-edge detection or high-level detection can be selected.

23.2.3 Input Level Control/Status Register 3 (ICSR3)

Address(es): POE.ICSR3 0009 E408h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	INV	—	—	POE8F	—	—	POE8E	PIE3	POE8M2[3:0]				POE8M[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	POE8M[3:0]	POE8 Mode Select	b3 b0 0 0 0 0: Accepts the request on the falling or rising edge of the POE8# pin input. 0 0 0 1: Samples the input from the POE8# pin with PCLK/8 and accepts the request when the low or high level is detected consecutively a specified number of times. 0 0 1 0: Samples the input from the POE8# pin with PCLK/16 and accepts the request when the low or high level is detected consecutively a specified number of times. 0 0 1 1: Samples the input from the POE8# pin with PCLK/128 and accepts the request when the low or high level is detected consecutively a specified number of times. 0 1 0 0: Samples the input from the POE8# pin with PCLK and accepts the request when the low or high level is detected consecutively a specified number of times. 0 1 0 1: Samples the input from the POE8# pin with PCLK/2 and accepts the request when the low or high level is detected consecutively a specified number of times. 0 1 1 0: Samples the input from the POE8# pin with PCLK/4 and accepts the request when the low or high level is detected consecutively a specified number of times. Settings other than above are prohibited.	R/W*1
b7 to b4	POE8M2[3:0]	POE8 Sampling Count Select	b7 b4 0 0 0 0: 16 times 0 0 0 1: 4 times 0 0 1 0: 8 times 0 0 1 1: 9 times 0 1 0 0: 10 times 0 1 0 1: 11 times 0 1 1 0: 12 times 0 1 1 1: 13 times 1 0 0 0: 14 times 1 0 0 1: 15 times Settings other than above are prohibited.	R/W*1
b8	PIE3	Port Interrupt Enable 3	0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
b9	POE8E	POE8 Output Disable	0: Does not disable the output by POE8# signal. 1: Disable the output by POE8# signal.	R/W*1
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	POE8F	POE8 Flag	0: Indicates that an output disabling request has not been input to the POE8# pin. 1: Indicates that an output disabling request has been input to the POE8# pin.	R/(W)*2
b14, b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	INV	POE8# Pin Input Invert	0: Does not invert the input signal from the POE8# pin (falling edge or low-level detection) 1: Inverts the input signal from the POE8# pin (rising edge or high-level detection)	R/W

Note 1. Can be modified only once after a reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

The ICSR3 register selects the POE8# pin input mode, controls the enable/disable of interrupts, and indicates status.

POE8M[3:0] Bits (POE8 Mode Select)

These bits select detection mode for the POE8# pin input. Either of edge detection or level detection can be selected. The INV bit specifies which edge to use or which level to use.
When level detection is selected, a sampling interval can be specified.

POE8M2[3:0] Bits (POE8 Sampling Count Select)

These bits specify sampling count when level detection is selected.
When edge detection is selected, setting value for these bits is ignored.

PIE3 Bit (Port Interrupt Enable 3)

This bit enables or disables interrupt requests when the POE8F flag is set to 1.

POE8E Bit (POE8 Output Disable)

This bit specifies whether to disable the output of the corresponding pin when the POE8F flag is set to 1.

POE8F Flag (POE8 Flag)

This flag indicates that an output disabling request has been input to the POE8# pin.
[Setting condition]

- When the input set by the POE8M[3:0] and POE8M2[3:0] bits occurs at the POE8# pin

[Clearing condition]

- By writing 0 to the POE8F flag after reading POE8F = 1
  When low-level detection is set by the POE8M[3:0] bits, the high level needs to be input to the POE8# pin to write 0 to this flag. Similarly, when high-level detection is set, the low level needs to be input to the POE8# pin to write 0 to this flag. For details, refer to section 23.3.10, Recover from Output Disabled State.

INV Bit (POE8# Pin Input Invert)

This bit logically inverts the input signal from the POE8# pin.
When this bit is 0, falling-edge detection or low-level detection can be selected. When it is 1, rising-edge detection or high-level detection can be selected.

23.2.4 Input Level Control/Status Register 4 (ICSR4)

Address(es): POE.ICSR4 0009 E416h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	INV	—	—	POE10F	—	—	POE10E	PIE4	POE10M2[3:0]				POE10M[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	POE10M[3:0]	POE10 Mode Select	b3 b0 0 0 0 0: Accepts the request on the falling or rising edge of the POE10# pin input. 0 0 0 1: Samples the input from the POE10# pin with PCLK/8 and accepts the request when the low or high level is detected consecutively a specified number of times. 0 0 1 0: Samples the input from the POE10# pin with PCLK/16 and accepts the request when the low or high level is detected consecutively a specified number of times. 0 0 1 1: Samples the input from the POE10# pin with PCLK/128 and accepts the request when the low or high level is detected consecutively a specified number of times. 0 1 0 0: Samples the input from the POE10# pin with PCLK and accepts the request when the low or high level is detected consecutively a specified number of times. 0 1 0 1: Samples the input from the POE10# pin with PCLK/2 and accepts the request when the low or high level is detected consecutively a specified number of times. 0 1 1 0: Samples the input from the POE10# pin with PCLK/4 and accepts the request when the low or high level is detected consecutively a specified number of times. Settings other than above are prohibited.	R/W*1
b7 to b4	POE10M2[3:0]	POE10 Sampling Count Select	b7 b4 0 0 0 0: 16 times 0 0 0 1: 4 times 0 0 1 0: 8 times 0 0 1 1: 9 times 0 1 0 0: 10 times 0 1 0 1: 11 times 0 1 1 0: 12 times 0 1 1 1: 13 times 1 0 0 0: 14 times 1 0 0 1: 15 times Settings other than above are prohibited.	R/W*1
b8	PIE4	Port Interrupt Enable 4	0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
b9	POE10E	POE10 Output Disable	0: Does not disable the output by POE10# signal. 1: Disable the output by POE10# signal.	R/W*1
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	POE10F	POE10 Flag	0: Indicates that an output disabling request has not been input to the POE10# pin. 1: Indicates that an output disabling request has been input to the POE10# pin.	R/(W)*2
b14, b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	INV	POE10# Pin Input Invert	0: Does not invert the input signal from the POE10# pin (falling edge or low-level detection) 1: Inverts the input signal from the POE10# pin (rising edge or high-level detection)	R/W

Note 1. Can be modified only once after a reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

The ICSR4 register selects the POE10# pin input mode, controls the enable/disable of interrupts, and indicates status.

POE10M[3:0] Bits (POE10 Mode Select)

These bits select detection mode for the POE10# pin input. Either of edge detection or level detection can be selected.
The INV bit specifies which edge to use or which level to use.
When level detection is selected, a sampling interval can be specified.

POE10M2[3:0] Bits (POE10 Sampling Count Select)

These bits specify sampling count when level detection is selected.
When edge detection is selected, setting value for these bits is ignored.

PIE4 Bit (Port Interrupt Enable 4)

This bit enables or disables interrupt requests when the POE10F flag is set to 1.

POE10E Bit (POE10 Output Disable)

This bit specifies whether to disable the output of the corresponding pin when the POE10F flag is set to 1.

POE10F Flag (POE10 Flag)

This flag indicates that an output disabling request has been input to the POE10# pin.
[Setting condition]

- When the input set by the POE10M[3:0] and POE10M2[3:0] bits occurs at the POE10# pin

[Clearing condition]

- By writing 0 to the POE10F flag after reading POE10F = 1
  When low-level detection is set by the POE10M[3:0] bits, the high level needs to be input to the POE10# pin to write 0 to this flag. Similarly, when high-level detection is set, the low level needs to be input to the POE10# pin to write 0 to this flag. For details, refer to section 23.3.10, Recover from Output Disabled State.

INV Bit (POE10# Pin Input Invert)

This bit logically inverts the input signal from the POE10# pin.
When this bit is 0, falling-edge detection or low-level detection can be selected. When it is 1, rising-edge detection or high-level detection can be selected.

23.2.5 Input Level Control/Status Register 5 (ICSR5)

Address(es): POE.ICSR5 0009 E418h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	INV	—	—	POE11F	—	—	POE11E	PIE5	POE11M2[3:0]				POE11M[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	POE11M[3:0]	POE11 Mode Select	b3 b0 0 0 0 0: Accepts the request on the falling or rising edge of the POE11# pin input. 0 0 0 1: Samples the input from the POE11# pin with PCLK/8 and accepts the request when the low or high level is detected consecutively a specified number of times. 0 0 1 0: Samples the input from the POE11# pin with PCLK/16 and accepts the request when the low or high level is detected consecutively a specified number of times. 0 0 1 1: Samples the input from the POE11# pin with PCLK/128 and accepts the request when the low or high level is detected consecutively a specified number of times. 0 1 0 0: Samples the input from the POE11# pin with PCLK and accepts the request when the low or high level is detected consecutively a specified number of times. 0 1 0 1: Samples the input from the POE11# pin with PCLK/2 and accepts the request when the low or high level is detected consecutively a specified number of times. 0 1 1 0: Samples the input from the POE11# pin with PCLK/4 and accepts the request when the low or high level is detected consecutively a specified number of times. Settings other than above are prohibited.	R/W*1
b7 to b4	POE11M2[3:0]	POE11 Sampling Count Select	b7 b4 0 0 0 0: 16 times 0 0 0 1: 4 times 0 0 1 0: 8 times 0 0 1 1: 9 times 0 1 0 0: 10 times 0 1 0 1: 11 times 0 1 1 0: 12 times 0 1 1 1: 13 times 1 0 0 0: 14 times 1 0 0 1: 15 times Settings other than above are prohibited.	R/W*1
b8	PIE5	Port Interrupt Enable 5	0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
b9	POE11E	POE11 Output Disable	0: Does not disable the output by POE11# signal. 1: Disable the output by POE11# signal.	R/W*1
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	POE11F	POE11 Flag	0: Indicates that an output disabling request has not been input to the POE11# pin. 1: Indicates that an output disabling request has been input to the POE11# pin.	R/(W)*2
b14, b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	INV	POE11# Pin Input Invert	0: Does not invert the input signal from the POE11# pin (falling edge or low-level detection) 1: Inverts the input signal from the POE11# pin (rising edge or high-level detection)	R/W

Note 1. Can be modified only once after a reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

The ICSR5 register selects the POE11# pin input mode, controls the enable/disable of interrupts, and indicates status.

POE11M[3:0] Bits (POE11 Mode Select)

These bits select detection mode for the POE11# pin input. Either of edge detection or level detection can be selected. The INV bit specifies which edge to use or which level to use.
When level detection is selected, a sampling interval can be specified.

POE11M2[3:0] Bits (POE11 Sampling Count Select)

These bits specify sampling count when level detection is selected.
When edge detection is selected, setting value for these bits is ignored.

PIE5 Bit (Port Interrupt Enable 5)

This bit enables or disables interrupt requests when the POE11F flag is set to 1.

POE11E Bit (POE11 Output Disable)

This bit specifies whether to disable the output of the corresponding pin when the POE11F flag is set to 1.

POE11F Flag (POE11 Flag)

This flag indicates that an output disabling request has been input to the POE11# pin.
[Setting condition]

- When the input set by the POE11M[3:0] and POE11M2[3:0] bits occurs at the POE11# pin

[Clearing condition]

- By writing 0 to the POE11F flag after reading POE11F = 1
  When low-level detection is set by the POE11M[3:0] bits, the high level needs to be input to the POE11# pin to write 0 to this flag. Similarly, when high-level detection is set, the low level needs to be input to the POE11# pin to write 0 to this flag. For details, refer to section 23.3.10, Recover from Output Disabled State.

INV Bit (POE11# Pin Input Invert)

This bit logically inverts the input signal from the POE11# pin.
When this bit is 0, falling-edge detection or low-level detection can be selected. When it is 1, rising-edge detection or high-level detection can be selected.

23.2.6 Input Level Control/Status Register 6 (ICSR6)

Address(es): POE.ICSR6 0009 E41Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	OSTSTF	—	—	OSTSTE	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b8 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9	OSTSTE	Oscillation Stop Output Disable	0: Does not disable the output when the oscillation stop is detected. 1: Disable the output when the oscillation stop is detected.	R/W*1
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	OSTSTF	Oscillation Stop Detection Flag	0: Indicates that an output disabling request by oscillation stop has not been generated. 1: Indicates that an output disabling request by oscillation stop has been generated.	R/W*2
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

The ICSR6 register controls the oscillation stop output disabling and indicates status.

OSTSTE Bit (Oscillation Stop Output Disable)

This bit specifies whether to disable the output of the target pin when oscillation stop is detected.

OSTSTF Flag (Oscillation Stop Detection Flag)

This flag indicates that an output disabling request by the oscillation stop has been generated.
When the main clock oscillation stops, this flag is set to 1. To clear this flag, wait for at least 10 cycles of PCLK after this flag becomes 1 and write 0 to this flag while the OSTDSR.OSTDF flag is 0. Writing 0 to this flag while the OSTDSR.OSTDF flag is 1 cannot clear this flag. After clearing this flag, confirm that the flag has actually been modified to 0.
[Setting condition]

- When oscillation stop is detected

[Clearing condition]

- By writing 0 to the OSTSTF flag after reading OSTSTF = 1

23.2.7 Input Level Control/Status Register 7 (ICSR7)

Address(es): POE.ICSR7 0009 E420h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	INV	—	—	POE12F	—	—	POE12E	PIE7	POE12M2[3:0]				POE12M[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	POE12M[3:0]	POE12 Mode Select	b3 b0 0 0 0 0: Accepts the request on the falling or rising edge of the POE12# pin input. 0 0 0 1: Samples the input from the POE12# pin with PCLK/8 and accepts the request when the low or high level is detected consecutively a specified number of times. 0 0 1 0: Samples the input from the POE12# pin with PCLK/16 and accepts the request when the low or high level is detected consecutively a specified number of times. 0 0 1 1: Samples the input from the POE12# pin with PCLK/128 and accepts the request when the low or high level is detected consecutively a specified number of times. 0 1 0 0: Samples the input from the POE12# pin with PCLK and accepts the request when the low or high level is detected consecutively a specified number of times. 0 1 0 1: Samples the input from the POE12# pin with PCLK/2 and accepts the request when the low or high level is detected consecutively a specified number of times. 0 1 1 0: Samples the input from the POE12# pin with PCLK/4 and accepts the request when the low or high level is detected consecutively a specified number of times. Settings other than above are prohibited.	R/W*1
b7 to b4	POE12M2[3:0]	POE12 Sampling Count Select	b7 b4 0 0 0 0: 16 times 0 0 0 1: 4 times 0 0 1 0: 8 times 0 0 1 1: 9 times 0 1 0 0: 10 times 0 1 0 1: 11 times 0 1 1 0: 12 times 0 1 1 1: 13 times 1 0 0 0: 14 times 1 0 0 1: 15 times Settings other than above are prohibited.	R/W*1
b8	PIE7	Port Interrupt Enable 7	0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
b9	POE12E	POE12 Output Disable	0: Does not disable the output by POE12# signal. 1: Disable the output by POE12# signal.	R/W*1
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	POE12F	POE12 Flag	0: Indicates that an output disabling request has not been input to the POE12# pin. 1: Indicates that an output disabling request has been input to the POE12# pin.	R/(W)*2
b14, b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	INV	POE12# Pin Input Invert	0: Does not invert the input signal from the POE12# pin (falling edge or low-level detection) 1: Inverts the input signal from the POE12# pin (rising edge or high-level detection)	R/W

Note 1. Can be modified only once after a reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

The ICSR7 register selects the POE12# pin input mode, controls the enable/disable of interrupts, and indicates status.

POE12M[3:0] Bits (POE12 Mode Select)

These bits select detection mode for the POE12# pin input. Either of edge detection or level detection can be selected. The INV bit specifies which edge to use or which level to use.
When level detection is selected, a sampling interval can be specified.

POE12M2[3:0] Bits (POE12 Sampling Count Select)

These bits specify sampling count when level detection is selected.
When edge detection is selected, setting value for these bits is ignored.

PIE7 Bit (Port Interrupt Enable 7)

This bit enables or disables interrupt requests when the POE12F flag is set to 1.

POE12E Bit (POE12 Output Disable)

This bit specifies whether to disable the output of the corresponding pin when the POE12F flag is set to 1.

POE12F Flag (POE12 Flag)

This flag indicates that an output disabling request has been input to the POE12# pin.
[Setting condition]

- When the input set by the POE12M[3:0] and POE12M2[3:0] bits occurs at the POE12# pin

[Clearing condition]

- By writing 0 to the POE12F flag after reading POE12F = 1
  When low-level detection is set by the POE12M[3:0] bits, the high level needs to be input to the POE12# pin to write 0 to this flag. Similarly, when high-level detection is set, the low level needs to be input to the POE12# pin to write 0 to this flag. For details, refer to section 23.3.10, Recover from Output Disabled State.

INV Bit (POE12# Pin Input Invert)

This bit logically inverts the input signal from the POE12# pin.
When this bit is 0, falling-edge detection or low-level detection can be selected. When it is 1, rising-edge detection or high-level detection can be selected.

23.2.8 Input Level Control/Status Register 8 (ICSR8)

Address(es): POE.ICSR8 0009 E440h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	INV	—	—	POE9F	—	—	POE9E	PIE8	POE9M2[3:0]				POE9M[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	POE9M[3:0]	POE9 Mode Select	b3 b0 0 0 0 0: Accepts the request on the falling or rising edge of the POE9# pin input. 0 0 0 1: Samples the input from the POE9# pin with PCLK/8 and accepts the request when the low or high level is detected consecutively a specified number of times. 0 0 1 0: Samples the input from the POE9# pin with PCLK/16 and accepts the request when the low or high level is detected consecutively a specified number of times. 0 0 1 1: Samples the input from the POE9# pin with PCLK/128 and accepts the request when the low or high level is detected consecutively a specified number of times. 0 1 0 0: Samples the input from the POE9# pin with PCLK and accepts the request when the low or high level is detected consecutively a specified number of times. 0 1 0 1: Samples the input from the POE9# pin with PCLK/2 and accepts the request when the low or high level is detected consecutively a specified number of times. 0 1 1 0: Samples the input from the POE9# pin with PCLK/4 and accepts the request when the low or high level is detected consecutively a specified number of times. Settings other than above are prohibited.	R/W*1
b7 to b4	POE9M2[3:0]	POE9 Sampling Count Select	b7 b4 0 0 0 0: 16 times 0 0 0 1: 4 times 0 0 1 0: 8 times 0 0 1 1: 9 times 0 1 0 0: 10 times 0 1 0 1: 11 times 0 1 1 0: 12 times 0 1 1 1: 13 times 1 0 0 0: 14 times 1 0 0 1: 15 times Settings other than above are prohibited.	R/W*1
b8	PIE8	Port Interrupt Enable 8	0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
b9	POE9E	POE9 Output Disable	0: Does not disable the output by POE9# signal. 1: Disable the output by POE9# signal.	R/W*1
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	POE9F	POE9 Flag	0: Indicates that an output disabling request has not been input to the POE9# pin. 1: Indicates that an output disabling request has been input to the POE9# pin.	R/(W)*2
b14, b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	INV	POE9# Pin Input Invert	0: Does not invert the input signal from the POE9# pin (falling edge or low-level detection) 1: Inverts the input signal from the POE9# pin (rising edge or high-level detection)	R/W

Note 1. Can be modified only once after a reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

The ICSR8 register selects the POE9# pin input mode, controls the enable/disable of interrupts, and indicates status.

POE9M[3:0] Bits (POE9 Mode Select)

These bits select detection mode for the POE9# pin input. Either of edge detection or level detection can be selected. The INV bit specifies which edge to use or which level to use.
When level detection is selected, a sampling interval can be specified.

POE9M2[3:0] Bits (POE9 Sampling Count Select)

These bits specify sampling count when level detection is selected.
When edge detection is selected, setting value for these bits is ignored.

PIE8 Bit (Port Interrupt Enable 8)

This bit enables or disables interrupt requests when the POE9F flag is set to 1.

POE9E Bit (POE9 Output Disable)

This bit specifies whether to disable the output of the corresponding pin when the POE9F flag is set to 1.

POE9F Flag (POE9 Flag)

This flag indicates that an output disabling request has been input to the POE9# pin.
[Setting condition]

- When the input set by the POE9M[3:0] and POE9M2[3:0] bits occurs at the POE9# pin

[Clearing condition]

- By writing 0 to the POE9F flag after reading POE9F = 1
  When low-level detection is set by the POE9M[3:0] bits, the high level needs to be input to the POE9# pin to write 0 to this flag. Similarly, when high-level detection is set, the low level needs to be input to the POE9# pin to write 0 to this flag. For details, refer to section 23.3.10, Recover from Output Disabled State.

INV Bit (POE9# Pin Input Invert)

This bit logically inverts the input signal from the POE9# pin.
When this bit is 0, falling-edge detection or low-level detection can be selected. When it is 1, rising-edge detection or high-level detection can be selected.

23.2.9 Output Level Control/Status Register 1 (OCSR1)

Address(es): POE.OCSR1 0009 E402h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OSF1	—	—	—	—	—	OCE1	OIE1	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	OIE1	Simultaneous Conduction Interrupt Enable 1	0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
b9	OCE1	Simultaneous Conduction Output Disable 1	0: Does not disable the outputs when they simultaneously go to an active level. 1: Disable the outputs when they simultaneously go to an active level.	R/W*1
b14 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	OSF1	Simultaneous Conduction Flag 1	0: Indicates that outputs have not simultaneously become an active level. 1: Indicates that outputs have simultaneously become an active level.	R/(W)*2

Note 1. Can be modified only once after a reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

The OCSR1 register enables/disables comparison of output levels, enables/disables a simultaneous conduction interrupt, and indicates the status for the MTU3 and MTU4 pins.

OIE1 Bit (Simultaneous Conduction Interrupt Enable 1)

This bit enables or disables interrupt requests when the OSF1 flag is set to 1.

OCE1 Bit (Simultaneous Conduction Output Disable 1)

This bit specifies whether to disable the output of the corresponding pin when the OSF1 flag is set to 1.

OSF1 Flag (Simultaneous Conduction Flag 1)

This flag indicates that at least one of the three pairs of two-phase output signals for MTU complementary PWM output (MTU3 and MTU4) has simultaneously become at the active level. If the output disabling control for the corresponding pins is not enabled, this flag does not become 1.
For setting the active level, refer to section 23.2.14, Active Level Setting Register 1 (ALR1).
[Setting condition]

- When the MTIOC3B and MTIOC3D signals simultaneously go to the active level*1 for at least one cycle of PCLK while the value of the POECR2.MTU3BDZE bit, or either or both of the PMMCR1.MTU3BME and PMMCR1.MTU3DME bits, is 1.
- When the MTIOC4A and MTIOC4C signals simultaneously go to the active level*1 for at least one cycle of PCLK while the value of the POECR2.MTU4ACZE bit, or either or both of the PMMCR1.MTU4AME and PMMCR1.MTU4CME bits, is 1.
- When the MTIOC4B and MTIOC4D signals simultaneously go to the active level*1 for at least one cycle of PCLK while the value of the POECR2.MTU4BDZE bit, or either or both of the PMMCR1.MTU4BME and PMMCR1.MTU4DME bits, is 1.

Note 1. The setting condition is judged only by the level of the signal regardless the setting of the MPC.PxyPFS register.

[Clearing condition]

- By writing 0 to the OSF1 flag after reading OSF1 = 1

To write 0 to this flag, the MTU complementary PWM output signals should be set to the inactive level. For details, refer to section 23.3.10, Recover from Output Disabled State.

23.2.10 Output Level Control/Status Register 2 (OCSR2)

Address(es): POE.OCSR2 0009 E406h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OSF2	—	—	—	—	—	OCE2	OIE2	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	OIE2	Simultaneous Conduction Interrupt Enable 2	0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
b9	OCE2	Simultaneous Conduction Output Disable 2	0: Does not disable the outputs when they simultaneously go to an active level. 1: Disable the outputs when they simultaneously go to an active level.	R/W*1
b14 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	OSF2	Simultaneous Conduction Flag 2	0: Indicates that outputs have not simultaneously become an active level. 1: Indicates that outputs have simultaneously become an active level.	R/(W)*2

Note 1. Can be modified only once after a reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

The OCSR2 register enables/disables comparison of output levels, enables/disables a simultaneous conduction interrupt, and indicates the status for the MTU6 and MTU7 pins.

OIE2 Bit (Simultaneous Conduction Interrupt Enable 2)

This bit enables or disables interrupt requests when the OSF2 flag is set to 1.

OCE2 Bit (Simultaneous Conduction Output Disable 2)

This bit specifies whether to disable the output of the corresponding pin when the OSF2 flag is set to 1.

OSF2 Flag (Simultaneous Conduction Flag 2)

This flag indicates that at least one of the three pairs of two-phase output signals for MTU complementary PWM output (MTU6 and MTU7) has simultaneously become at the active level. If the output disabling control for the corresponding pins is not enabled, this flag does not become 1.

For setting the active level, refer to section 23.2.15, Active Level Setting Register 2 (ALR2).

[Setting condition]

- When the MTIOC6B and MTIOC6D signals simultaneously go to the active level*1 for at least one cycle of PCLK while the value of the POECR2.MTU6BDZE bit, or either or both of the PMMCR1.MTU6BME and PMMCR1.MTU6DME bits, is 1.
- When the MTIOC7A and MTIOC7C signals simultaneously go to the active level*1 for at least one cycle of PCLK

while the value of the POECR2.MTU7ACZE bit, or either or both of the PMMCR1.MTU7AME and PMMCR1.MTU7CME bits, is 1.

- When the MTIOC7B and MTIOC7D signals simultaneously go to the active level*1 for at least one cycle of PCLK while the value of the POECR2.MTU7BDZE bit, or either or both of the PMMCR1.MTU7BME and PMMCR1.MTU7DME bits, is 1.

Note 1. The setting condition is judged only by the level of the signal regardless the setting of the MPC.PxyPFS register.

[Clearing condition]

- By writing 0 to the OSF2 flag after reading OSF2 = 1

To write 0 to this flag, the MTU complementary PWM output signals should be set to the inactive level. For details, refer to section 23.3.10, Recover from Output Disabled State.

23.2.11 Output Level Control/Status Register 3 (OCSR3)

Address(es): POE.OCSR3 0009 E42Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OSF3	—	—	—	—	—	OCE3	OIE3	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	OIE3	Simultaneous Conduction Interrupt Enable 3	0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
b9	OCE3	Simultaneous Conduction Output Disable 3	0: Does not disable the outputs when they simultaneously go to an active level. 1: Disable the outputs when they simultaneously go to an active level.	R/W*1
b14 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	OSF3	Simultaneous Conduction Flag 3	0: Indicates that outputs have not simultaneously become an active level. 1: Indicates that outputs have simultaneously become an active level.	R/(W)*2

Note 1. Can be modified only once after a reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

The OCSR3 register enables/disables comparison of output levels, enables/disables a simultaneous conduction interrupt, and indicates the status for the GPTW0 to GPTW2 pins.

OIE3 Bit (Simultaneous Conduction Interrupt Enable 3)

This bit enables or disables interrupt requests when the OSF3 flag is set to 1.

OCE3 Bit (Simultaneous Conduction Output Disable 3)

This bit specifies whether to disable the output of the corresponding pin when the OSF3 flag is set to 1.

OSF3 Flag (Simultaneous Conduction Flag 3)

This flag indicates that at least one of the three pairs of two-phase output signals for GPTW complementary PWM output

(GPTW0 to GPTW2) has simultaneously become at the active level. If the output disabling control for the corresponding pins is not enabled, this flag does not become 1.
For setting the active level, refer to section 23.2.16, Active Level Setting Register 3 (ALR3).
[Setting condition]

- When the GTIOC0A and GTIOC0B signals simultaneously go to the active level*1 for at least one cycle of PCLK while the value of the POECR3.GPT0ABZE bit, or either or both of the PMMCR2.GPT0AME and PMMCR2.GPT0BME bits, is 1.
- When the GTIOC1A and GTIOC1B signals simultaneously go to the active level*1 for at least one cycle of PCLK while the value of the POECR3.GPT1ABZE bit, or either or both of the PMMCR2.GPT1AME and PMMCR2.GPT1BME bits, is 1.
- When the GTIOC2A and GTIOC2B signals simultaneously go to the active level*1 for at least one cycle of PCLK while the value of the POECR3.GPT2ABZE bit, or either or both of the PMMCR2.GPT2AME and PMMCR2.GPT2BME bits, is 1.

Note 1. The setting condition is judged only by the level of the signal regardless the setting of the MPC.PxyPFS register.

[Clearing condition]

- By writing 0 to the OSF3 flag after reading OSF3 = 1

To write 0 to this flag, the GPTW complementary PWM output signals should be set to the inactive level. For details, refer to section 23.3.10, Recover from Output Disabled State.

23.2.12 Output Level Control/Status Register 4 (OCSR4)

Address(es): POE.OCSR4 0009 E446h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OSF4	—	—	—	—	—	OCE4	OIE4	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	OIE4	Simultaneous Conduction Interrupt Enable 4	0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
b9	OCE4	Simultaneous Conduction Output Disable 4	0: Does not disable the outputs when they simultaneously go to an active level. 1: Disable the outputs when they simultaneously go to an active level.	R/W*1
b14 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	OSF4	Simultaneous Conduction Flag 4	0: Indicates that outputs have not simultaneously become an active level. 1: Indicates that outputs have simultaneously become an active level.	R/(W)*2

Note 1. Can be modified only once after a reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

The OCSR4 register enables/disables comparison of output levels, enables/disables a simultaneous conduction interrupt, and indicates the status for the GPTW4 to GPTW6 pins.

OIE4 Bit (Simultaneous Conduction Interrupt Enable 4)

This bit enables or disables interrupt requests when the OSF4 flag is set to 1.

OCE4 Bit (Simultaneous Conduction Output Disable 4)

This bit specifies whether to disable the output of the corresponding pin when the OSF4 flag is set to 1.

OSF4 Flag (Simultaneous Conduction Flag 4)

This flag indicates that at least one of the three pairs of two-phase output signals for GPTW complementary PWM output (GPTW4 to GPTW6) has simultaneously become at the active level. If the output disabling control for the corresponding pins is not enabled, this flag does not become 1.
For setting the active level, refer to section 23.2.17, Active Level Setting Register 4 (ALR4).
[Setting condition]

- When the GTIOC4A and GTIOC4B signals simultaneously go to the active level*1 for at least one cycle of PCLK while the value of the POECR3.GPT4ABZE bit, or either or both of the PMMCR2.GPT4AME and PMMCR2.GPT4BME bits, is 1.
- When the GTIOC5A and GTIOC5B signals simultaneously go to the active level*1 for at least one cycle of PCLK while the value of the POECR3.GPT5ABZE bit, or either or both of the PMMCR2.GPT5AME and PMMCR2.GPT5BME bits, is 1.
- When the GTIOC6A and GTIOC6B signals simultaneously go to the active level*1 for at least one cycle of PCLK while the value of the POECR3.GPT6ABZE bit, or either or both of the PMMCR2.GPT6AME and PMMCR2.GPT6BME bits, is 1.

Note 1. The setting condition is judged only by the level of the signal regardless the setting of the MPC.PxyPFS register.

[Clearing condition]

- By writing 0 to the OSF4 flag after reading OSF4 = 1

To write 0 to this flag, the GPTW complementary PWM output signals should be set to the inactive level. For details, refer to section 23.3.10, Recover from Output Disabled State.

23.2.13 Output Level Control/Status Register 5 (OCSR5)

Address(es): POE.OCSR5 0009 E448h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OSF5	—	—	—	—	—	OCE5	OIE5	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	OIE5	Simultaneous Conduction Interrupt Enable 5	0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
b9	OCE5	Simultaneous Conduction Output Disable 5	0: Does not disable the outputs when they simultaneously go to an active level. 1: Disable the outputs when they simultaneously go to an active level.	R/W*1
b14 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	OSF5	Simultaneous Conduction Flag 5	0: Indicates that outputs have not simultaneously become an active level. 1: Indicates that outputs have simultaneously become an active level.	R/(W)*2

Note 1. Can be modified only once after a reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

The OCSR5 register enables/disables comparison of output levels, enables/disables a simultaneous conduction interrupt, and indicates the status for the GPTW7 pins.

OIE5 Bit (Simultaneous Conduction Interrupt Enable 5)

This bit enables or disables interrupt requests when the OSF5 flag is set to 1.

OCE5 Bit (Simultaneous Conduction Output Disable 5)

This bit specifies whether to disable the output of the corresponding pin when the OSF5 flag is set to 1.

OSF5 Flag (Simultaneous Conduction Flag 5)

This flag indicates that the two-phase output signals for GPTW complementary PWM output (GPTW7) has simultaneously become at the active level. If the output disabling control for the corresponding pins is not enabled, this flag does not become 1.
For setting the active level, refer to section 23.2.18, Active Level Setting Register 5 (ALR5).
[Setting condition]

- When the GTIOC7A and GTIOC7B signals simultaneously go to the active level*1 for at least one cycle of PCLK while the value of the POECR3.GPT7ABZE bit, or either or both of the PMMCR2.GPT7AME and PMMCR2.GPT7BME bits, is 1.

Note 1. The setting condition is judged only by the level of the signal regardless the setting of the MPC.PxyPFS register.

[Clearing condition]

- By writing 0 to the OSF5 flag after reading OSF5 = 1

To write 0 to this flag, the GPTW complementary PWM output signals should be set to the inactive level. For details, refer to section 23.3.10, Recover from Output Disabled State.

23.2.14 Active Level Setting Register 1 (ALR1)

Address(es): POE.ALR1 0009 E41Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	OLSEN	—	OLSG2B	OLSG2A	OLSG1B	OLSG1A	OLSG0B	OLSG0A
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OLSG0A	MTIOC3B Signal Active Level Setting	0: Active low 1: Active high	R/W*1
b1	OLSG0B	MTIOC3D Signal Active Level Setting	0: Active low 1: Active high	R/W*1
b2	OLSG1A	MTIOC4A Signal Active Level Setting	0: Active low 1: Active high	R/W*1
b3	OLSG1B	MTIOC4C Signal Active Level Setting	0: Active low 1: Active high	R/W*1
b4	OLSG2A	MTIOC4B Signal Active Level Setting	0: Active low 1: Active high	R/W*1
b5	OLSG2B	MTIOC4D Signal Active Level Setting	0: Active low 1: Active high	R/W*1
b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	OLSEN	Active Level Setting Enable	0: Disabled 1: Enabled	R/W*1
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

The ALR1 register specifies active levels for when detecting a simultaneous conduction for the MTU3 and MTU4 by the OCSR1 register.

OLSG0A Bit (MTIOC3B Signal Active Level Setting)

This bit sets the active level of the MTIOC3B output. Specifically, setting the OLSG0A bit to 0 sets the low level and to 1 sets the high level as the active level for detection of simultaneous conduction.

OLSG0B Bit (MTIOC3D Signal Active Level Setting)

This bit sets the active level of the MTIOC3D output. Specifically, setting the OLSG0B bit to 0 sets the low level and to 1 sets the high level as the active level for detection of simultaneous conduction.

OLSG1A Bit (MTIOC4A Signal Active Level Setting)

This bit sets the active level of the MTIOC4A output. Specifically, setting the OLSG1A bit to 0 sets the low level and to 1 sets the high level as the active level for detection of simultaneous conduction.

OLSG1B Bit (MTIOC4C Signal Active Level Setting)

This bit sets the active level of the MTIOC4C output. Specifically, setting the OLSG1B bit to 0 sets the low level and to 1 sets the high level as the active level for detection of simultaneous conduction.

OLSG2A Bit (MTIOC4B Signal Active Level Setting)

This bit sets the active level of the MTIOC4B output. Specifically, setting the OLSG2A bit to 0 sets the low level and to 1 sets the high level as the active level for detection of simultaneous conduction.

OLSG2B Bit (MTIOC4D Signal Active Level Setting)

This bit sets the active level of the MTIOC4D output. Specifically, setting the OLSG2B bit to 0 sets the low level and to 1 sets the high level as the active level for detection of simultaneous conduction.

OLSEN Bit (Active Level Setting Enable)

This bit determines enabling/disabling the active level setting by the OLSGnm bit (n = 0 to 2; m = A, B). When the OLSEN bit is 0, the OLSGnm bit setting becomes disabled, and the active level for the MTU3 and MTU4 outputs follows the settings for the MTU.TOCR1A and MTU.TOCR2A registers. When the OLSEN bit is 1, the active level for the MTU3 and MTU4 outputs follows the OLSGnm bit setting.

23.2.15 Active Level Setting Register 2 (ALR2)

Address(es): POE.ALR2 0009 E41Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	OLSEN	—	OLSG6B	OLSG6A	OLSG5B	OLSG5A	OLSG4B	OLSG4A
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OLSG4A	MTIOC6B Signal Active Level Setting	0: Active low 1: Active high	R/W*1
b1	OLSG4B	MTIOC6D Signal Active Level Setting	0: Active low 1: Active high	R/W*1
b2	OLSG5A	MTIOC7A Signal Active Level Setting	0: Active low 1: Active high	R/W*1
b3	OLSG5B	MTIOC7C Signal Active Level Setting	0: Active low 1: Active high	R/W*1
b4	OLSG6A	MTIOC7B Signal Active Level Setting	0: Active low 1: Active high	R/W*1
b5	OLSG6B	MTIOC7D Signal Active Level Setting	0: Active low 1: Active high	R/W*1
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	OLSEN	Active Level Setting Enable	0: Disabled 1: Enabled	R/W*1
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

The ALR2 register specifies active levels for when detecting a simultaneous conduction for the MTU6 and MTU7 by the OCSR2 register.

OLSG4A Bit (MTIOC6B Signal Active Level Setting)

This bit sets the active level of the MTIOC6B output. Specifically, setting the OLSG4A bit to 0 sets the low level and to 1 sets the high level as the active level for detection of simultaneous conduction.

OLSG4B Bit (MTIOC6D Signal Active Level Setting)

This bit sets the active level of the MTIOC6D output. Specifically, setting the OLSG4B bit to 0 sets the low level and to 1 sets the high level as the active level for detection of simultaneous conduction.

OLSG5A Bit (MTIOC7A Signal Active Level Setting)

This bit sets the active level of the MTIOC7A output. Specifically, setting the OLSG5A bit to 0 sets the low level and to 1 sets the high level as the active level for detection of simultaneous conduction.

OLSG5B Bit (MTIOC7C Signal Active Level Setting)

This bit sets the active level of the MTIOC7C output. Specifically, setting the OLSG5B bit to 0 sets the low level and to 1 sets the high level as the active level for detection of simultaneous conduction.

OLSG6A Bit (MTIOC7B Signal Active Level Setting)

This bit sets the active level of the MTIOC7B output. Specifically, setting the OLSG6A bit to 0 sets the low level and to 1 sets the high level as the active level for detection of simultaneous conduction.

OLSG6B Bit (MTIOC7D Signal Active Level Setting)

This bit sets the active level of the MTIOC7D output. Specifically, setting the OLSG6B bit to 0 sets the low level and to 1 sets the high level as the active level for detection of simultaneous conduction.

OLSEN Bit (Active Level Setting Enable)

This bit determines enabling/disabling the active level setting by the OLSGnm bit (n = 4 to 6; m = A, B). When the OLSEN bit is 0, the OLSGnm bit setting becomes disabled, and the active level for the MTU6 and MTU7 outputs follows the settings for the MTU.TOCR1B and MTU.TOCR2B registers. When the OLSEN bit is 1, the active level for the MTU6 and MTU7 outputs follows the OLSGnm bit setting.

23.2.16 Active Level Setting Register 3 (ALR3)

Address(es): POE.ALR3 0009 E42Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	OLSEN	—	OLSG2B	OLSG2A	OLSG1B	OLSG1A	OLSG0B	OLSG0A
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OLSG0A	GTIOC0A Signal Active Level Setting	0: Active low 1: Active high	R/W*1
b1	OLSG0B	GTIOC0B Signal Active Level Setting	0: Active low 1: Active high	R/W*1
b2	OLSG1A	GTIOC1A Signal Active Level Setting	0: Active low 1: Active high	R/W*1
b3	OLSG1B	GTIOC1B Signal Active Level Setting	0: Active low 1: Active high	R/W*1
b4	OLSG2A	GTIOC2A Signal Active Level Setting	0: Active low 1: Active high	R/W*1
b5	OLSG2B	GTIOC2B Signal Active Level Setting	0: Active low 1: Active high	R/W*1
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	OLSEN	Active Level Setting Enable	0: Disabled 1: Enabled	R/W*1
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

The ALR3 register specifies active levels for when detecting a simultaneous conduction for the GPTW0 to GPTW2 by the OCSR3 register.

OLSG0A Bit (GTIOC0A Signal Active Level Setting)

This bit sets the active level of the GTIOC0A output. Specifically, setting the OLSG0A bit to 0 sets the low level and to 1 sets the high level as the active level for detection of simultaneous conduction.

OLSG0B Bit (GTIOC0B Signal Active Level Setting)

This bit sets the active level of the GTIOC0B output. Specifically, setting the OLSG0B bit to 0 sets the low level and to 1 sets the high level as the active level for detection of simultaneous conduction.

OLSG1A Bit (GTIOC1A Signal Active Level Setting)

This bit sets the active level of the GTIOC1A output. Specifically, setting the OLSG1A bit to 0 sets the low level and to 1 sets the high level as the active level for detection of simultaneous conduction.

OLSG1B Bit (GTIOC1B Signal Active Level Setting)

This bit sets the active level of the GTIOC1B output. Specifically, setting the OLSG1B bit to 0 sets the low level and to 1 sets the high level as the active level for detection of simultaneous conduction.

OLSG2A Bit (GTIOC2A Signal Active Level Setting)

This bit sets the active level of the GTIOC2A output. Specifically, setting the OLSG2A bit to 0 sets the low level and to 1 sets the high level as the active level for detection of simultaneous conduction.

OLSG2B Bit (GTIOC2B Signal Active Level Setting)

This bit sets the active level of the GTIOC2B output. Specifically, setting the OLSG2B bit to 0 sets the low level and to 1 sets the high level as the active level for detection of simultaneous conduction.

OLSEN Bit (Active Level Setting Enable)

This bit determines enabling/disabling the active level setting by the OLSGnm bit (n = 0 to 2; m = A, B). When the OLSEN bit is 0, the OLSGnm bit setting becomes disabled.
The active level for the GPTW output can be specified only when the OLSEN bit is 1. When detecting a simultaneous conduction for the GPTW, set the OLSEN bit to 1, and specify the active level for the GPTW output by the OLSGnm bit.

23.2.17 Active Level Setting Register 4 (ALR4)

Address(es): POE.ALR4 0009 E44Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	OLSEN	—	OLSG2B	OLSG2A	OLSG1B	OLSG1A	OLSG0B	OLSG0A
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OLSG0A	GTIOC4A Signal Active Level Setting	0: Active low 1: Active high	R/W*1
b1	OLSG0B	GTIOC4B Signal Active Level Setting	0: Active low 1: Active high	R/W*1
b2	OLSG1A	GTIOC5A Signal Active Level Setting	0: Active low 1: Active high	R/W*1
b3	OLSG1B	GTIOC5B Signal Active Level Setting	0: Active low 1: Active high	R/W*1
b4	OLSG2A	GTIOC6A Signal Active Level Setting	0: Active low 1: Active high	R/W*1
b5	OLSG2B	GTIOC6B Signal Active Level Setting	0: Active low 1: Active high	R/W*1
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	OLSEN	Active Level Setting Enable	0: Disabled 1: Enabled	R/W*1
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

The ALR4 register specifies active levels for when detecting a simultaneous conduction for the GPTW4 to GPTW6 by the OCSR4 register.

OLSG0A Bit (GTIOC4A Signal Active Level Setting)

This bit sets the active level of the GTIOC4A output. Specifically, setting the OLSG0A bit to 0 sets the low level and to 1 sets the high level as the active level for detection of simultaneous conduction.

OLSG0B Bit (GTIOC4B Signal Active Level Setting)

This bit sets the active level of the GTIOC4B output. Specifically, setting the OLSG0B bit to 0 sets the low level and to 1 sets the high level as the active level for detection of simultaneous conduction.

OLSG1A Bit (GTIOC5A Signal Active Level Setting)

This bit sets the active level of the GTIOC5A output. Specifically, setting the OLSG1A bit to 0 sets the low level and to 1 sets the high level as the active level for detection of simultaneous conduction.

OLSG1B Bit (GTIOC5B Signal Active Level Setting)

This bit sets the active level of the GTIOC5B output. Specifically, setting the OLSG1B bit to 0 sets the low level and to 1 sets the high level as the active level for detection of simultaneous conduction.

OLSG2A Bit (GTIOC6A Signal Active Level Setting)

This bit sets the active level of the GTIOC6A output. Specifically, setting the OLSG2A bit to 0 sets the low level and to 1 sets the high level as the active level for detection of simultaneous conduction.

OLSG2B Bit (GTIOC6B Signal Active Level Setting)

This bit sets the active level of the GTIOC6B output. Specifically, setting the OLSG2B bit to 0 sets the low level and to 1 sets the high level as the active level for detection of simultaneous conduction.

OLSEN Bit (Active Level Setting Enable)

This bit determines enabling/disabling the active level setting by the OLSGnm bit (n = 0 to 2; m = A, B). When the OLSEN bit is 0, the OLSGnm bit setting becomes disabled.

The active level for the GPTW output can be specified only when the OLSEN bit is 1. When detecting a simultaneous conduction for the GPTW, set the OLSEN bit to 1, and specify the active level for the GPTW output by the OLSGnm bit.

23.2.18 Active Level Setting Register 5 (ALR5)

Address(es): POE.ALR5 0009 E44Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	OLSEN	—	—	—	—	—	OLSG0B	OLSG0A
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OLSG0A	GTIOC7A Signal Active Level Setting	0: Active low 1: Active high	R/W*1
b1	OLSG0B	GTIOC7B Signal Active Level Setting	0: Active low 1: Active high	R/W*1
b6 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	OLSEN	Active Level Setting Enable	0: Disabled 1: Enabled	R/W*1
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

The ALR5 register specifies active levels for when detecting a simultaneous conduction for the GPTW7 by the OCSR5 register.

OLSG0A Bit (GTIOC7A Signal Active Level Setting)

This bit sets the active level of the GTIOC7A output. Specifically, setting the OLSG0A bit to 0 sets the low level and to 1 sets the high level as the active level for detection of simultaneous conduction.

OLSG0B Bit (GTIOC7B Signal Active Level Setting)

This bit sets the active level of the GTIOC7B output. Specifically, setting the OLSG0B bit to 0 sets the low level and to 1 sets the high level as the active level for detection of simultaneous conduction.

OLSEN Bit (Active Level Setting Enable)

This bit determines enabling/disabling the active level setting by the OLSG0m bit (m = A, B). When the OLSEN bit is 0, the OLSG0m bit setting becomes disabled.

The active level for the GPTW output can be specified only when the OLSEN bit is 1. When detecting a simultaneous conduction for the GPTW, set the OLSEN bit to 1, and specify the active level for the GPTW output by the OLSG0m bit.

23.2.19 Software Port Output Enable Register (SPOER)

Address(es): POE.SPOER 0009 E42Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	GPT79 HIZ	GPT46 HIZ	GPT02 HIZ	—	MTUC H9HIZ	—	GPT23 HIZ	GPT01 HIZ	MTUC H0HIZ	MTUC H67HIZ	MTUC H34HIZ
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MTUCH34HIZ	MTU3 and MTU4 Pin Output Disable	0: Does not disable the outputs 1: Disable the outputs.	R/W
b1	MTUCH67HIZ	MTU6 and MTU7 Pin Output Disable	0: Does not disable the outputs 1: Disable the outputs.	R/W
b2	MTUCH0HIZ	MTU0 Pin Output Disable	0: Does not disable the outputs 1: Disable the outputs.	R/W
b3	GPT01HIZ	GPTW0 and GPTW1 Pin Output Disable	0: Does not disable the outputs 1: Disable the outputs.	R/W
b4	GPT23HIZ	GPTW2 and GPTW3 Pin Output Disable	0: Does not disable the outputs 1: Disable the outputs.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	MTUCH9HIZ	MTU9 Pin Output Disable	0: Does not disable the outputs 1: Disable the outputs.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b8	GPT02HIZ	GPTW0 to GPTW2 Pin Output Disable	0: Does not disable the outputs 1: Disable the outputs.	R/W
b9	GPT46HIZ	GPTW4 to GPTW6 Pin Output Disable	0: Does not disable the outputs 1: Disable the outputs.	R/W
b10	GPT79HIZ	GPTW7 Pin Output Disable	0: Does not disable the outputs 1: Disable the outputs.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SPOER register is used to disable the outputs of the corresponding pins.

MTUCH34HIZ Bit (MTU3 and MTU4 Pin Output Disable)

This bit specifies whether to disable the outputs of the MTU complementary PWM output pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, MTIOC4D). To set this bit to 0, confirm that the bit is 1 and then write 0 to the bit.

MTUCH67HIZ Bit (MTU6 and MTU7 Pin Output Disable)

This bit specifies whether to disable the outputs of the MTU complementary PWM output pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7B, MTIOC7C, MTIOC7D). To set this bit to 0, confirm that the bit is 1 and then write 0 to the bit.

MTUCH0HIZ Bit (MTU0 Pin Output Disable)

This bit specifies whether to disable the outputs of the MTU0 pins. To set this bit to 0, confirm that the bit is 1 and then write 0 to the bit.

GPT01HIZ Bit (GPTW0 and GPTW1 Pin Output Disable)

This bit specifies whether to disable the outputs of the GPTW0 and GPTW1 pins (GTIOC0A, GTIOC0B, GTIOC1A, GTIOC1B). To set this bit to 0, confirm that the bit is 1 and then write 0 to the bit.

GPT23HIZ Bit (GPTW2 and GPTW3 Pin Output Disable)

This bit specifies whether to disable the outputs of the GPTW2 and GPTW3 pins (GTIOC2A, GTIOC2B, GTIOC3A, GTIOC3B). To set this bit to 0, confirm that the bit is 1 and then write 0 to the bit.

MTUCH9HIZ Bit (MTU9 Pin Output Disable)

This bit specifies whether to disable the outputs of the MTU9 pins. To set this bit to 0, confirm that the bit is 1 and then write 0 to the bit.

GPT02HIZ Bit (GPTW0 to GPTW2 Pin Output Disable)

This bit specifies whether to disable the outputs of the GPTW0 to GPTW2 pins (GTIOC0A, GTIOC0B, GTIOC1A, GTIOC1B, GTIOC2A, GTIOC2B). To set this bit to 0, confirm that the bit is 1 and then write 0 to the bit.

GPT46HIZ Bit (GPTW4 to GPTW6 Pin Output Disable)

This bit specifies whether to disable the outputs of the GPTW4 to GPTW6 pins (GTIOC4A, GTIOC4B, GTIOC5A, GTIOC5B, GTIOC6A, GTIOC6B). To set this bit to 0, confirm that the bit is 1 and then write 0 to the bit.

GPT79HIZ Bit (GPTW7 Pin Output Disable)

This bit specifies whether to disable the outputs of the GPTW7 pins (GTIOC7A, GTIOC7B). To set this bit to 0, confirm that the bit is 1 and then write 0 to the bit.

23.2.20 Port Output Enable Control Register 1 (POECR1)

Address(es): POE.POECR1 0009 E40Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	MTU0DZE	MTU0CZE	MTU0BZE	MTU0AZE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MTU0AZE	MTIOC0A Pin High-Impedance Enable	0: Does not switch the pin to high-impedance state. 1: Switch the pin to high-impedance state.	R/W*1
b1	MTU0BZE	MTIOC0B Pin High-Impedance Enable	0: Does not switch the pin to high-impedance state. 1: Switch the pin to high-impedance state.	R/W*1
b2	MTU0CZE	MTIOC0C Pin High-Impedance Enable	0: Does not switch the pin to high-impedance state. 1: Switch the pin to high-impedance state.	R/W*1
b3	MTU0DZE	MTIOC0D Pin High-Impedance Enable	0: Does not switch the pin to high-impedance state. 1: Switch the pin to high-impedance state.	R/W*1
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

The POECR1 register controls high-impedance state of the MTU0 pins.

MTU0AZE Bit (MTIOC0A Pin High-Impedance Enable)

This bit specifies whether to switch the MTIOC0A output to the high-impedance state when any of the SPOER.MTUCH0HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag (n = 1 to 5, 7, 8; m = 0, 4, 8, 10, 11, 12, 9) additionally selected in the POECR5 register, or the POECMPFR.CjFLAG flag (j = 0 to 5), is set to 1.

Set the PMMCR0.MTU0AME bit to 0 when setting this bit to 1.

MTU0BZE Bit (MTIOC0B Pin High-Impedance Enable)

This bit specifies whether to switch the MTIOC0B output to the high-impedance state when any of the SPOER.MTUCH0HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag additionally selected in the POECR5 register, or the POECMPFR.CjFLAG flag, is set to 1.

Set the PMMCR0.MTU0BME bit to 0 when setting this bit to 1.

MTU0CZE Bit (MTIOC0C Pin High-Impedance Enable)

This bit specifies whether to switch the MTIOC0C output to the high-impedance state when any of the SPOER.MTUCH0HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag additionally selected in the POECR5 register, or the POECMPFR.CjFLAG flag, is set to 1.

Set the PMMCR0.MTU0CME bit to 0 when setting this bit to 1.

MTU0DZE Bit (MTIOC0D Pin High-Impedance Enable)

This bit specifies whether to switch the MTIOC0D output to the high-impedance state when any of the SPOER.MTUCH0HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag additionally selected in the POECR5 register, or the POECMPFR.CjFLAG flag, is set to 1.

Set the PMMCR0.MTU0DME bit to 0 when setting this bit to 1.

23.2.21 Port Output Enable Control Register 2 (POECR2)

Address(es): POE.POECR2 0009 E40Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	MTU3BDZE	MTU4ACZE	MTU4BDZE	—	—	—	—	—	MTU6BDZE	MTU7ACZE	MTU7BDZE
Value after reset:	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	MTU7BDZE	MTIOC7B/MTIOC7D Pin High-Impedance Enable*2	0: Does not switch the pin to high-impedance state. 1: Switch the pin to high-impedance state.	R/W*1
b1	MTU7ACZE	MTIOC7A/MTIOC7C Pin High-Impedance Enable*2	0: Does not switch the pin to high-impedance state. 1: Switch the pin to high-impedance state.	R/W*1
b2	MTU6BDZE	MTIOC6B/MTIOC6D Pin High-Impedance Enable*2	0: Does not switch the pin to high-impedance state. 1: Switch the pin to high-impedance state.	R/W*1
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	MTU4BDZE	MTIOC4B/MTIOC4D Pin High-Impedance Enable	0: Does not switch the pin to high-impedance state. 1: Switch the pin to high-impedance state.	R/W*1
b9	MTU4ACZE	MTIOC4A/MTIOC4C Pin High-Impedance Enable	0: Does not switch the pin to high-impedance state. 1: Switch the pin to high-impedance state.	R/W*1
b10	MTU3BDZE	MTIOC3B/MTIOC3D Pin High-Impedance Enable	0: Does not switch the pin to high-impedance state. 1: Switch the pin to high-impedance state.	R/W*1
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.
Note 2. Set this bit to 0 when MTU6 or MTU7 is not used.

The POECR2 register controls high-impedance state of the MTU complementary PWM output pins (MTU3, and MTU4, MTU6, and MTU7 pins).

MTU7BDZE Bit (MTIOC7B/MTIOC7D Pin High-Impedance Enable)

This bit specifies whether to switch the MTIOC7B output and MTIOC7D output to the high-impedance state when at least one of the OCSR2.OSF2 flag (when the OCSR2.OCE2 bit is 1), the SPOER.MTUCH67HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag (n = 1 to 5, 7, 8; m = 0, 4, 8, 10, 11, 12, 9) additionally selected in the POECR4B register, or the POECMPFR.CjFLAG flag (j = 0 to 5) is set to 1.
Set the PMMCR1.MTU7BME and PMMCR1.MTU7DME bits to 0 when setting this bit to 1.

MTU7ACZE Bit (MTIOC7A/MTIOC7C Pin High-Impedance Enable)

This bit specifies whether to switch the MTIOC7A output and MTIOC7C output to the high-impedance state when at least one of the OCSR2.OSF2 flag (when the OCSR2.OCE2 bit is 1), the SPOER.MTUCH67HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag additionally selected in the POECR4B register, or the POECMPFR.CjFLAG flag is set to 1.
Set the PMMCR1.MTU7AME and PMMCR1.MTU7CME bits to 0 when setting this bit to 1.

MTU6BDZE Bit (MTIOC6B/MTIOC6D Pin High-Impedance Enable)

This bit specifies whether to switch the MTIOC6B output and MTIOC6D output to the high-impedance state when at least one of the OCSR2.OSF2 flag (when the OCSR2.OCE2 bit is 1), the SPOER.MTUCH67HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag additionally selected in the POECR4B register, or the POECMPFR.CjFLAG flag is set to 1.
Set the PMMCR1.MTU6BME and PMMCR1.MTU6DME bits to 0 when setting this bit to 1.

MTU4BDZE Bit (MTIOC4B/MTIOC4D Pin High-Impedance Enable)

This bit specifies whether to switch the MTIOC4B output and MTIOC4D output to the high-impedance state when at least one of the OCSR1.OSF1 flag (when the OCSR1.OCE1 bit is 1), the SPOER.MTUCH34HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag additionally selected in the POECR4 register, or the POECMPFR.CjFLAG flag is set to 1.
Set the PMMCR1.MTU4BME and PMMCR1.MTU4DME bits to 0 when setting this bit to 1.

MTU4ACZE Bit (MTIOC4A/MTIOC4C Pin High-Impedance Enable)

This bit specifies whether to switch the MTIOC4A output and MTIOC4C output to the high-impedance state when at least one of the OCSR1.OSF1 flag (when the OCSR1.OCE1 bit is 1), the SPOER.MTUCH34HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag additionally selected in the POECR4 register, or the POECMPFR.CjFLAG flag is set to 1.
Set the PMMCR1.MTU4AME and PMMCR1.MTU4CME bits to 0 when setting this bit to 1.

MTU3BDZE Bit (MTIOC3B/MTIOC3D Pin High-Impedance Enable)

This bit specifies whether to switch the MTIOC3B output and MTIOC3D output to the high-impedance state when at least one of the OCSR1.OSF1 flag (when the OCSR1.OCE1 bit is 1), the SPOER.MTUCH34HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag additionally selected in the POECR4 register, or the POECMPFR.CjFLAG flag is set to 1.
Set the PMMCR1.MTU3BME and PMMCR1.MTU3DME bits to 0 when setting this bit to 1.

23.2.22 Port Output Enable Control Register 3 (POECR3)

Address(es): POE.POECR3 0009 E40Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	GPT7ABZE	GPT6ABZE	GPT5ABZE	GPT4ABZE	GPT3ABZE	GPT2ABZE	GPT1ABZE	GPT0ABZE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	GPT0ABZE	GTIOC0A/GTIOC0B Pin High-Impedance Enable	0: Does not switch the pin to high-impedance state. 1: Switch the pin to high-impedance state.	R/W*1
b1	GPT1ABZE	GTIOC1A/GTIOC1B Pin High-Impedance Enable	0: Does not switch the pin to high-impedance state. 1: Switch the pin to high-impedance state.	R/W*1
b2	GPT2ABZE	GTIOC2A/GTIOC2B Pin High-Impedance Enable	0: Does not switch the pin to high-impedance state. 1: Switch the pin to high-impedance state.	R/W*1
b3	GPT3ABZE	GTIOC3A/GTIOC3B Pin High-Impedance Enable	0: Does not switch the pin to high-impedance state. 1: Switch the pin to high-impedance state.	R/W*1
b4	GPT4ABZE	GTIOC4A/GTIOC4B Pin High-Impedance Enable	0: Does not switch the pin to high-impedance state. 1: Switch the pin to high-impedance state.	R/W*1
b5	GPT5ABZE	GTIOC5A/GTIOC5B Pin High-Impedance Enable	0: Does not switch the pin to high-impedance state. 1: Switch the pin to high-impedance state.	R/W*1
b6	GPT6ABZE	GTIOC6A/GTIOC6B Pin High-Impedance Enable	0: Does not switch the pin to high-impedance state. 1: Switch the pin to high-impedance state.	R/W*1
b7	GPT7ABZE	GTIOC7A/GTIOC7B Pin High-Impedance Enable	0: Does not switch the pin to high-impedance state. 1: Switch the pin to high-impedance state.	R/W*1
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

The POECR3 register controls high-impedance state of the GPTW pins (GPTW0 to GPTW7).

GPT0ABZE Bit (GTIOC0A/GTIOC0B Pin High-Impedance Enable)

This bit determines whether to switch the GTIOC0A and GTIOC0B outputs to the high-impedance state when at least one of the followings is set to 1: the OCSR3.OSF3 flag (when the OCSR3.OCE3 bit is 1), the SPOER.GPT01HIZ bit, the SPOER.GPT02HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit = 1), the ICSRn.POEmF flag (n = 1 to 5, 7, 8; m = 0, 4, 8, 10, 11, 12, 9) additionally selected in the POECR6 register, POECMPFR.CjFLAG flag (j = 0 to 5), the ICSRn.POEmF flag additionally selected in the POECR9 register, and the POECMPFR.CjFLAG flag.
Set the PMMCR2.GPT0AME and PMMCR2.GPT0BME bits to 0 when setting this bit to 1.

GPT1ABZE Bit (GTIOC1A/GTIOC1B Pin High-Impedance Enable)

This bit determines whether to switch the GTIOC1A and GTIOC1B outputs to the high-impedance state when at least one of the followings is set to 1: the OCSR3.OSF3 flag (when the OCSR3.OCE3 bit is 1), the SPOER.GPT01HIZ bit, the SPOER.GPT02HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit = 1), the ICSRn.POEmF flag additionally selected in the POECR6 register, POECMPFR.CjFLAG flag, the ICSRn.POEmF flag additionally selected in the POECR9 register, and the POECMPFR.CjFLAG flag.
Set the PMMCR2.GPT1AME and PMMCR2.GPT1BME bits to 0 when setting this bit to 1.

GPT2ABZE Bit (GTIOC2A/GTIOC2B Pin High-Impedance Enable)

This bit determines whether to switch the GTIOC2A and GTIOC2B outputs to the high-impedance state when at least one of the followings is set to 1: the OCSR3.OSF3 flag (when the OCSR3.OCE3 bit is 1), the SPOER.GPT23HIZ bit, the SPOER.GPT02HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit = 1), the ICSRn.POEmF flag

additionally selected in the POECR6B register, POECMPFR.CjFLAG flag, the ICSRn.POEmF flag additionally selected in the POECR9 register, and the POECMPFR.CjFLAG flag.
Set the PMMCR2.GPT2AME and PMMCR2.GPT2BME bits to 0 when setting this bit to 1.

GPT3ABZE Bit (GTIOC3A/GTIOC3B Pin High-Impedance Enable)

This bit specifies whether to switch the GTIOC3A output and GTIOC3B output to the high-impedance state when at least one of the SPOER.GPT23HIZ bit, ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag additionally selected in the POECR6B register, or POECMPFR.CjFLAG flag is set to 1.
Set the PMMCR2.GPT3AME and PMMCR2.GPT3BME bits to 0 when setting this bit to 1.

GPT4ABZE Bit (GTIOC4A/GTIOC4B Pin High-Impedance Enable)

This bit specifies whether to switch the GTIOC4A output and GTIOC4B output to the high-impedance state when at least one of the OCSR4.OSF4 flag (when the OCSR4.OCE4 bit is 1), the SPOER.GPT46HIZ bit, ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag additionally selected in the POECR10 register, or POECMPFR.CjFLAG flag is set to 1.
Set the PMMCR2.GPT4AME and PMMCR2.GPT4BME bits to 0 when setting this bit to 1.

GPT5ABZE Bit (GTIOC5A/GTIOC5B Pin High-Impedance Enable)

This bit specifies whether to switch the GTIOC5A output and GTIOC5B output to the high-impedance state when at least one of the OCSR4.OSF4 flag (when the OCSR4.OCE4 bit is 1), the SPOER.GPT46HIZ bit, ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag additionally selected in the POECR10 register, or POECMPFR.CjFLAG flag is set to 1.
Set the PMMCR2.GPT5AME and PMMCR2.GPT5BME bits to 0 when setting this bit to 1.

GPT6ABZE Bit (GTIOC6A/GTIOC6B Pin High-Impedance Enable)

This bit specifies whether to switch the GTIOC6A output and GTIOC6B output to the high-impedance state when at least one of the OCSR4.OSF4 flag (when the OCSR4.OCE4 bit is 1), the SPOER.GPT46HIZ bit, ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag additionally selected in the POECR10 register, or POECMPFR.CjFLAG flag is set to 1.
Set the PMMCR2.GPT6AME and PMMCR2.GPT6BME bits to 0 when setting this bit to 1.

GPT7ABZE Bit (GTIOC7A/GTIOC7B Pin High-Impedance Enable)

This bit specifies whether to switch the GTIOC7A output and GTIOC7B output to the high-impedance state when at least one of the OCSR5.OSF5 flag (when the OCSR5.OCE5 bit is 1), the SPOER.GPT79HIZ bit, ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag additionally selected in the POECR11 register, or POECMPFR.CjFLAG flag is set to 1.
Set the PMMCR2.GPT7AME and PMMCR2.GPT7BME bits to 0 when setting this bit to 1.

23.2.23 Port Output Enable Control Register 4 (POECR4)

Address(es): POE.POECR4 0009 E410h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	IC8ADDMT34ZE	—	IC6ADDMT34ZE	IC5ADDMT34ZE	IC4ADDMT34ZE	IC3ADDMT34ZE	IC2ADDMT34ZE	IC1ADDMT34ZE	CMADDMT34ZE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMADDMT34ZE	MTU3 and MTU4 Output Disabling Condition CFLAG Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b1	IC1ADDMT34ZE	MTU3 and MTU4 Output Disabling Condition POE0F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b2	IC2ADDMT34ZE	MTU3 and MTU4 Output Disabling Condition POE4F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b3	IC3ADDMT34ZE	MTU3 and MTU4 Output Disabling Condition POE8F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b4	IC4ADDMT34ZE	MTU3 and MTU4 Output Disabling Condition POE10F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b5	IC5ADDMT34ZE	MTU3 and MTU4 Output Disabling Condition POE11F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b6	IC6ADDMT34ZE	MTU3 and MTU4 Output Disabling Condition POE12F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b8	IC8ADDMT34ZE	MTU3 and MTU4 Output Disabling Condition POE9F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

The POECR4 register is used to set the control conditions to disable the output of the MTU complementary PWM output pins (MTU3 and MTU4 pins).

CMADDMT34ZE Bit (MTU3 and MTU4 Output Disabling Condition CFLAG Add)

Adds the POECMPFR.CjFLAG flag (j = 0 to 5) to the output disabling control conditions for the MTU3 and MTU4 pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D).
However, when the pins are disabled by the flag, an OEIk interrupt (k = 1 to 5) will not be generated.

IC1ADDMT34ZE Bit (MTU3 and MTU4 Output Disabling Condition POE0F Add)

Adds the ICSR1.POE0F flag to the output disabling control conditions for the MTU3 and MTU4 pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D).

IC2ADDMT34ZE Bit (MTU3 and MTU4 Output Disabling Condition POE4F Add)

Adds the ICSR2.POE4F flag to the output disabling control conditions for the MTU3 and MTU4 pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D).

IC3ADDMT34ZE Bit (MTU3 and MTU4 Output Disabling Condition POE8F Add)

Adds the ICSR3.POE8F flag to the output disabling control conditions for the MTU3 and MTU4 pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D).

IC4ADDMT34ZE Bit (MTU3 and MTU4 Output Disabling Condition POE10F Add)

Adds the ICSR4.POE10F flag to the output disabling control conditions for the MTU3 and MTU4 pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D).

IC5ADDMT34ZE Bit (MTU3 and MTU4 Output Disabling Condition POE11F Add)

Adds the ICSR5.POE11F flag to the output disabling control conditions for the MTU3 and MTU4 pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D).

IC6ADDMT34ZE Bit (MTU3 and MTU4 Output Disabling Condition POE12F Add)

Adds the ICSR7.POE12F flag to the output disabling control conditions for the MTU3 and MTU4 pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D).

IC8ADDMT34ZE Bit (MTU3 and MTU4 Output Disabling Condition POE9F Add)

Adds the ICSR8.POE9F flag to the output disabling control conditions for the MTU3 and MTU4 pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D).

23.2.24 Port Output Enable Control Register 4B (POECR4B)

Address(es): POE.POECR4B 0009 E44Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	IC8ADDMT67ZE	—	IC6ADDMT67ZE	IC5ADDMT67ZE	IC4ADDMT67ZE	IC3ADDMT67ZE	IC2ADDMT67ZE	IC1ADDMT67ZE	CMADDMT67ZE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMADDMT67ZE	MTU6 and MTU7 Output Disabling Condition CFLAG Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b1	IC1ADDMT67ZE	MTU6 and MTU7 Output Disabling Condition POE0F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b2	IC2ADDMT67ZE	MTU6 and MTU7 Output Disabling Condition POE4F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b3	IC3ADDMT67ZE	MTU6 and MTU7 Output Disabling Condition POE8F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b4	IC4ADDMT67ZE	MTU6 and MTU7 Output Disabling Condition POE10F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b5	IC5ADDMT67ZE	MTU6 and MTU7 Output Disabling Condition POE11F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b6	IC6ADDMT67ZE	MTU6 and MTU7 Output Disabling Condition POE12F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b8	IC8ADDMT67ZE	MTU6 and MTU7 Output Disabling Condition POE9F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

The POECR4B register is used to set the control conditions to disable the output of the MTU complementary PWM output pins (MTU6 and MTU7 pins).

CMADDMT67ZE Bit (MTU6 and MTU7 Output Disabling Condition CFLAG Add)

Adds the POECMPFR.CjFLAG flag (j = 0 to 5) to the output disabling control conditions for the MTU6 and MTU7 pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, MTIOC7D).
However, when the pins are disabled by the flag, an OEIk interrupt (k = 1 to 5) will not be generated.

IC1ADDMT67ZE Bit (MTU6 and MTU7 Output Disabling Condition POE0F Add)

Adds the ICSR1.POE0F flag to the output disabling control conditions for the MTU6 and MTU7 pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, MTIOC7D).

IC2ADDMT67ZE Bit (MTU6 and MTU7 Output Disabling Condition POE4F Add)

Adds the ICSR2.POE4F flag to the output disabling control conditions for the MTU6 and MTU7 pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, MTIOC7D).

IC3ADDMT67ZE Bit (MTU6 and MTU7 Output Disabling Condition POE8F Add)

Adds the ICSR3.POE8F flag to the output disabling control conditions for the MTU6 and MTU7 pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, MTIOC7D).

IC4ADDMT67ZE Bit (MTU6 and MTU7 Output Disabling Condition POE10F Add)

Adds the ICSR4.POE10F flag to the output disabling control conditions for the MTU6 and MTU7 pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, MTIOC7D).

IC5ADDMT67ZE Bit (MTU6 and MTU7 Output Disabling Condition POE11F Add)

Adds the ICSR5.POE11F flag to the output disabling control conditions for the MTU6 and MTU7 pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, MTIOC7D).

IC6ADDMT67ZE Bit (MTU6 and MTU7 Output Disabling Condition POE12F Add)

Adds the ICSR7.POE12F flag to the output disabling control conditions for the MTU6 and MTU7 pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, MTIOC7D).

IC8ADDMT67ZE Bit (MTU6 and MTU7 Output Disabling Condition POE9F Add)

Adds the ICSR8.POE9F flag to the output disabling control conditions for the MTU6 and MTU7 pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, MTIOC7D).

23.2.25 Port Output Enable Control Register 5 (POECR5)

Address(es): POE.POECR5 0009 E412h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	IC8ADDMT0ZE	—	IC6ADDMT0ZE	IC5ADDMT0ZE	IC4ADDMT0ZE	IC3ADDMT0ZE	IC2ADDMT0ZE	IC1ADDMT0ZE	CMADDMT0ZE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMADDMT0ZE	MTU0 Output Disabling Condition CFLAG Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b1	IC1ADDMT0ZE	MTU0 Output Disabling Condition POE0F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b2	IC2ADDMT0ZE	MTU0 Output Disabling Condition POE4F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b3	IC3ADDMT0ZE	MTU0 Output Disabling Condition POE8F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b4	IC4ADDMT0ZE	MTU0 Output Disabling Condition POE10F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b5	IC5ADDMT0ZE	MTU0 Output Disabling Condition POE11F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b6	IC6ADDMT0ZE	MTU0 Output Disabling Condition POE12F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b8	IC8ADDMT0ZE	MTU0 Output Disabling Condition POE9F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

The POECR5 register is used to set the control conditions to disable the output of the MTU0 pins.

CMADDMT0ZE Bit (MTU0 Output Disabling Condition CFLAG Add)

Adds the POECMPFR.CjFLAG flag (j = 0 to 5) to the output disabling control conditions for the MTU0 pins (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D).
However, when the pins are disabled by the flag, an OEIk interrupt (k = 1 to 5) will not be generated.

IC1ADDMT0ZE Bit (MTU0 Output Disabling Condition POE0F Add)

Adds the ICSR1.POE0F flag to the output disabling control conditions for the MTU0 pins (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D).

IC2ADDMT0ZE Bit (MTU0 Output Disabling Condition POE4F Add)

Adds the ICSR2.POE4F flag to the output disabling control conditions for the MTU0 pins (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D).

IC3ADDMT0ZE Bit (MTU0 Output Disabling Condition POE8F Add)

Adds the ICSR3.POE8F flag to the output disabling control conditions for the MTU0 pins (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D).

IC4ADDMT0ZE Bit (MTU0 Output Disabling Condition POE10F Add)

Adds the ICSR4.POE10F flag to the output disabling control conditions for the MTU0 pins (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D).

IC5ADDMT0ZE Bit (MTU0 Output Disabling Condition POE11F Add)

Adds the ICSR5.POE11F flag to the output disabling control conditions for the MTU0 pins (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D).

IC6ADDMT0ZE Bit (MTU0 Output Disabling Condition POE12F Add)

Adds the ICSR7.POE12F flag to the output disabling control conditions for the MTU0 pins (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D).

IC8ADDMT0ZE Bit (MTU0 Output Disabling Condition POE9F Add)

Adds the ICSR8.POE9F flag to the output disabling control conditions for the MTU0 pins (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D).

23.2.26 Port Output Enable Control Register 6 (POECR6)

Address(es): POE.POECR6 0009 E414h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	IC8ADDGPT01ZE	—	IC6ADDGPT01ZE	IC5ADDGPT01ZE	IC4ADDGPT01ZE	IC3ADDGPT01ZE	IC2ADDGPT01ZE	IC1ADDGPT01ZE	CMADDGPT01ZE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMADDGPT01ZE	GPTW0 and GPTW1 Output Disabling Condition CFLAG Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b1	IC1ADDGPT01ZE	GPTW0 and GPTW1 Output Disabling Condition POE0F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b2	IC2ADDGPT01ZE	GPTW0 and GPTW1 Output Disabling Condition POE4F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b3	IC3ADDGPT01ZE	GPTW0 and GPTW1 Output Disabling Condition POE8F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b4	IC4ADDGPT01ZE	GPTW0 and GPTW1 Output Disabling Condition POE10F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b5	IC5ADDGPT01ZE	GPTW0 and GPTW1 Output Disabling Condition POE11F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b6	IC6ADDGPT01ZE	GPTW0 and GPTW1 Output Disabling Condition POE12F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b8	IC8ADDGPT01ZE	GPTW0 and GPTW1 Output Disabling Condition POE9F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

The POECR6 register is used to set the control conditions to disable the output of the GPTW0 and GPTW1 pins.

CMADDGPT01ZE Bit (GPTW0 and GPTW1 Output Disabling Condition CFLAG Add)

Adds the POECMPFR.CjFLAG flag (j = 0 to 5) to the output disabling control conditions for the GPTW0 and GPTW1 pins (GTIOC0A, GTIOC0B, GTIOC1A, GTIOC1B).
However, when the pins are disabled by the flag, an OEIk interrupt (k = 1 to 5) will not be generated.

IC1ADDGPT01ZE Bit (GPTW0 and GPTW1 Output Disabling Condition POE0F Add)

Adds the ICSR1.POE0F flag to the output disabling control conditions for the GPTW0 and GPTW1 pins (GTIOC0A, GTIOC0B, GTIOC1A, GTIOC1B).

IC2ADDGPT01ZE Bit (GPTW0 and GPTW1 Output Disabling Condition POE4F Add)

Adds the ICSR2.POE4F flag to the output disabling control conditions for the GPTW0 and GPTW1 pins (GTIOC0A, GTIOC0B, GTIOC1A, GTIOC1B).

IC3ADDGPT01ZE Bit (GPTW0 and GPTW1 Output Disabling Condition POE8F Add)

Adds the ICSR3.POE8F flag to the output disabling control conditions for the GPTW0 and GPTW1 pins (GTIOC0A, GTIOC0B, GTIOC1A, GTIOC1B).

IC4ADDGPT01ZE Bit (GPTW0 and GPTW1 Output Disabling Condition POE10F Add)

Adds the ICSR4.POE10F flag to the output disabling control conditions for the GPTW0 and GPTW1 pins (GTIOC0A, GTIOC0B, GTIOC1A, GTIOC1B).

IC5ADDGPT01ZE Bit (GPTW0 and GPTW1 Output Disabling Condition POE11F Add)

Adds the ICSR5.POE11F flag to the output disabling control conditions for the GPTW0 and GPTW1 pins (GTIOC0A, GTIOC0B, GTIOC1A, GTIOC1B).

IC6ADDGPT01ZE Bit (GPTW0 and GPTW1 Output Disabling Condition POE12F Add)

Adds the ICSR7.POE12F flag to the output disabling control conditions for the GPTW0 and GPTW1 pins (GTIOC0A, GTIOC0B, GTIOC1A, GTIOC1B).

IC8ADDGPT01ZE Bit (GPTW0 and GPTW1 Output Disabling Condition POE9F Add)

Adds the ICSR8.POE9F flag to the output disabling control conditions for the GPTW0 and GPTW1 pins (GTIOC0A, GTIOC0B, GTIOC1A, GTIOC1B).

23.2.27 Port Output Enable Control Register 6B (POECR6B)

Address(es): POE.POECR6B 0009 E450h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	IC8ADDGPT23ZE	—	IC6ADDGPT23ZE	IC5ADDGPT23ZE	IC4ADDGPT23ZE	IC3ADDGPT23ZE	IC2ADDGPT23ZE	IC1ADDGPT23ZE	CMADDGPT23ZE
Value after reset:	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMADDGPT23ZE	GPTW2 and GPTW3 Output Disabling Condition CFLAG Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b1	IC1ADDGPT23ZE	GPTW2 and GPTW3 Output Disabling Condition POE0F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b2	IC2ADDGPT23ZE	GPTW2 and GPTW3 Output Disabling Condition POE4F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b3	IC3ADDGPT23ZE	GPTW2 and GPTW3 Output Disabling Condition POE8F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b4	IC4ADDGPT23ZE	GPTW2 and GPTW3 Output Disabling Condition POE10F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b5	IC5ADDGPT23ZE	GPTW2 and GPTW3 Output Disabling Condition POE11F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b6	IC6ADDGPT23ZE	GPTW2 and GPTW3 Output Disabling Condition POE12F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W*1
b8	IC8ADDGPT23ZE	GPTW2 and GPTW3 Output Disabling Condition POE9F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

The POECR6B register is used to set the control conditions to disable the output of the GPTW2 and GPTW3 pins.

CMADDGPT23ZE Bit (GPTW2 and GPTW3 Output Disabling Condition CFLAG Add)

Adds the POECMPFR.CjFLAG flag (j = 0 to 5) to the output disabling control conditions for the GPTW2 and GPTW3 pins (GTIOC2A, GTIOC2B, GTIOC3A, GTIOC3B).
However, when the pins are disabled by the flag, an OEIk interrupt (k = 1 to 5) will not be generated.

IC1ADDGPT23ZE Bit (GPTW2 and GPTW3 Output Disabling Condition POE0F Add)

Adds the ICSR1.POE0F flag to the output disabling control conditions for the GPTW2 and GPTW3 pins (GTIOC2A, GTIOC2B, GTIOC3A, GTIOC3B).

IC2ADDGPT23ZE Bit (GPTW2 and GPTW3 Output Disabling Condition POE4F Add)

Adds the ICSR2.POE4F flag to the output disabling control conditions for the GPTW2 and GPTW3 pins (GTIOC2A, GTIOC2B, GTIOC3A, GTIOC3B).

IC3ADDGPT23ZE Bit (GPTW2 and GPTW3 Output Disabling Condition POE8F Add)

Adds the ICSR3.POE8F flag to the output disabling control conditions for the GPTW2 and GPTW3 pins (GTIOC2A, GTIOC2B, GTIOC3A, GTIOC3B).

IC4ADDGPT23ZE Bit (GPTW2 and GPTW3 Output Disabling Condition POE10F Add)

Adds the ICSR4.POE10F flag to the output disabling control conditions for the GPTW2 and GPTW3 pins (GTIOC2A, GTIOC2B, GTIOC3A, GTIOC3B).

IC5ADDGPT23ZE Bit (GPTW2 and GPTW3 Output Disabling Condition POE11F Add)

Adds the ICSR5.POE11F flag to the output disabling control conditions for the GPTW2 and GPTW3 pins (GTIOC2A, GTIOC2B, GTIOC3A, GTIOC3B).

IC6ADDGPT23ZE Bit (GPTW2 and GPTW3 Output Disabling Condition POE12F Add)

Adds the ICSR7.POE12F flag to the output disabling control conditions for the GPTW2 and GPTW3 pins (GTIOC2A, GTIOC2B, GTIOC3A, GTIOC3B).

IC8ADDGPT23ZE Bit (GPTW2 and GPTW3 Output Disabling Condition POE9F Add)

Adds the ICSR8.POE9F flag to the output disabling control conditions for the GPTW2 and GPTW3 pins (GTIOC2A, GTIOC2B, GTIOC3A, GTIOC3B).

23.2.28 Port Output Enable Control Register 7 (POECR7)

Address(es): POE.POECR7 0009 E422h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	MTU9DZE	MTU9CZE	MTU9BZE	MTU9AZE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MTU9AZE	MTIOC9A Pin High-Impedance Enable	0: Does not switch the pin to high-impedance state. 1: Switch the pin to high-impedance state.	R/W*1
b1	MTU9BZE	MTIOC9B Pin High-Impedance Enable	0: Does not switch the pin to high-impedance state. 1: Switch the pin to high-impedance state.	R/W*1
b2	MTU9CZE	MTIOC9C Pin High-Impedance Enable	0: Does not switch the pin to high-impedance state. 1: Switch the pin to high-impedance state.	R/W*1
b3	MTU9DZE	MTIOC9D Pin High-Impedance Enable	0: Does not switch the pin to high-impedance state. 1: Switch the pin to high-impedance state.	R/W*1
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

The POECR7 register controls high-impedance state of the MTU9 pins.

MTU9AZE Bit (MTIOC9A Pin High-Impedance Enable)

This bit specifies whether to switch the MTIOC9A output for the MTU9 pin to high-impedance state when any of the SPOER.MTUCH9HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag (n = 1 to 5, 7, 8; m = 0, 4, 8, 10, 11, 12, 9) additionally selected in the POECR8 register, or the POECMPFR.CjFLAG flag (j = 0 to 5), is set to 1.
Set the PMMCR0.MTU9AME bit to 0 when setting this bit to 1.

MTU9BZE Bit (MTIOC9B Pin High-Impedance Enable)

This bit specifies whether to switch the MTIOC9B output for the MTU9 pin to high-impedance state when any of the SPOER.MTUCH9HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag additionally selected in the POECR8 register, or the POECMPFR.CjFLAG flag, is set to 1.
Set the PMMCR0.MTU9BME bit to 0 when setting this bit to 1.

MTU9CZE Bit (MTIOC9C Pin High-Impedance Enable)

This bit specifies whether to switch the MTIOC9C output for the MTU9 pin to high-impedance state when any of the SPOER.MTUCH9HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag additionally selected in the POECR8 register, or the POECMPFR.CjFLAG flag, is set to 1.
Set the PMMCR0.MTU9CME bit to 0 when setting this bit to 1.

MTU9DZE Bit (MTIOC9D Pin High-Impedance Enable)

This bit specifies whether to switch the MTIOC9D output for the MTU9 pin to high-impedance state when any of the SPOER.MTUCH9HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag additionally selected in the POECR8 register, or POECMPFR.CjFLAG flag, is set to 1.
Set the PMMCR0.MTU9DME bit to 0 when setting this bit to 1.

23.2.29 Port Output Enable Control Register 8 (POECR8)

Address(es): POE.POECR8 0009 E424h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	IC8ADDMT9ZE	—	IC6ADDMT9ZE	IC5ADDMT9ZE	IC4ADDMT9ZE	IC3ADDMT9ZE	IC2ADDMT9ZE	IC1ADDMT9ZE	CMADDMT9ZE
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMADDMT9ZE	MTU9 Output Disabling Condition CFLAG Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b1	IC1ADDMT9ZE	MTU9 Output Disabling Condition POE0F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b2	IC2ADDMT9ZE	MTU9 Output Disabling Condition POE4F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b3	IC3ADDMT9ZE	MTU9 Output Disabling Condition POE8F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b4	IC4ADDMT9ZE	MTU9 Output Disabling Condition POE10F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b5	IC5ADDMT9ZE	MTU9 Output Disabling Condition POE11F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b6	IC6ADDMT9ZE	MTU9 Output Disabling Condition POE12F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b8	IC8ADDMT9ZE	MTU9 Output Disabling Condition POE9F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

The POECR8 register is used to set the control conditions to disable the output of the MTU9 pins.

CMADDMT9ZE Bit (MTU9 Output Disabling Condition CFLAG Add)

Adds the POECMPFR.CjFLAG flag (j = 0 to 5) to the output disabling control conditions for the MTU9 pins (MTIOC9A, MTIOC9B, MTIOC9C, MTIOC9D).
However, when the pins are disabled by the flag, an OEIk interrupt (k = 1 to 5) will not be generated.

IC1ADDMT9ZE Bit (MTU9 Output Disabling Condition POE0F Add)

Adds the ICSR1.POE0F flag to the output disabling control conditions for the MTU9 pins (MTIOC9A, MTIOC9B, MTIOC9C, MTIOC9D).

IC2ADDMT9ZE Bit (MTU9 Output Disabling Condition POE4F Add)

Adds the ICSR2.POE4F flag to the output disabling control conditions for the MTU9 pins (MTIOC9A, MTIOC9B, MTIOC9C, MTIOC9D).

IC3ADDMT9ZE Bit (MTU9 Output Disabling Condition POE8F Add)

Adds the ICSR3.POE8F flag to the output disabling control conditions for the MTU9 pins (MTIOC9A, MTIOC9B, MTIOC9C, MTIOC9D).

IC4ADDMT9ZE Bit (MTU9 Output Disabling Condition POE10F Add)

Adds the ICSR4.POE10F flag to the output disabling control conditions for the MTU9 pins (MTIOC9A, MTIOC9B, MTIOC9C, MTIOC9D).

IC5ADDMT9ZE Bit (MTU9 Output Disabling Condition POE11F Add)

Adds the ICSR5.POE11F flag to the output disabling control conditions for the MTU9 pins (MTIOC9A, MTIOC9B, MTIOC9C, MTIOC9D).

IC6ADDMT9ZE Bit (MTU9 Output Disabling Condition POE12F Add)

Adds the ICSR7.POE12F flag to the output disabling control conditions for the MTU9 pins (MTIOC9A, MTIOC9B, MTIOC9C, MTIOC9D).

IC8ADDMT9ZE Bit (MTU9 Output Disabling Condition POE9F Add)

Adds the ICSR8.POE9F flag to the output disabling control conditions for the MTU9 pins (MTIOC9A, MTIOC9B, MTIOC9C, MTIOC9D).

23.2.30 Port Output Enable Control Register 9 (POECR9)

Address(es): POE.POECR9 0009 E452h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	IC8ADDGPT02ZE	—	IC6ADDGPT02ZE	IC5ADDGPT02ZE	IC4ADDGPT02ZE	IC3ADDGPT02ZE	IC2ADDGPT02ZE	IC1ADDGPT02ZE	CMADDGPT02ZE
Value after reset:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMADDGPT02ZE	GPTW0 to GPTW2 Output Disabling Condition CFLAG Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b1	IC1ADDGPT02ZE	GPTW0 to GPTW2 Output Disabling Condition POE0F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b2	IC2ADDGPT02ZE	GPTW0 to GPTW2 Output Disabling Condition POE4F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b3	IC3ADDGPT02ZE	GPTW0 to GPTW2 Output Disabling Condition POE8F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b4	IC4ADDGPT02ZE	GPTW0 to GPTW2 Output Disabling Condition POE10F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b5	IC5ADDGPT02ZE	GPTW0 to GPTW2 Output Disabling Condition POE11F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b6	IC6ADDGPT02ZE	GPTW0 to GPTW2 Output Disabling Condition POE12F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b8	IC8ADDGPT02ZE	GPTW0 to GPTW2 Output Disabling Condition POE9F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

The POECR9 register is used to set the control conditions to disable the output of the GPTW0 to GPTW2 pins.

CMADDGPT02ZE Bit (GPTW0 to GPTW2 Output Disabling Condition CFLAG Add)

Adds the POECMPFR.CjFLAG flag (j = 0 to 5) to the output disabling control conditions for the GPTW0 to GPTW2 pins (GTIOC0A, GTIOC0B, GTIOC1A, GTIOC1B, GTIOC2A, GTIOC2B).
However, when the pins are disabled by the flag, an OEIk interrupt (k = 1 to 5) will not be generated.

IC1ADDGPT02ZE Bit (GPTW0 to GPTW2 Output Disabling Condition POE0F Add)

Adds the ICSR1.POE0F flag to the output disabling control conditions for the GPTW0 to GPTW2 pins (GTIOC0A, GTIOC0B, GTIOC1A, GTIOC1B, GTIOC2A, GTIOC2B).

IC2ADDGPT02ZE Bit (GPTW0 to GPTW2 Output Disabling Condition POE4F Add)

Adds theICSR2.POE4F flag to the output disabling control conditions for the GPTW0 to GPTW2 pins (GTIOC0A, GTIOC0B, GTIOC1A, GTIOC1B, GTIOC2A, GTIOC2B).

IC3ADDGPT02ZE Bit (GPTW0 to GPTW2 Output Disabling Condition POE8F Add)

Adds the ICSR3.POE8F flag to the output disabling control conditions for the GPTW0 to GPTW2 pins (GTIOC0A, GTIOC0B, GTIOC1A, GTIOC1B, GTIOC2A, GTIOC2B).

IC4ADDGPT02ZE Bit (GPTW0 to GPTW2 Output Disabling Condition POE10F Add)

Adds the ICSR4.POE10F flag to the output disabling control conditions for the GPTW0 to GPTW2 pins (GTIOC0A, GTIOC0B, GTIOC1A, GTIOC1B, GTIOC2A, GTIOC2B).

IC5ADDGPT02ZE Bit (GPTW0 to GPTW2 Output Disabling Condition POE11F Add)

Adds the ICSR5.POE11F flag to the output disabling control conditions for the GPTW0 to GPTW2 pins (GTIOC0A, GTIOC0B, GTIOC1A, GTIOC1B, GTIOC2A, GTIOC2B).

IC6ADDGPT02ZE Bit (GPTW0 to GPTW2 Output Disabling Condition POE12F Add)

Adds the ICSR7.POE12F flag to the output disabling control conditions for the GPTW0 to GPTW2 pins (GTIOC0A, GTIOC0B, GTIOC1A, GTIOC1B, GTIOC2A, GTIOC2B).

IC8ADDGPT02ZE Bit (GPTW0 to GPTW2 Output Disabling Condition POE9F Add)

Adds the ICSR8.POE9F flag to the output disabling control conditions for the GPTW0 to GPTW2 pins (GTIOC0A, GTIOC0B, GTIOC1A, GTIOC1B, GTIOC2A, GTIOC2B).

23.2.31 Port Output Enable Control Register 10 (POECR10)

Address(es): POE.POECR10 0009 E454h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	IC8ADDGPT46ZE	—	IC6ADDGPT46ZE	IC5ADDGPT46ZE	IC4ADDGPT46ZE	IC3ADDGPT46ZE	IC2ADDGPT46ZE	IC1ADDGPT46ZE	CMADDGPT46ZE
Value after reset:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMADDGPT46ZE	GPTW4 to GPTW6 Output Disabling Condition CFLAG Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b1	IC1ADDGPT46ZE	GPTW4 to GPTW6 Output Disabling Condition POE0F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b2	IC2ADDGPT46ZE	GPTW4 to GPTW6 Output Disabling Condition POE4F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b3	IC3ADDGPT46ZE	GPTW4 to GPTW6 Output Disabling Condition POE8F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b4	IC4ADDGPT46ZE	GPTW4 to GPTW6 Output Disabling Condition POE10F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b5	IC5ADDGPT46ZE	GPTW4 to GPTW6 Output Disabling Condition POE11F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b6	IC6ADDGPT46ZE	GPTW4 to GPTW6 Output Disabling Condition POE12F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b8	IC8ADDGPT46ZE	GPTW4 to GPTW6 Output Disabling Condition POE9F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

The POECR10 register is used to set the control conditions to disable the output of the GPTW4 to GPTW6 pins.

CMADDGPT46ZE Bit (GPTW4 to GPTW6 Output Disabling Condition CFLAG Add)

Adds the POECMPFR.CjFLAG flag (j = 0 to 5) to the output disabling control conditions for the GPTW4 to GPTW6 pins (GTIOC4A, GTIOC4B, GTIOC5A, GTIOC5B, GTIOC6A, GTIOC6B).
However, when the pins are disabled by the flag, an OEIk interrupt (k = 1 to 5) will not be generated.

IC1ADDGPT46ZE Bit (GPTW4 to GPTW6 Output Disabling Condition POE0F Add)

Adds the ICSR1.POE0F flag to the output disabling control conditions for the GPTW4 to GPTW6 pins (GTIOC4A, GTIOC4B, GTIOC5A, GTIOC5B, GTIOC6A, GTIOC6B).

IC2ADDGPT46ZE Bit (GPTW4 to GPTW6 Output Disabling Condition POE4F Add)

Adds the ICSR2.POE4F flag to the output disabling control conditions for the GPTW4 to GPTW6 pins (GTIOC4A, GTIOC4B, GTIOC5A, GTIOC5B, GTIOC6A, GTIOC6B).

IC3ADDGPT46ZE Bit (GPTW4 to GPTW6 Output Disabling Condition POE8F Add)

Adds the ICSR3.POE8F flag to the output disabling control conditions for the GPTW4 to GPTW6 pins (GTIOC4A, GTIOC4B, GTIOC5A, GTIOC5B, GTIOC6A, GTIOC6B).

IC4ADDGPT46ZE Bit (GPTW4 to GPTW6 Output Disabling Condition POE10F Add)

Adds the ICSR4.POE10F flag to the output disabling control conditions for the GPTW4 to GPTW6 pins (GTIOC4A, GTIOC4B, GTIOC5A, GTIOC5B, GTIOC6A, GTIOC6B).

IC5ADDGPT46ZE Bit (GPTW4 to GPTW6 Output Disabling Condition POE11F Add)

Adds the ICSR5.POE11F flag to the output disabling control conditions for the GPTW4 to GPTW6 pins (GTIOC4A, GTIOC4B, GTIOC5A, GTIOC5B, GTIOC6A, GTIOC6B).

IC6ADDGPT46ZE Bit (GPTW4 to GPTW6 Output Disabling Condition POE12F Add)

Adds the ICSR7.POE12F flag to the output disabling control conditions for the GPTW4 to GPTW6 pins (GTIOC4A, GTIOC4B, GTIOC5A, GTIOC5B, GTIOC6A, GTIOC6B).

IC8ADDGPT46ZE Bit (GPTW4 to GPTW6 Output Disabling Condition POE9F Add)

Adds the ICSR8.POE9F flag to the output disabling control conditions for the GPTW4 to GPTW6 pins (GTIOC4A, GTIOC4B, GTIOC5A, GTIOC5B, GTIOC6A, GTIOC6B).

23.2.32 Port Output Enable Control Register 11 (POECR11)

Address(es): POE.POECR11 0009 E456h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	IC8ADDGPT79ZE	—	IC6ADDGPT79ZE	IC5ADDGPT79ZE	IC4ADDGPT79ZE	IC3ADDGPT79ZE	IC2ADDGPT79ZE	IC1ADDGPT79ZE	CMADDGPT79ZE
Value after reset:	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMADDGPT79ZE	GPTW7 Output Disabling Condition CFLAG Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b1	IC1ADDGPT79ZE	GPTW7 Output Disabling Condition POE0F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b2	IC2ADDGPT79ZE	GPTW7 Output Disabling Condition POE4F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b3	IC3ADDGPT79ZE	GPTW7 Output Disabling Condition POE8F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b4	IC4ADDGPT79ZE	GPTW7 Output Disabling Condition POE10F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b5	IC5ADDGPT79ZE	GPTW7 Output Disabling Condition POE11F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b6	IC6ADDGPT79ZE	GPTW7 Output Disabling Condition POE12F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b8	IC8ADDGPT79ZE	GPTW7 Output Disabling Condition POE9F Add	0: Does not add the flags to the conditions to disable the output. 1: Adds the flags to the conditions to disable the output.	R/W*1
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

The POECR11 register is used to set the control conditions to disable the output of the GPTW7 pins.

CMADDGPT79ZE Bit (GPTW7 Output Disabling Condition CFLAG Add)

Adds the POECMPFR.CjFLAG flag (j = 0 to 5) to the output disabling control conditions for the GPTW7 pins (GTIOC7A, GTIOC7B).

However, when the pins are disabled by the flag, an OEIk interrupt (k = 1 to 5) will not be generated.

IC1ADDGPT79ZE Bit (GPTW7 Output Disabling Condition POE0F Add)

Adds the ICSR1.POE0F flag to the output disabling control conditions for the GPTW7 pins (GTIOC7A, GTIOC7B).

IC2ADDGPT79ZE Bit (GPTW7 Output Disabling Condition POE4F Add)

Adds the ICSR2.POE4F flag to the output disabling control conditions for the GPTW7 pins (GTIOC7A, GTIOC7B).

IC3ADDGPT79ZE Bit (GPTW7 Output Disabling Condition POE8F Add)

Adds the ICSR3.POE8F flag to the output disabling control conditions for the GPTW7 pins (GTIOC7A, GTIOC7B).

IC4ADDGPT79ZE Bit (GPTW7 Output Disabling Condition POE10F Add)

Adds the ICSR4.POE10F flag to the output disabling control conditions for the GPTW7 pins (GTIOC7A, GTIOC7B).

IC5ADDGPT79ZE Bit (GPTW7 Output Disabling Condition POE11F Add)

Adds the ICSR5.POE11F flag to the output disabling control conditions for the GPTW7 pins (GTIOC7A, GTIOC7B).

IC6ADDGPT79ZE Bit (GPTW7 Output Disabling Condition POE12F Add)

Adds the ICSR7.POE12F flag to the output disabling control conditions for the GPTW7 pins (GTIOC7A, GTIOC7B).

IC8ADDGPT79ZE Bit (GPTW7 Output Disabling Condition POE9F Add)

Adds the ICSR8.POE9F flag to the output disabling control conditions for the GPTW7 pins (GTIOC7A, GTIOC7B).

23.2.33 Port Mode Mask Control Register 0 (PMMCR0)

Address(es): POE.PMMCR0 0009 E430h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	MTU9DME	MTU9CME	MTU9BME	MTU9AME	—	—	—	—	MTU0DME	MTU0CME	MTU0BME	MTU0AME
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MTU0AME	MTIOC0A Pin Port Mode Mask Enable	0: Switching to the general I/O port pin is disabled. 1: Switching to the general I/O port pin is enabled.	R/W*1
b1	MTU0BME	MTIOC0B Pin Port Mode Mask Enable	0: Switching to the general I/O port pin is disabled. 1: Switching to the general I/O port pin is enabled.	R/W*1
b2	MTU0CME	MTIOC0C Pin Port Mode Mask Enable	0: Switching to the general I/O port pin is disabled. 1: Switching to the general I/O port pin is enabled.	R/W*1
b3	MTU0DME	MTIOC0D Pin Port Mode Mask Enable	0: Switching to the general I/O port pin is disabled. 1: Switching to the general I/O port pin is enabled.	R/W*1
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	MTU9AME	MTIOC9A Pin Port Mode Mask Enable	0: Switching to the general I/O port pin is disabled. 1: Switching to the general I/O port pin is enabled.	R/W*1
b9	MTU9BME	MTIOC9B Pin Port Mode Mask Enable	0: Switching to the general I/O port pin is disabled. 1: Switching to the general I/O port pin is enabled.	R/W*1
b10	MTU9CME	MTIOC9C Pin Port Mode Mask Enable	0: Switching to the general I/O port pin is disabled. 1: Switching to the general I/O port pin is enabled.	R/W*1
b11	MTU9DME	MTIOC9D Pin Port Mode Mask Enable	0: Switching to the general I/O port pin is disabled. 1: Switching to the general I/O port pin is enabled.	R/W*1
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

The PMMCR0 register is used to mask the settings of the PMR register related to the MTU0 and MTU9 pins.

MTU0AME Bit (MTIOC0A Pin Port Mode Mask Enable)

This bit specifies whether switching the MTIOC0A pin to the general I/O port when one of the following flags and bit is set to 1: the SPOER.MTUCH0HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag (n = 1 to 5, 7, 8; m = 0, 4, 8, 10, 11, 12, 9) additionally selected in the POECR5 register, and the POECMPFR.CjFLAG flag (j = 0 to 5).

When setting this bit to 1, set the POECR1.MTU0AZE bit to 0. When the POECR1.MTU0AZE bit is 1, the setting of the bit is ignored.

MTU0BME Bit (MTIOC0B Pin Port Mode Mask Enable)

This bit specifies whether switching the MTIOC0B pin to the general I/O port when one of the following flags and bit is set to 1: the SPOER.MTUCH0HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag additionally selected in the POECR5 register, and the POECMPFR.CjFLAG flag.

When setting this bit to 1, set the POECR1.MTU0BZE bit to 0. When the POECR1.MTU0BZE bit is 1, the setting of the bit is ignored.

MTU0CME Bit (MTIOC0C Pin Port Mode Mask Enable)

This bit specifies whether switching the MTIOC0C pin to the general I/O port when one of the following flags and bit is set to 1: the SPOER.MTUCH0HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag additionally selected in the POECR5 register, and the POECMPFR.CjFLAG flag.

When setting this bit to 1, set the POECR1.MTU0CZE bit to 0. When the POECR1.MTU0CZE bit is 1, the setting of the bit is ignored.

MTU0DME Bit (MTIOC0D Pin Port Mode Mask Enable)

This bit specifies whether switching the MTIOC0D pin to the general I/O port when one of the following flags and bit is set to 1: the SPOER.MTUCH0HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag additionally selected in the POECR5 register, and the POECMPFR.CjFLAG flag.
When setting this bit to 1, set the POECR1.MTU0DZE bit to 0. When the POECR1.MTU0DZE bit is 1, the setting of the bit is ignored.

MTU9AME Bit (MTIOC9A Pin Port Mode Mask Enable)

This bit specifies whether switching the MTIOC9A pin to the general I/O port when one of the following flags and bit is set to 1: the SPOER.MTUCH9HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag additionally selected in the POECR8 register, and the POECMPFR.CjFLAG flag.
When setting this bit to 1, set the POECR7.MTU9AZE bit to 0. When the POECR7.MTU9AZE bit is 1, the setting of the bit is ignored.

MTU9BME Bit (MTIOC9B Pin Port Mode Mask Enable)

This bit specifies whether switching the MTIOC9B pin to the general I/O port when one of the following flags and bit is set to 1: the SPOER.MTUCH9HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag additionally selected in the POECR8 register, and the POECMPFR.CjFLAG flag.
When setting this bit to 1, set the POECR7.MTU9BZE bit to 0. When the POECR7.MTU9BZE bit is 1, the setting of the bit is ignored.

MTU9CME Bit (MTIOC9C Pin Port Mode Mask Enable)

This bit specifies whether switching the MTIOC9C pin to the general I/O port when one of the following flags and bit is set to 1: the SPOER.MTUCH9HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag additionally selected in the POECR8 register, and the POECMPFR.CjFLAG flag.
When setting this bit to 1, set the POECR7.MTU9CZE bit to 0. When the POECR7.MTU9CZE bit is 1, the setting of the bit is ignored.

MTU9DME Bit (MTIOC9D Pin Port Mode Mask Enable)

This bit specifies whether switching the MTIOC9D pin to the general I/O port when one of the following flags and bit is set to 1: the SPOER.MTUCH9HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag additionally selected in the POECR8 register, and the POECMPFR.CjFLAG flag.
When setting this bit to 1, set the POECR7.MTU9DZE bit to 0. When the POECR7.MTU9DZE bit is 1, the setting of the bit is ignored.

23.2.34 Port Mode Mask Control Register 1 (PMMCR1)

Address(es): POE.PMMCR1 0009 E432h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	MTU3DME	MTU4CME	MTU4DME	MTU3BME	MTU4AME	MTU4BME	—	—	MTU6DME	MTU7CME	MTU7DME	MTU6BME	MTU7AME	MTU7BME
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MTU7BME	MTIOC7B Pin Port Mode Mask Enable	0: Switching to the general I/O port pin is disabled. 1: Switching to the general I/O port pin is enabled.	R/W*1
b1	MTU7AME	MTIOC7A Pin Port Mode Mask Enable	0: Switching to the general I/O port pin is disabled. 1: Switching to the general I/O port pin is enabled.	R/W*1
b2	MTU6BME	MTIOC6B Pin Port Mode Mask Enable	0: Switching to the general I/O port pin is disabled. 1: Switching to the general I/O port pin is enabled.	R/W*1
b3	MTU7DME	MTIOC7D Pin Port Mode Mask Enable	0: Switching to the general I/O port pin is disabled. 1: Switching to the general I/O port pin is enabled.	R/W*1
b4	MTU7CME	MTIOC7C Pin Port Mode Mask Enable	0: Switching to the general I/O port pin is disabled. 1: Switching to the general I/O port pin is enabled.	R/W*1
b5	MTU6DME	MTIOC6D Pin Port Mode Mask Enable	0: Switching to the general I/O port pin is disabled. 1: Switching to the general I/O port pin is enabled.	R/W*1
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	MTU4BME	MTIOC4B Pin Port Mode Mask Enable	0: Switching to the general I/O port pin is disabled. 1: Switching to the general I/O port pin is enabled.	R/W*1
b9	MTU4AME	MTIOC4A Pin Port Mode Mask Enable	0: Switching to the general I/O port pin is disabled. 1: Switching to the general I/O port pin is enabled.	R/W*1
b10	MTU3BME	MTIOC3B Pin Port Mode Mask Enable	0: Switching to the general I/O port pin is disabled. 1: Switching to the general I/O port pin is enabled.	R/W*1
b11	MTU4DME	MTIOC4D Pin Port Mode Mask Enable	0: Switching to the general I/O port pin is disabled. 1: Switching to the general I/O port pin is enabled.	R/W*1
b12	MTU4CME	MTIOC4C Pin Port Mode Mask Enable	0: Switching to the general I/O port pin is disabled. 1: Switching to the general I/O port pin is enabled.	R/W*1
b13	MTU3DME	MTIOC3D Pin Port Mode Mask Enable	0: Switching to the general I/O port pin is disabled. 1: Switching to the general I/O port pin is enabled.	R/W*1
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

The PMMCR1 register is used to mask the settings of the PMR register related to the MTU complementary PWM output pins (MTU3, MTU4, MTU6, MTU7).

MTU7BME Bit (MTIOC7B Pin Port Mode Mask Enable)

This bit specifies whether switching the MTIOC7B pin to the general I/O port when one of the following flags and bit is set to 1: the OCSR2.OSF2 flag (when the OCSR2.OCE2 bit is 1), the SPOER.MTUCH67HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag (n = 1 to 5, 7, 8; m = 0, 4, 8, 10, 11, 12, 9) additionally selected in the POECR4B register, and the POECMPFR.CjFLAG flag (j = 0 to 5).
When setting this bit to 1, set the POECR2.MTU7BDZE bit to 0. When the POECR2.MTU7BDZE bit is 1, the setting of the bit is ignored.

MTU7AME Bit (MTIOC7A Pin Port Mode Mask Enable)

This bit specifies whether switching the MTIOC7A pin to the general I/O port when one of the following flags and bit is set to 1: the OCSR2.OSF2 flag (when the OCSR2.OCE2 bit is 1), the SPOER.MTUCH67HIZ bit, the ICSR6.OSTSTF

flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag additionally selected in the POECR4B register, and the POECMPFR.CjFLAG flag.
When setting this bit to 1, set the POECR2.MTU7ACZE bit to 0. When the POECR2.MTU7ACZE bit is 1, the setting of the bit is ignored.

MTU6BME Bit (MTIOC6B Pin Port Mode Mask Enable)

This bit specifies whether switching the MTIOC6B pin to the general I/O port when one of the following flags and bit is set to 1: the OCSR2.OSF2 flag (when the OCSR2.OCE2 bit is 1), the SPOER.MTUCH67HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag additionally selected in the POECR4B register, and the POECMPFR.CjFLAG flag.
When setting this bit to 1, set the POECR2.MTU6BDZE bit to 0. When the POECR2.MTU6BDZE bit is 1, the setting of the bit is ignored.

MTU7DME Bit (MTIOC7D Pin Port Mode Mask Enable)

This bit specifies whether switching the MTIOC7D pin to the general I/O port when one of the following flags and bit is set to 1: the OCSR2.OSF2 flag (when the OCSR2.OCE2 bit is 1), the SPOER.MTUCH67HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag additionally selected in the POECR4B register, and the POECMPFR.CjFLAG flag.
When setting this bit to 1, set the POECR2.MTU7BDZE bit to 0. When the POECR2.MTU7BDZE bit is 1, the setting of the bit is ignored.

MTU7CME Bit (MTIOC7C Pin Port Mode Mask Enable)

This bit specifies whether switching the MTIOC7C pin to the general I/O port when one of the following flags and bit is set to 1: the OCSR2.OSF2 flag (when the OCSR2.OCE2 bit is 1), the SPOER.MTUCH67HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag additionally selected in the POECR4B register, and the POECMPFR.CjFLAG flag.
When setting this bit to 1, set the POECR2.MTU7ACZE bit to 0. When the POECR2.MTU7ACZE bit is 1, the setting of the bit is ignored.

MTU6DME Bit (MTIOC6D Pin Port Mode Mask Enable)

This bit specifies whether switching the MTIOC6D pin to the general I/O port when one of the following flags and bit is set to 1: the OCSR2.OSF2 flag (when the OCSR2.OCE2 bit is 1), the SPOER.MTUCH67HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag additionally selected in the POECR4B register, and the POECMPFR.CjFLAG flag.
When setting this bit to 1, set the POECR2.MTU6BDZE bit to 0. When the POECR2.MTU6BDZE bit is 1, the setting of the bit is ignored.

MTU4BME Bit (MTIOC4B Pin Port Mode Mask Enable)

This bit specifies whether switching the MTIOC4B pin to the general I/O port when one of the following flags and bit is set to 1: the OCSR1.OSF1 flag (when the OCSR1.OCE1 bit is 1), the SPOER.MTUCH34HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag additionally selected in the POECR4 register, and the POECMPFR.CjFLAG flag.
When setting this bit to 1, set the POECR2.MTU4BDZE bit to 0. When the POECR2.MTU4BDZE bit is 1, the setting of the bit is ignored.

MTU4AME Bit (MTIOC4A Pin Port Mode Mask Enable)

This bit specifies whether switching the MTIOC4A pin to the general I/O port when one of the following flags and bit is set to 1: the OCSR1.OSF1 flag (when the OCSR1.OCE1 bit is 1), the SPOER.MTUCH34HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag additionally selected in the POECR4 register, and the

POECMPFR.CjFLAG flag.
When setting this bit to 1, set the POECR2.MTU4ACZE bit to 0. When the POECR2.MTU4ACZE bit is 1, the setting of the bit is ignored.

MTU3BME Bit (MTIOC3B Pin Port Mode Mask Enable)

This bit specifies whether switching the MTIOC3B pin to the general I/O port when one of the following flags and bit is set to 1: the OCSR1.OSF1 flag (when the OCSR1.OCE1 bit is 1), the SPOER.MTUCH34HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag additionally selected in the POECR4 register, and the POECMPFR.CjFLAG flag.
When setting this bit to 1, set the POECR2.MTU3BDZE bit to 0. When the POECR2.MTU3BDZE bit is 1, the setting of the bit is ignored.

MTU4DME Bit (MTIOC4D Pin Port Mode Mask Enable)

This bit specifies whether switching the MTIOC4D pin to the general I/O port when one of the following flags and bit is set to 1: the OCSR1.OSF1 flag (when the OCSR1.OCE1 bit is 1), the SPOER.MTUCH34HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag additionally selected in the POECR4 register, and the POECMPFR.CjFLAG flag.
When setting this bit to 1, set the POECR2.MTU4BDZE bit to 0. When the POECR2.MTU4BDZE bit is 1, the setting of the bit is ignored.

MTU4CME Bit (MTIOC4C Pin Port Mode Mask Enable)

This bit specifies whether switching the MTIOC4C pin to the general I/O port when one of the following flags and bit is set to 1: the OCSR1.OSF1 flag (when the OCSR1.OCE1 bit is 1), the SPOER.MTUCH34HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag additionally selected in the POECR4 register, and the POECMPFR.CjFLAG flag.
When setting this bit to 1, set the POECR2.MTU4ACZE bit to 0. When the POECR2.MTU4ACZE bit is 1, the setting of the bit is ignored.

MTU3DME Bit (MTIOC3D Pin Port Mode Mask Enable)

This bit specifies whether switching the MTIOC3D pin to the general I/O port when one of the following flags and bit is set to 1: the OCSR1.OSF1 flag (when the OCSR1.OCE1 bit is 1), the SPOER.MTUCH34HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag additionally selected in the POECR4 register, and the POECMPFR.CjFLAG flag.
When setting this bit to 1, set the POECR2.MTU3BDZE bit to 0. When the POECR2.MTU3BDZE bit is 1, the setting of the bit is ignored.

23.2.35 Port Mode Mask Control Register 2 (PMMCR2)

Address(es): POE.PMMCR2 0009 E434h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	GPT7BME	GPT7AME	GPT6BME	GPT6AME	GPT5BME	GPT5AME	GPT4BME	GPT4AME	GPT3BME	GPT3AME	GPT2BME	GPT2AME	GPT1BME	GPT1AME	GPT0BME	GPT0AME
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	GPT0AME	GTIOC0A Pin Port Mode Mask Enable	0: Switching to the general I/O port pin is disabled. 1: Switching to the general I/O port pin is enabled.	R/W*1
b1	GPT0BME	GTIOC0B Pin Port Mode Mask Enable	0: Switching to the general I/O port pin is disabled. 1: Switching to the general I/O port pin is enabled.	R/W*1
b2	GPT1AME	GTIOC1A Pin Port Mode Mask Enable	0: Switching to the general I/O port pin is disabled. 1: Switching to the general I/O port pin is enabled.	R/W*1
b3	GPT1BME	GTIOC1B Pin Port Mode Mask Enable	0: Switching to the general I/O port pin is disabled. 1: Switching to the general I/O port pin is enabled.	R/W*1
b4	GPT2AME	GTIOC2A Pin Port Mode Mask Enable	0: Switching to the general I/O port pin is disabled. 1: Switching to the general I/O port pin is enabled.	R/W*1
b5	GPT2BME	GTIOC2B Pin Port Mode Mask Enable	0: Switching to the general I/O port pin is disabled. 1: Switching to the general I/O port pin is enabled.	R/W*1
b6	GPT3AME	GTIOC3A Pin Port Mode Mask Enable	0: Switching to the general I/O port pin is disabled. 1: Switching to the general I/O port pin is enabled.	R/W*1
b7	GPT3BME	GTIOC3B Pin Port Mode Mask Enable	0: Switching to the general I/O port pin is disabled. 1: Switching to the general I/O port pin is enabled.	R/W*1
b8	GPT4AME	GTIOC4A Pin Port Mode Mask Enable	0: Switching to the general I/O port pin is disabled. 1: Switching to the general I/O port pin is enabled.	R/W*1
b9	GPT4BME	GTIOC4B Pin Port Mode Mask Enable	0: Switching to the general I/O port pin is disabled. 1: Switching to the general I/O port pin is enabled.	R/W*1
b10	GPT5AME	GTIOC5A Pin Port Mode Mask Enable	0: Switching to the general I/O port pin is disabled. 1: Switching to the general I/O port pin is enabled.	R/W*1
b11	GPT5BME	GTIOC5B Pin Port Mode Mask Enable	0: Switching to the general I/O port pin is disabled. 1: Switching to the general I/O port pin is enabled.	R/W*1
b12	GPT6AME	GTIOC6A Pin Port Mode Mask Enable	0: Switching to the general I/O port pin is disabled. 1: Switching to the general I/O port pin is enabled.	R/W*1
b13	GPT6BME	GTIOC6B Pin Port Mode Mask Enable	0: Switching to the general I/O port pin is disabled. 1: Switching to the general I/O port pin is enabled.	R/W*1
b14	GPT7AME	GTIOC7A Pin Port Mode Mask Enable	0: Switching to the general I/O port pin is disabled. 1: Switching to the general I/O port pin is enabled.	R/W*1
b15	GPT7BME	GTIOC7B Pin Port Mode Mask Enable	0: Switching to the general I/O port pin is disabled. 1: Switching to the general I/O port pin is enabled.	R/W*1

Note 1. Can be modified only once after a reset.

The PMMCR2 register is used to mask the settings of the PMR register related to the GPTW output pins (GPTW0 to GPTW7).

GPT0AME Bit (GTIOC0A Pin Port Mode Mask Enable)

This bit determines whether to switch the GTIOC0A pin to the general I/O port when at least one of the followings is set to 1: the OCSR3.OSF3 flag (when the OCSR3.OCE3 bit =1), the SPOER.GPT01HIZ bit, the SPOER.GPT02HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit = 1), the ICSRn.POEmF flag (n = 1 to 5, 7, 8; m = 0, 4, 8, 10, 11, 12, 9) additionally selected in the POECR6 register, POECMPFR.CjFLAG flag (j = 0 to 5), the ICSRn.POEmF flag additionally selected in the POECR9 register, and the POECMPFR.CjFLAG flag.

To set this bit to 1, set the POECR3.GPT0ABZE bit to 0. If the POECR3.GPT0ABZE bit is 1, setting of this bit is ignored.

GPT0BME Bit (GTIOC0B Pin Port Mode Mask Enable)

This bit determines whether to switch the GTIOC0B pin to the general I/O port when at least one of the followings is set to 1: the OCSR3.OSF3 flag (when the OCSR3.OCE3 bit =1), the SPOER.GPT01HIZ bit, the SPOER.GPT02HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit = 1), the ICSRn.POEmF flag additionally selected in the POECR6 register, POECMPFR.CjFLAG flag, the ICSRn.POEmF flag additionally selected in the POECR9 register, and the POECMPFR.CjFLAG flag.
To make this bit 1, set the POECR3.GPT0ABZE bit to 0. If the POECR3.GPT0ABZE bit is 1, setting of this bit is ignored.

GPT1AME Bit (GTIOC1A Pin Port Mode Mask Enable)

This bit determines whether to switch the GTIOC1A pin to the general I/O port when at least one of the followings is set to 1: the OCSR3.OSF3 flag (when the OCSR3.OCE3 bit =1), the SPOER.GPT01HIZ bit, the SPOER.GPT02HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit = 1), the ICSRn.POEmF flag additionally selected in the POECR6 register, POECMPFR.CjFLAG flag, the ICSRn.POEmF flag additionally selected in the POECR9 register, and the POECMPFR.CjFLAG flag.
To make this bit 1, set the POECR3.GPT1ABZE bit to 0. If the POECR3.GPT1ABZE bit is 1, setting of this bit is ignored.

GPT1BME Bit (GTIOC1B Pin Port Mode Mask Enable)

This bit determines whether to switch the GTIOC1B pin to the general I/O port when at least one of the followings is set to 1: the OCSR3.OSF3 flag (when the OCSR3.OCE3 bit =1), the SPOER.GPT01HIZ bit, the SPOER.GPT02HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit = 1), the ICSRn.POEmF flag additionally selected in the POECR6 register, POECMPFR.CjFLAG flag, the ICSRn.POEmF flag additionally selected in the POECR9 register, and the POECMPFR.CjFLAG flag.
To make this bit 1, set the POECR3.GPT1ABZE bit to 0. If the POECR3.GPT1ABZE bit is 1, setting of this bit is ignored.

GPT2AME Bit (GTIOC2A Pin Port Mode Mask Enable)

This bit determines whether to switch the GTIOC2A pin to the general I/O port when at least one of the followings is set to 1: the OCSR3.OSF3 flag (when the OCSR3.OCE3 bit =1), the SPOER.GPT23HIZ bit, the SPOER.GPT02HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit = 1), the ICSRn.POEmF flag additionally selected in the POECR6B register, POECMPFR.CjFLAG flag, the ICSRn.POEmF flag additionally selected in the POECR9 register, and the POECMPFR.CjFLAG flag.
To make this bit 1, set the POECR3.GPT2ABZE bit to 0. If the POECR3.GPT2ABZE bit is 1, setting of this bit is ignored.

GPT2BME Bit (GTIOC2B Pin Port Mode Mask Enable)

This bit determines whether to switch the GTIOC2B pin to the general I/O port when at least one of the followings is set to 1: the OCSR3.OSF3 flag (when the OCSR3.OCE3 bit =1), the SPOER.GPT23HIZ bit, the SPOER.GPT02HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit = 1), the ICSRn.POEmF flag additionally selected in the POECR6B register, POECMPFR.CjFLAG flag, the ICSRn.POEmF flag additionally selected in the POECR9 register, and the POECMPFR.CjFLAG flag.
To make this bit 1, set the POECR3.GPT2ABZE bit to 0. If the POECR3.GPT2ABZE bit is 1, setting of this bit is ignored.

GPT3AME Bit (GTIOC3A Pin Port Mode Mask Enable)

This bit specifies whether switching the GTIOC3A pin to the general I/O port when one of the following flags and bit is set to 1: the SPOER.GPT23HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag additionally selected in the POECR6B register and the POECMPFR.CjFLAG flag.
When setting this bit to 1, set the POECR3.GPT3ABZE bit to 0. When the POECR3.GPT3ABZE bit is 1, the setting of the bit is ignored.

GPT3BME Bit (GTIOC3B Pin Port Mode Mask Enable)

This bit specifies whether switching the GTIOC3B pin to the general I/O port when one of the following flags and bit is set to 1: the SPOER.GPT23HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag additionally selected in the POECR6B register and the POECMPFR.CjFLAG flag.
When setting this bit to 1, set the POECR3.GPT3ABZE bit to 0. When the POECR3.GPT3ABZE bit is 1, the setting of the bit is ignored.

GPT4AME Bit (GTIOC4A Pin Port Mode Mask Enable)

This bit specifies whether switching the GTIOC4A pin to the general I/O port when one of the following flags and bit is set to 1: the OCSR4.OSF4 flag (when the OCSR4.OCE4 bit is 1), the SPOER.GPT46HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag additionally selected in the POECR10 register and the POECMPFR.CjFLAG flag.
When setting this bit to 1, set the POECR3.GPT4ABZE bit to 0. When the POECR3.GPT4ABZE bit is 1, the setting of the bit is ignored.

GPT4BME Bit (GTIOC4B Pin Port Mode Mask Enable)

This bit specifies whether switching the GTIOC4B pin to the general I/O port when one of the following flags and bit is set to 1: the OCSR4.OSF4 flag (when the OCSR4.OCE4 bit is 1), the SPOER.GPT46HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag additionally selected in the POECR10 register and the POECMPFR.CjFLAG flag.
When setting this bit to 1, set the POECR3.GPT4ABZE bit to 0. When the POECR3.GPT4ABZE bit is 1, the setting of the bit is ignored.

GPT5AME Bit (GTIOC5A Pin Port Mode Mask Enable)

This bit specifies whether switching the GTIOC5A pin to the general I/O port when one of the following flags and bit is set to 1: the OCSR4.OSF4 flag (when the OCSR4.OCE4 bit is 1), the SPOER.GPT46HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag additionally selected in the POECR10 register and the POECMPFR.CjFLAG flag.
When setting this bit to 1, set the POECR3.GPT5ABZE bit to 0. When the POECR3.GPT5ABZE bit is 1, the setting of the bit is ignored.

GPT5BME Bit (GTIOC5B Pin Port Mode Mask Enable)

This bit specifies whether switching the GTIOC5B pin to the general I/O port when one of the following flags and bit is set to 1: the OCSR4.OSF4 flag (when the OCSR4.OCE4 bit is 1), the SPOER.GPT46HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag additionally selected in the POECR10 register and the POECMPFR.CjFLAG flag.
When setting this bit to 1, set the POECR3.GPT5ABZE bit to 0. When the POECR3.GPT5ABZE bit is 1, the setting of the bit is ignored.

GPT6AME Bit (GTIOC6A Pin Port Mode Mask Enable)

This bit specifies whether switching the GTIOC6A pin to the general I/O port when one of the following flags and bit is set to 1: the OCSR4.OSF4 flag (when the OCSR4.OCE4 bit is 1), the SPOER.GPT46HIZ bit, the ICSR6.OSTSTF flag

(when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag additionally selected in the POECR10 register and the POECMPFR.CjFLAG flag.
When setting this bit to 1, set the POECR3.GPT6ABZE bit to 0. When the POECR3.GPT6ABZE bit is 1, the setting of the bit is ignored.

GPT6BME Bit (GTIOC6B Pin Port Mode Mask Enable)

This bit specifies whether switching the GTIOC6B pin to the general I/O port when one of the following flags and bit is set to 1: the OCSR4.OSF4 flag (when the OCSR4.OCE4 bit is 1), the SPOER.GPT46HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag additionally selected in the POECR10 register and the POECMPFR.CjFLAG flag.
When setting this bit to 1, set the POECR3.GPT6ABZE bit to 0. When the POECR3.GPT6ABZE bit is 1, the setting of the bit is ignored.

GPT7AME Bit (GTIOC7A Pin Port Mode Mask Enable)

This bit specifies whether switching the GTIOC7A pin to the general I/O port when one of the following flags and bit is set to 1: the OCSR5.OSF5 flag (when the OCSR5.OCE5 bit is 1), the SPOER.GPT79HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag additionally selected in the POECR11 register and the POECMPFR.CjFLAG flag.
When setting this bit to 1, set the POECR3.GPT7ABZE bit to 0. When the POECR3.GPT7ABZE bit is 1, the setting of the bit is ignored.

GPT7BME Bit (GTIOC7B Pin Port Mode Mask Enable)

This bit specifies whether switching the GTIOC7B pin to the general I/O port when one of the following flags and bit is set to 1: the OCSR5.OSF5 flag (when the OCSR5.OCE5 bit is 1), the SPOER.GPT79HIZ bit, the ICSR6.OSTSTF flag (when the ICSR6.OSTSTE bit is 1), the ICSRn.POEmF flag additionally selected in the POECR11 register and the POECMPFR.CjFLAG flag.
When setting this bit to 1, set the POECR3.GPT7ABZE bit to 0. When the POECR3.GPT7ABZE bit is 1, the setting of the bit is ignored.

23.2.36 Port Output Enable Comparator Output Detection Flag Register (POECMPFR)

Address(es): POE.POECMPFR 0009 E426h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	C5FLAG	C4FLAG	C3FLAG	C2FLAG	C1FLAG	C0FLAG
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	C0FLAG	Comparator Channel 0 Output Detection Flag	0: Comparator output not detected 1: Comparator output detected	R/(W) *1
b1	C1FLAG	Comparator Channel 1 Output Detection Flag	0: Comparator output not detected 1: Comparator output detected	R/(W) *1
b2	C2FLAG	Comparator Channel 2 Output Detection Flag	0: Comparator output not detected 1: Comparator output detected	R/(W) *1
b3	C3FLAG	Comparator Channel 3 Output Detection Flag	0: Comparator output not detected 1: Comparator output detected	R/(W) *1
b4	C4FLAG	Comparator Channel 4 Output Detection Flag	0: Comparator output not detected 1: Comparator output detected	R/(W) *1
b5	C5FLAG	Comparator Channel 5 Output Detection Flag	0: Comparator output not detected 1: Comparator output detected	R/(W) *1
b15 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. The flag can only be set to 0 by writing 0 after reading 1.

CjFLAG Flag (Comparator Channel j Output Detection Flag) (j = 0 to 5)

This flag indicates whether each comparator output is detected or not detected.

[Setting condition]

- A change from low level to high level in the comparator output is detected.
  - When the comparator is set to non-inverted output, the input voltage changes from lower to higher than the reference voltage
  - When the comparator is set to inverted output, the input voltage changes from higher to lower than the reference voltage

[Clearing condition]

- By writing 0 to the CjFLAG flag after reading CjFLAG = 1

23.2.37 Port Output Enable Comparator Request Select Register (POECMPSEL)

Address(es): POE.POECMPSEL 0009 E428h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	POEREQ5	POEREQ4	POEREQ3	POEREQ2	POEREQ1	POEREQ0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	POEREQ0	Comparator Channel 0 Output Disabling Request Enable	0: Disables output disabling request generation upon comparator output detection. 1: Enables output disabling request generation upon comparator output detection.	R/W*1
b1	POEREQ1	Comparator Channel 1 Output Disabling Request Enable	0: Disables output disabling request generation upon comparator output detection. 1: Enables output disabling request generation upon comparator output detection.	R/W*1
b2	POEREQ2	Comparator Channel 2 Output Disabling Request Enable	0: Disables output disabling request generation upon comparator output detection. 1: Enables output disabling request generation upon comparator output detection.	R/W*1
b3	POEREQ3	Comparator Channel 3 Output Disabling Request Enable	0: Disables output disabling request generation upon comparator output detection. 1: Enables output disabling request generation upon comparator output detection.	R/W*1
b4	POEREQ4	Comparator Channel 4 Output Disabling Request Enable	0: Disables output disabling request generation upon comparator output detection. 1: Enables output disabling request generation upon comparator output detection.	R/W*1
b5	POEREQ5	Comparator Channel 5 Output Disabling Request Enable	0: Disables output disabling request generation upon comparator output detection. 1: Enables output disabling request generation upon comparator output detection.	R/W*1
b15 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

The POECMPSEL register sets a comparator output detection flag to use as a control condition to disable the outputs.

POEREQj Bit (Comparator Channel j Output Disabling Request Enable) (j = 0 to 5)

This bit disables or enables output disabling request generation in response to each comparator output detection. An output disabling request is generated when one of the comparator outputs is detected.

23.2.38 Port Output Enable Comparator Request Extended Selection Register m (POECMPEXm) (m = 0 to 8)

Address(es): POE.POECMPEX0 0009 E438h, POE.POECMPEX1 0009 E439h, POE.POECMPEX2 0009 E43Ah, POE.POECMPEX3 0009 E43Bh, POE.POECMPEX4 0009 E43Ch, POE.POECMPEX5 0009 E43Dh, POE.POECMPEX6 0009 E458h, POE.POECMPEX7 0009 E459h, POE.POECMPEX8 0009 E45Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	POEREQ5	POEREQ4	POEREQ3	POEREQ2	POEREQ1	POEREQ0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	POEREQ0	Comparator Channel 0 Output Disabling Request Enable	0: Disables output disabling request generation upon comparator output detection. 1: Enables output disabling request generation upon comparator output detection.	R/W*1
b1	POEREQ1	Comparator Channel 1 Output Disabling Request Enable	0: Disables output disabling request generation upon comparator output detection. 1: Enables output disabling request generation upon comparator output detection.	R/W*1
b2	POEREQ2	Comparator Channel 2 Output Disabling Request Enable	0: Disables output disabling request generation upon comparator output detection. 1: Enables output disabling request generation upon comparator output detection.	R/W*1
b3	POEREQ3	Comparator Channel 3 Output Disabling Request Enable	0: Disables output disabling request generation upon comparator output detection. 1: Enables output disabling request generation upon comparator output detection.	R/W*1
b4	POEREQ4	Comparator Channel 4 Output Disabling Request Enable	0: Disables output disabling request generation upon comparator output detection. 1: Enables output disabling request generation upon comparator output detection.	R/W*1
b5	POEREQ5	Comparator Channel 5 Output Disabling Request Enable	0: Disables output disabling request generation upon comparator output detection. 1: Enables output disabling request generation upon comparator output detection.	R/W*1
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

The POECMPEXm register specifies the comparator output detection flag to use as a control condition to disable the outputs.

The register can select a source for each output pin group where as the POECMPSEL register selects the source for all pins. Table 23.4 lists the output pin groups and the corresponding POECMPEXm registers.

Table 23.4 Output Pin Groups and Corresponding POECMPEXm Registers

Output Pin Group	Corresponding POECMPEXm Register	Corresponding CFLAG Add Bit
MTU3, MTU4	POECMPEX0	POECR4.CMADDMT34ZE
MTU6, MTU7	POECMPEX1	POECR4B.CMADDMT67ZE
MTU0	POECMPEX2	POECR5.CMADDMT0ZE
MTU9	POECMPEX3	POECR8.CMADDMT9ZE
GPTW0, GPTW1	POECMPEX4	POECR6.CMADDGPT01ZE
GPTW2, GPTW3	POECMPEX5	POECR6B.CMADDGPT23ZE
GPTW0 to GPTW2	POECMPEX6	POECR9.CMADDGPT02ZE
GPTW4 to GPTW6	POECMPEX7	POECR10.CMADDGPT46ZE
GPTW7	POECMPEX8	POECR11.CMADDGPT79ZE

POEREQj Bit (Comparator Channel j Output Disabling Request Enable) (j = 0 to 5)

This bit disables or enables output disabling request generation in response to each comparator output detection. An output disabling request is generated when one of the comparator outputs is detected.

23.2.39 MTU0 Pin Select Register 1 (M0SELR1)

Address(es): POE.M0SELR1 0009 E460h

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	M0ASEL[3:0]	MTU0-A (MTIOC0A) Pin Select	b3 b0 0 0 0 0: Does not perform the output disabling control for any MTIOC0A pin. 0 0 0 1: Perform the output disabling control for the PB3 as the MTIOC0A pin. 0 0 1 0: Perform the output disabling control for the P31 as the MTIOC0A pin. 0 0 1 1: Perform the output disabling control for the P70 as the MTIOC0A pin. Settings other than above are prohibited.	R/W*1
b7 to b4	M0BSEL[3:0]	MTU0-B (MTIOC0B) Pin Select	b7 b4 0 0 0 0: Does not perform the output disabling control for any MTIOC0B pin. 0 0 0 1: Perform the output disabling control for the PB2 as the MTIOC0B pin. 0 0 1 0: Perform the output disabling control for the P30 as the MTIOC0B pin. Settings other than above are prohibited.	R/W*1

Note 1. Can be modified only once after a reset.

The M0SELR1 register is an 8-bit readable/writable register, and selects relevant pins to the output disabling control from the MTU0-A and MTU0-B pins respectively.

M0ASEL[3:0] Bits (MTU0-A (MTIOC0A) Pin Select)

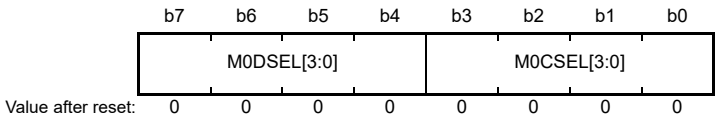
Selects a relevant MTIOC0A pin to the output disabling control.

M0BSEL[3:0] Bits (MTU0-B (MTIOC0B) Pin Select)

Selects a relevant MTIOC0B pin to the output disabling control.

23.2.40 MTU0 Pin Select Register 2 (M0SELR2)

Address(es): POE.M0SELR2 0009 E461h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	M0CSEL[3:0]	MTU0-C (MTIOC0C) Pin Select	b3 b0 0 0 0 0: Does not perform the output disabling control for any MTIOC0C pin. 0 0 0 1: Perform the output disabling control for the PB1 as the MTIOC0C pin. 0 0 1 0: Perform the output disabling control for the P27 as the MTIOC0C pin. Settings other than above are prohibited.	R/W*1
b7 to b4	M0DSEL[3:0]	MTU0-D (MTIOC0D) Pin Select	b7 b4 0 0 0 0: Does not perform the output disabling control for any MTIOC0D pin. 0 0 0 1: Perform the output disabling control for the PB0 as the MTIOC0D pin. Settings other than above are prohibited.	R/W*1

Note 1. Can be modified only once after a reset.

The M0SELR2 register is an 8-bit readable/writable register, and selects relevant pins to the output disabling control from the MTU0-C and MTU0-D pins respectively.

M0CSEL[3:0] Bits (MTU0-C (MTIOC0C) Pin Select)

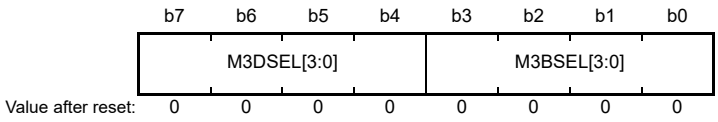
Selects a relevant MTIOC0C pin to the output disabling control.

M0DSEL[3:0] Bits (MTU0-D (MTIOC0D) Pin Select)

Selects a relevant MTIOC0D pin to the output disabling control.

23.2.41 MTU3 Pin Select Register (M3SELR)

Address(es): POE.M3SELR 0009 E462h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	M3BSEL[3:0]	MTU3-B (MTIOC3B) Pin Select	b3 b0 0 0 0 0: Does not perform the output disabling control for any MTIOC3B pin. 0 0 0 1: Perform the output disabling control for the P71 as the MTIOC3B pin. Settings other than above are prohibited.	R/W*1
b7 to b4	M3DSEL[3:0]	MTU3-D (MTIOC3D) Pin Select	b7 b4 0 0 0 0: Does not perform the output disabling control for any MTIOC3D pin. 0 0 0 1: Perform the output disabling control for the P74 as the MTIOC3D pin. Settings other than above are prohibited.	R/W*1

Note 1. Can be modified only once after a reset.

The M3SELR register is an 8-bit readable/writable register, and selects relevant pins to the output disabling control from the MTU3-B and MTU3-D pins respectively.

M3BSEL[3:0] Bits (MTU3-B (MTIOC3B) Pin Select)

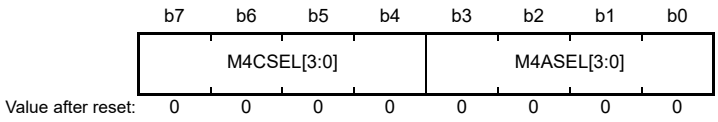
Selects a relevant MTIOC3B pin to the output disabling control.

M3DSEL[3:0] Bits (MTU3-D (MTIOC3D) Pin Select)

Selects a relevant MTIOC3D pin to the output disabling control.

23.2.42 MTU4 Pin Select Register 1 (M4SELR1)

Address(es): POE.M4SELR1 0009 E463h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	M4ASEL[3:0]	MTU4-A (MTIOC4A) Pin Select	b3 b0 0 0 0 0: Does not perform the output disabling control for any MTIOC4A pin. 0 0 0 1: Perform the output disabling control for the P72 as the MTIOC4A pin. Settings other than above are prohibited.	R/W*1
b7 to b4	M4CSEL[3:0]	MTU4-C (MTIOC4C) Pin Select	b7 b4 0 0 0 0: Does not perform the output disabling control for any MTIOC4C pin. 0 0 0 1: Perform the output disabling control for the P75 as the MTIOC4C pin. Settings other than above are prohibited.	R/W*1

Note 1. Can be modified only once after a reset.

The M4SELR1 register is an 8-bit readable/writable register, and selects relevant pins to the output disabling control from the MTU4-A and MTU4-C pins respectively.

M4ASEL[3:0] Bits (MTU4-A (MTIOC4A) Pin Select)

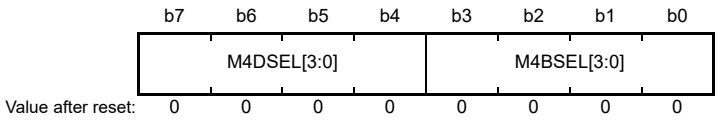
Selects a relevant MTIOC4A pin to the output disabling control.

M4CSEL[3:0] Bits (MTU4-C (MTIOC4C) Pin Select)

Selects a relevant MTIOC4C pin to the output disabling control.

23.2.43 MTU4 Pin Select Register 2 (M4SELR2)

Address(es): POE.M4SELR2 0009 E464h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	M4BSEL[3:0]	MTU4-B (MTIOC4B) Pin Select	b3 b0 0 0 0 0: Does not perform the output disabling control for any MTIOC4B pin. 0 0 0 1: Perform the output disabling control for the P73 as the MTIOC4B pin. Settings other than above are prohibited.	R/W*1
b7 to b4	M4DSEL[3:0]	MTU4-D (MTIOC4D) Pin Select	b7 b4 0 0 0 0: Does not perform the output disabling control for any MTIOC4D pin. 0 0 0 1: Perform the output disabling control for the P76 as the MTIOC4D pin. Settings other than above are prohibited.	R/W*1

Note 1. Can be modified only once after a reset.

The M4SELR2 register is an 8-bit readable/writable register, and selects relevant pins to the output disabling control from the MTU4-B and MTU4-D pins respectively.

M4BSEL[3:0] Bits (MTU4-B (MTIOC4B) Pin Select)

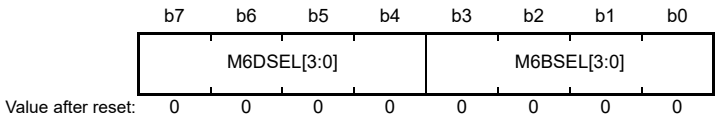
Selects a relevant MTIOC4B pin to the output disabling control.

M4DSEL[3:0] Bits (MTU4-D (MTIOC4D) Pin Select)

Selects a relevant MTIOC4D pin to the output disabling control.

23.2.44 MTU6 Pin Select Register (M6SELR)

Address(es): POE.M6SELR 0009 E465h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	M6BSEL[3:0]	MTU6-B (MTIOC6B) Pin Select	b3 b0 0 0 0 0: Does not perform the output disabling control for any MTIOC6B pin. 0 0 0 1: Perform the output disabling control for the P95 as the MTIOC6B pin. Settings other than above are prohibited.	R/W*1
b7 to b4	M6DSEL[3:0]	MTU6-D (MTIOC6D) Pin Select	b7 b4 0 0 0 0: Does not perform the output disabling control for any MTIOC6D pin. 0 0 0 1: Perform the output disabling control for the P92 as the MTIOC6D pin. Settings other than above are prohibited.	R/W*1

Note 1. Can be modified only once after a reset.

The M6SELR register is an 8-bit readable/writable register, and selects relevant pins to the output disabling control from the MTU6-B and MTU6-D pins respectively.

M6BSEL[3:0] Bits (MTU6-B (MTIOC6B) Pin Select)

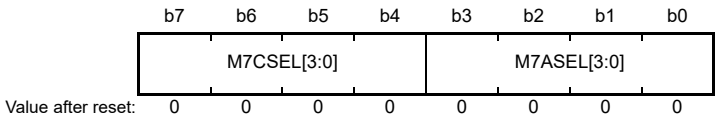
Selects a relevant MTIOC6B pin to the output disabling control.

M6DSEL[3:0] Bits (MTU6-D (MTIOC6D) Pin Select)

Selects a relevant MTIOC6D pin to the output disabling control.

23.2.45 MTU7 Pin Select Register 1 (M7SELR1)

Address(es): POE.M7SELR1 0009 E466h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	M7ASEL[3:0]	MTU7-A (MTIOC7A) Pin Select	b3 b0 0 0 0 0: Does not perform the output disabling control for any MTIOC7A pin. 0 0 0 1: Perform the output disabling control for the P94 as the MTIOC7A pin. Settings other than above are prohibited.	R/W*1
b7 to b4	M7CSEL[3:0]	MTU7-C (MTIOC7C) Pin Select	b7 b4 0 0 0 0: Does not perform the output disabling control for any MTIOC7C pin. 0 0 0 1: Perform the output disabling control for the P91 as the MTIOC7C pin. Settings other than above are prohibited.	R/W*1

Note 1. Can be modified only once after a reset.

The M7SELR1 register is an 8-bit readable/writable register, and selects relevant pins to the output disabling control from the MTU7-A and MTU7-C pins respectively.

M7ASEL[3:0] Bits (MTU7-A (MTIOC7A) Pin Select)

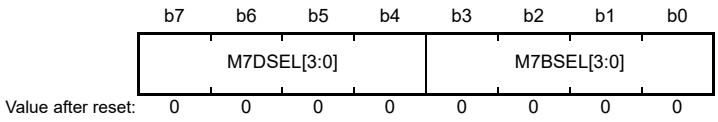
Selects a relevant MTIOC7A pin to the output disabling control.

M7CSEL[3:0] Bits (MTU7-C (MTIOC7C) Pin Select)

Selects a relevant MTIOC7C pin to the output disabling control.

23.2.46 MTU7 Pin Select Register 2 (M7SELR2)

Address(es): POE.M7SELR2 0009 E467h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	M7BSEL[3:0]	MTU7-B (MTIOC7B) Pin Select	b3 b0 0 0 0 0: Does not perform the output disabling control for any MTIOC7B pin. 0 0 0 1: Perform the output disabling control for the P93 as the MTIOC7B pin. Settings other than above are prohibited.	R/W*1
b7 to b4	M7DSEL[3:0]	MTU7-D (MTIOC7D) Pin Select	b7 b4 0 0 0 0: Does not perform the output disabling control for any MTIOC7D pin. 0 0 0 1: Perform the output disabling control for the P90 as the MTIOC7D pin. Settings other than above are prohibited.	R/W*1

Note 1. Can be modified only once after a reset.

The M7SELR2 register is an 8-bit readable/writable register, and selects relevant pins to the output disabling control from the MTU7-B and MTU7-D pins respectively.

M7BSEL[3:0] Bits (MTU7-B (MTIOC7B) Pin Select)

Selects a relevant MTIOC7B pin to the output disabling control.

M7DSEL[3:0] Bits (MTU7-D (MTIOC7D) Pin Select)

Selects a relevant MTIOC7D pin to the output disabling control.

23.2.47 MTU9 Pin Select Register 1 (M9SELR1)

Address(es): POE.M9SELR1 0009 E468h

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	M9ASEL[3:0]	MTU9-A (MTIOC9A) Pin Select	b3 b0 0 0 0 0: Does not perform the output disabling control for any MTIOC9A pin. 0 0 0 1: Perform the output disabling control for the PD7 as the MTIOC9A pin. 0 0 1 0: Perform the output disabling control for the P21 as the MTIOC9A pin. 0 0 1 1: Perform the output disabling control for the P00 as the MTIOC9A pin. Settings other than above are prohibited.	R/W*1
b7 to b4	M9BSEL[3:0]	MTU9-B (MTIOC9B) Pin Select	b7 b4 0 0 0 0: Does not perform the output disabling control for any MTIOC9B pin. 0 0 0 1: Perform the output disabling control for the PE0 as the MTIOC9B pin. 0 0 1 1: Perform the output disabling control for the P10 as the MTIOC9B pin. 0 1 0 0: Perform the output disabling control for the P22 as the MTIOC9B pin. Settings other than above are prohibited.	R/W*1

Note 1. Can be modified only once after a reset.

The M9SELR1 register is an 8-bit readable/writable register, and selects relevant pins to the output disabling control from the MTU9-A and MTU9-B pins respectively.

M9ASEL[3:0] Bits (MTU9-A (MTIOC9A) Pin Select)

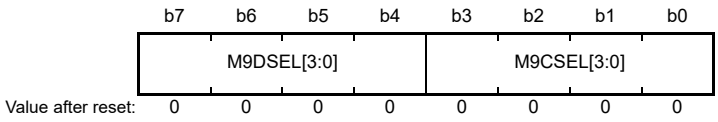
Selects a relevant MTIOC9A pin to the output disabling control.

M9BSEL[3:0] Bits (MTU9-B (MTIOC9B) Pin Select)

Selects a relevant MTIOC9B pin to the output disabling control.

23.2.48 MTU9 Pin Select Register 2 (M9SELR2)

Address(es): POE.M9SELR2 0009 E469h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	M9CSEL[3:0]	MTU9-C (MTIOC9C) Pin Select	b3 b0 0 0 0 0: Does not perform the output disabling control for any MTIOC9C pin. 0 0 0 1: Perform the output disabling control for the PD6 as the MTIOC9C pin. 0 0 1 0: Perform the output disabling control for the P20 as the MTIOC9C pin. 0 0 1 1: Perform the output disabling control for the P01 as the MTIOC9C pin. Settings other than above are prohibited.	R/W*1
b7 to b4	M9DSEL[3:0]	MTU9-D (MTIOC9D) Pin Select	b7 b4 0 0 0 0: Does not perform the output disabling control for any MTIOC9D pin. 0 0 0 1: Perform the output disabling control for the PE1 as the MTIOC9D pin. 0 0 1 0: Perform the output disabling control for the PN7 as the MTIOC9D pin. 0 0 1 1: Perform the output disabling control for the PE5 as the MTIOC9D pin. 0 1 0 0: Perform the output disabling control for the P11 as the MTIOC9D pin. Settings other than above are prohibited.	R/W*1

Note 1. Can be modified only once after a reset.

The M9SELR2 register is an 8-bit readable/writable register, and selects relevant pins to the output disabling control from the MTU9-C and MTU9-D pins respectively.

M9CSEL[3:0] Bits (MTU9-C (MTIOC9C) Pin Select)

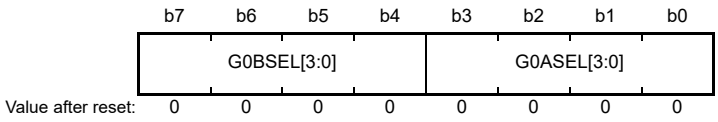
Selects a relevant MTIOC9C pin to the output disabling control.

M9DSEL[3:0] Bits (MTU9-D (MTIOC9D) Pin Select)

Selects a relevant MTIOC9D pin to the output disabling control.

23.2.49 GPTW0 Pin Select Register (G0SELR)

Address(es): POE.G0SELR 0009 E46Ah



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	G0ASEL[3:0]	GPTW0-A (GTIOC0A) Pin Select	b3 b0 0 0 0 0: Does not perform the output disabling control for any GTIOC0A pin. 0 0 0 1: Perform the output disabling control for the P71 as the GTIOC0A pin. 0 0 1 0: Perform the output disabling control for the PD7 as the GTIOC0A pin. 0 1 0 0: Perform the output disabling control for the PD2 as the GTIOC0A pin. Settings other than above are prohibited.	R/W*1
b7 to b4	G0BSEL[3:0]	GPTW0-B (GTIOC0B) Pin Select	b7 b4 0 0 0 0: Does not perform the output disabling control for any GTIOC0B pin. 0 0 0 1: Perform the output disabling control for the P74 as the GTIOC0B pin. 0 0 1 0: Perform the output disabling control for the PD6 as the GTIOC0B pin. 0 1 0 0: Perform the output disabling control for the PD1 as the GTIOC0B pin. Settings other than above are prohibited.	R/W*1

Note 1. Can be modified only once after a reset.

The G0SELR register is an 8-bit readable/writable register, and selects relevant pins to the output disabling control from the GPTW0-A and GPTW0-B pins respectively.

G0ASEL[3:0] Bits (GPTW0-A (GTIOC0A) Pin Select)

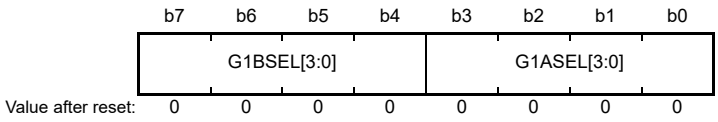
Selects a relevant GTIOC0A pin to the output disabling control.

G0BSEL[3:0] Bits (GPTW0-B (GTIOC0B) Pin Select)

Selects a relevant GTIOC0B pin to the output disabling control.

23.2.50 GPTW1 Pin Select Register (G1SELR)

Address(es): POE.G1SELR 0009 E46Bh



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	G1ASEL[3:0]	GPTW1-A (GTIOC1A) Pin Select	b3 b0 0 0 0 0: Does not perform the output disabling control for any GTIOC1A pin. 0 0 0 1: Perform the output disabling control for the P72 as the GTIOC1A pin. 0 0 1 0: Perform the output disabling control for the PD5 as the GTIOC1A pin. 0 1 0 0: Perform the output disabling control for the PD0 as the GTIOC1A pin. Settings other than above are prohibited.	R/W*1
b7 to b4	G1BSEL[3:0]	GPTW1-B (GTIOC1B) Pin Select	b7 b4 0 0 0 0: Does not perform the output disabling control for any GTIOC1B pin. 0 0 0 1: Perform the output disabling control for the P75 as the GTIOC1B pin. 0 0 1 0: Perform the output disabling control for the PD4 as the GTIOC1B pin. 0 1 0 0: Perform the output disabling control for the PB7 as the GTIOC1B pin. Settings other than above are prohibited.	R/W*1

Note 1. Can be modified only once after a reset.

The G1SELR register is an 8-bit readable/writable register, and selects relevant pins to the output disabling control from the GPTW1-A and GPTW1-B pins respectively.

G1ASEL[3:0] Bits (GPTW1-A (GTIOC1A) Pin Select)

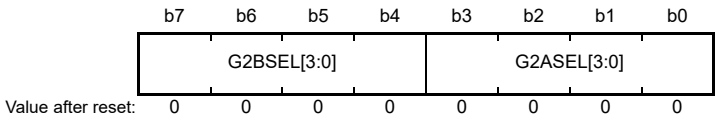
Selects a relevant GTIOC1A pin to the output disabling control.

G1BSEL[3:0] Bits (GPTW1-B (GTIOC1B) Pin Select)

Selects a relevant GTIOC1B pin to the output disabling control.

23.2.51 GPTW2 Pin Select Register (G2SELR)

Address(es): POE.G2SELR 0009 E46Ch



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	G2ASEL[3:0]	GPTW2-A (GTIOC2A) Pin Select	b3 b0 0 0 0 0: Does not perform the output disabling control for any GTIOC2A pin. 0 0 0 1: Perform the output disabling control for the P73 as the GTIOC2A pin. 0 0 1 0: Perform the output disabling control for the PD3 as the GTIOC2A pin. 0 1 0 0: Perform the output disabling control for the PB6 as the GTIOC2A pin. Settings other than above are prohibited.	R/W*1
b7 to b4	G2BSEL[3:0]	GPTW2-B (GTIOC2B) Pin Select	b7 b4 0 0 0 0: Does not perform the output disabling control for any GTIOC2B pin. 0 0 0 1: Perform the output disabling control for the P76 as the GTIOC2B pin. 0 0 1 0: Perform the output disabling control for the PD2 as the GTIOC2B pin. 0 1 0 0: Perform the output disabling control for the PB5 as the GTIOC2B pin. Settings other than above are prohibited.	R/W*1

Note 1. Can be modified only once after a reset.

The G2SELR register is an 8-bit readable/writable register, and selects relevant pins to the output disabling control from the GPTW2-A and GPTW2-B pins respectively.

G2ASEL[3:0] Bits (GPTW2-A (GTIOC2A) Pin Select)

Selects a relevant GTIOC2A pin to the output disabling control.

G2BSEL[3:0] Bits (GPTW2-B (GTIOC2B) Pin Select)

Selects a relevant GTIOC2B pin to the output disabling control.

23.2.52 GPTW3 Pin Select Register (G3SELR)

Address(es): POE.G3SELR 0009 E46Dh

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	G3ASEL[3:0]	GPTW3-A (GTIOC3A) Pin Select	b3 b0 0 0 0 0: Does not perform the output disabling control for any GTIOC3A pin. 0 0 0 1: Perform the output disabling control for the P32 as the GTIOC3A pin. 0 0 1 0: Perform the output disabling control for the PD1 as the GTIOC3A pin. 0 0 1 1: Perform the output disabling control for the PE5 as the GTIOC3A pin. 0 1 0 0: Perform the output disabling control for the PD7 as the GTIOC3A pin. 0 1 0 1: Perform the output disabling control for the PB6 as the GTIOC3A pin. 0 1 1 0: Perform the output disabling control for the P10 as the GTIOC3A pin. Settings other than above are prohibited.	R/W*1
b7 to b4	G3BSEL[3:0]	GPTW3-B (GTIOC3B) Pin Select	b7 b4 0 0 0 0: Does not perform the output disabling control for any GTIOC3B pin. 0 0 0 1: Perform the output disabling control for the P33 as the GTIOC3B pin. 0 0 1 0: Perform the output disabling control for the PD0 as the GTIOC3B pin. 0 0 1 1: Perform the output disabling control for the P11 as the GTIOC3B pin. 0 1 0 0: Perform the output disabling control for the PD6 as the GTIOC3B pin. 0 1 0 1: Perform the output disabling control for the PB5 as the GTIOC3B pin. Settings other than above are prohibited.	R/W*1

Note 1. Can be modified only once after a reset.

The G3SELR register is an 8-bit readable/writable register, and selects relevant pins to the output disabling control from the GPTW3-A and GPTW3-B pins respectively.

G3ASEL[3:0] Bits (GPTW3-A (GTIOC3A) Pin Select)

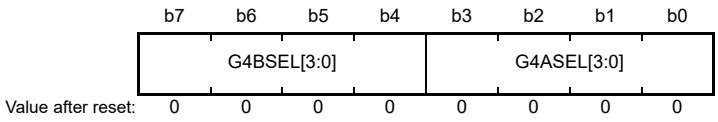
Selects a relevant GTIOC3A pin to the output disabling control.

G3BSEL[3:0] Bits (GPTW3-B (GTIOC3B) Pin Select)

Selects a relevant GTIOC3B pin to the output disabling control.

23.2.53 GPTW4 Pin Select Register (G4SELR)

Address(es): POE.G4SELR 0009 E46Eh



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	G4ASEL[3:0]	GPTW4-A (GTIOC4A) Pin Select	b3 b0 0 0 0 0: Does not perform the output disabling control for any GTIOC4A pin. 0 0 0 1: Perform the output disabling control for the P95 as the GTIOC4A pin. 0 0 1 0: Perform the output disabling control for the P71 as the GTIOC4A pin. Settings other than above are prohibited.	R/W*1
b7 to b4	G4BSEL[3:0]	GPTW4-B (GTIOC4B) Pin Select	b7 b4 0 0 0 0: Does not perform the output disabling control for any GTIOC4B pin. 0 0 0 1: Perform the output disabling control for the P92 as the GTIOC4B pin. 0 0 1 0: Perform the output disabling control for the P74 as the GTIOC4B pin. Settings other than above are prohibited.	R/W*1

Note 1. Can be modified only once after a reset.

The G4SELR register is an 8-bit readable/writable register, and selects relevant pins to the output disabling control from the GPTW4-A and GPTW4-B pins respectively.

G4ASEL[3:0] Bits (GPTW4-A (GTIOC4A) Pin Select)

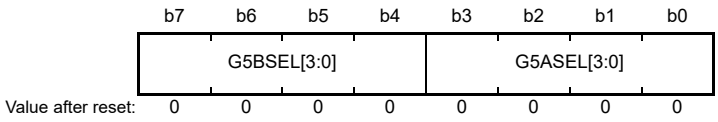
Selects a relevant GTIOC4A pin to the output disabling control.

G4BSEL[3:0] Bits (GPTW4-B (GTIOC4B) Pin Select)

Selects a relevant GTIOC4B pin to the output disabling control.

23.2.54 GPTW5 Pin Select Register (G5SELR)

Address(es): POE.G5SELR 0009 E46Fh



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	G5ASEL[3:0]	GPTW5-A (GTIOC5A) Pin Select	b3 b0 0 0 0 0: Does not perform the output disabling control for any GTIOC5A pin. 0 0 0 1: Perform the output disabling control for the P94 as the GTIOC5A pin. 0 0 1 0: Perform the output disabling control for the P72 as the GTIOC5A pin. Settings other than above are prohibited.	R/W*1
b7 to b4	G5BSEL[3:0]	GPTW5-B (GTIOC5B) Pin Select	b7 b4 0 0 0 0: Does not perform the output disabling control for any GTIOC5B pin. 0 0 0 1: Perform the output disabling control for the P91 as the GTIOC5B pin. 0 0 1 0: Perform the output disabling control for the P75 as the GTIOC5B pin. Settings other than above are prohibited.	R/W*1

Note 1. Can be modified only once after a reset.

The G5SELR register is an 8-bit readable/writable register, and selects relevant pins to the output disabling control from the GPTW5-A and GPTW5-B pins respectively.

G5ASEL[3:0] Bits (GPTW5-A (GTIOC5A) Pin Select)

Selects a relevant GTIOC5A pin to the output disabling control.

G5BSEL[3:0] Bits (GPTW5-B (GTIOC5B) Pin Select)

Selects a relevant GTIOC5B pin to the output disabling control.

23.2.55 GPTW6 Pin Select Register (G6SELR)

Address(es): POE.G6SELR 0009 E470h

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	G6ASEL[3:0]	GPTW6-A (GTIOC6A) Pin Select	b3 b0 0 0 0 0: Does not perform the output disabling control for any GTIOC6A pin. 0 0 0 1: Perform the output disabling control for the P93 as the GTIOC6A pin. 0 0 1 0: Perform the output disabling control for the P73 as the GTIOC6A pin. Settings other than above are prohibited.	R/W*1
b7 to b4	G6BSEL[3:0]	GPTW6-B (GTIOC6B) Pin Select	b7 b4 0 0 0 0: Does not perform the output disabling control for any GTIOC6B pin. 0 0 0 1: Perform the output disabling control for the P90 as the GTIOC6B pin. 0 0 1 0: Perform the output disabling control for the P76 as the GTIOC6B pin. Settings other than above are prohibited.	R/W*1

Note 1. Can be modified only once after a reset.

The G6SELR register is an 8-bit readable/writable register, and selects relevant pins to the output disabling control from the GPTW6-A and GPTW6-B pins respectively.

G6ASEL[3:0] Bits (GPTW6-A (GTIOC6A) Pin Select)

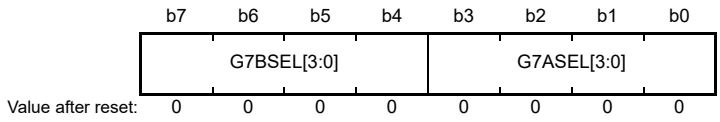
Selects a relevant GTIOC6A pin to the output disabling control.

G6BSEL[3:0] Bits (GPTW6-B (GTIOC6B) Pin Select)

Selects a relevant GTIOC6B pin to the output disabling control.

23.2.56 GPTW7 Pin Select Register (G7SELR)

Address(es): POE.G7SELR 0009 E471h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	G7ASEL[3:0]	GPTW7-A (GTIOC7A) Pin Select	b3 b0 0 0 0 0: Does not perform the output disabling control for any GTIOC7A pin. 0 0 0 1: Perform the output disabling control for the P95 as the GTIOC7A pin. 0 0 1 0: Perform the output disabling control for the PB2 as the GTIOC7A pin. 0 0 1 1: Perform the output disabling control for the P32 as the GTIOC7A pin. 0 1 0 0: Perform the output disabling control for the PD5 as the GTIOC7A pin. Settings other than above are prohibited.	R/W*1
b7 to b4	G7BSEL[3:0]	GPTW7-B (GTIOC7B) Pin Select	b7 b4 0 0 0 0: Does not perform the output disabling control for any GTIOC7B pin. 0 0 0 1: Perform the output disabling control for the P92 as the GTIOC7B pin. 0 0 1 0: Perform the output disabling control for the PB1 as the GTIOC7B pin. 0 0 1 1: Perform the output disabling control for the P33 as the GTIOC7B pin. 0 1 0 0: Perform the output disabling control for the PD3 as the GTIOC7B pin. Settings other than above are prohibited.	R/W*1

Note 1. Can be modified only once after a reset.

The G7SELR register is an 8-bit readable/writable register, and selects relevant pins to the output disabling control from the GPTW7-A and GPTW7-B pins respectively.

G7ASEL[3:0] Bits (GPTW7-A (GTIOC7A) Pin Select)

Selects a relevant GTIOC7A pin to the output disabling control.

G7BSEL[3:0] Bits (GPTW7-B (GTIOC7B) Pin Select)

Selects a relevant GTIOC7B pin to the output disabling control.

23.2.57 Input Signal Mask Control Register 0 (IMCR0)

Address(es): POE.IMCR0 0009 E45Ch

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	POE0MS[5:0]	POE0 Mask Signal Select	b5 b0 000000: Does not mask the POE0# signal 000001: Mask the POE0# signal with the MTIOC0A signal 000010: Mask the POE0# signal with the MTIOC0B signal 000011: Mask the POE0# signal with the MTIOC0C signal 000100: Mask the POE0# signal with the MTIOC0D signal 000101: Mask the POE0# signal with the MTIOC1A signal 000110: Mask the POE0# signal with the MTIOC1B signal 000111: Mask the POE0# signal with the MTIOC2A signal 001000: Mask the POE0# signal with the MTIOC2B signal 001001: Mask the POE0# signal with the MTIOC3A signal 001010: Mask the POE0# signal with the MTIOC3B signal 001011: Mask the POE0# signal with the MTIOC3C signal 001100: Mask the POE0# signal with the MTIOC3D signal 001101: Mask the POE0# signal with the MTIOC4A signal 001110: Mask the POE0# signal with the MTIOC4B signal 001111: Mask the POE0# signal with the MTIOC4C signal 010000: Mask the POE0# signal with the MTIOC4D signal 010001: Mask the POE0# signal with the MTIOC6A signal 010010: Mask the POE0# signal with the MTIOC6B signal 010011: Mask the POE0# signal with the MTIOC6C signal 010100: Mask the POE0# signal with the MTIOC6D signal 010101: Mask the POE0# signal with the MTIOC7A signal 010110: Mask the POE0# signal with the MTIOC7B signal 010111: Mask the POE0# signal with the MTIOC7C signal 011000: Mask the POE0# signal with the MTIOC7D signal 011001: Mask the POE0# signal with the MTIOC9A signal 011010: Mask the POE0# signal with the MTIOC9B signal 011011: Mask the POE0# signal with the MTIOC9C signal 011100: Mask the POE0# signal with the MTIOC9D signal 011101: Mask the POE0# signal with the GTIOC0A signal 011110: Mask the POE0# signal with the GTIOC0B signal 011111: Mask the POE0# signal with the GTIOC1A signal 100000: Mask the POE0# signal with the GTIOC1B signal 100001: Mask the POE0# signal with the GTIOC2A signal 100010: Mask the POE0# signal with the GTIOC2B signal 100011: Mask the POE0# signal with the GTIOC3A signal 100100: Mask the POE0# signal with the GTIOC3B signal 100101: Mask the POE0# signal with the GTIOC4A signal 100110: Mask the POE0# signal with the GTIOC4B signal 100111: Mask the POE0# signal with the GTIOC5A signal 101000: Mask the POE0# signal with the GTIOC5B signal 101001: Mask the POE0# signal with the GTIOC6A signal 101010: Mask the POE0# signal with the GTIOC6B signal 101011: Mask the POE0# signal with the GTIOC7A signal 101100: Mask the POE0# signal with the GTIOC7B signal Settings other than above are prohibited.	R/W*1
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

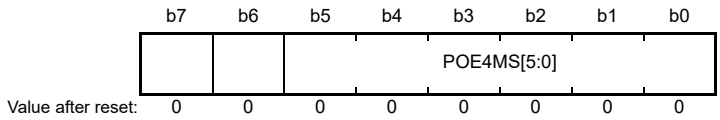
The IMCR0 register is an 8-bit readable/writable register used to mask the output disabling requests by the POE0# pin.

POE0MS[5:0] Bits (POE0 Mask Signal Select)

These bits are used to select a signal that masks the POE0# signal. When the selected pin drives high, the POE0# signal is masked.

23.2.58 Input Signal Mask Control Register 1 (IMCR1)

Address(es): POE.IMCR1 0009 E45Dh



Bit	Symbol	Bit Name	Description	R/W
b5 to b0	POE4MS[5:0]	POE4 Mask Signal Select	b5 b0 000000: Does not mask the POE4# signal 000001: Mask the POE4# signal with the MTIOC0A signal 000010: Mask the POE4# signal with the MTIOC0B signal 000011: Mask the POE4# signal with the MTIOC0C signal 000100: Mask the POE4# signal with the MTIOC0D signal 000101: Mask the POE4# signal with the MTIOC1A signal 000110: Mask the POE4# signal with the MTIOC1B signal 000111: Mask the POE4# signal with the MTIOC2A signal 001000: Mask the POE4# signal with the MTIOC2B signal 001001: Mask the POE4# signal with the MTIOC3A signal 001010: Mask the POE4# signal with the MTIOC3B signal 001011: Mask the POE4# signal with the MTIOC3C signal 001100: Mask the POE4# signal with the MTIOC3D signal 001101: Mask the POE4# signal with the MTIOC4A signal 001110: Mask the POE4# signal with the MTIOC4B signal 001111: Mask the POE4# signal with the MTIOC4C signal 010000: Mask the POE4# signal with the MTIOC4D signal 010001: Mask the POE4# signal with the MTIOC6A signal 010010: Mask the POE4# signal with the MTIOC6B signal 010011: Mask the POE4# signal with the MTIOC6C signal 010100: Mask the POE4# signal with the MTIOC6D signal 010101: Mask the POE4# signal with the MTIOC7A signal 010110: Mask the POE4# signal with the MTIOC7B signal 010111: Mask the POE4# signal with the MTIOC7C signal 011000: Mask the POE4# signal with the MTIOC7D signal 011001: Mask the POE4# signal with the MTIOC9A signal 011010: Mask the POE4# signal with the MTIOC9B signal 011011: Mask the POE4# signal with the MTIOC9C signal 011100: Mask the POE4# signal with the MTIOC9D signal 011101: Mask the POE4# signal with the GTIOC0A signal 011110: Mask the POE4# signal with the GTIOC0B signal 011111: Mask the POE4# signal with the GTIOC1A signal 100000: Mask the POE4# signal with the GTIOC1B signal 100001: Mask the POE4# signal with the GTIOC2A signal 100010: Mask the POE4# signal with the GTIOC2B signal 100011: Mask the POE4# signal with the GTIOC3A signal 100100: Mask the POE4# signal with the GTIOC3B signal 100101: Mask the POE4# signal with the GTIOC4A signal 100110: Mask the POE4# signal with the GTIOC4B signal 100111: Mask the POE4# signal with the GTIOC5A signal 101000: Mask the POE4# signal with the GTIOC5B signal 101001: Mask the POE4# signal with the GTIOC6A signal 101010: Mask the POE4# signal with the GTIOC6B signal 101011: Mask the POE4# signal with the GTIOC7A signal 101100: Mask the POE4# signal with the GTIOC7B signal Settings other than above are prohibited.	R/W*1
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

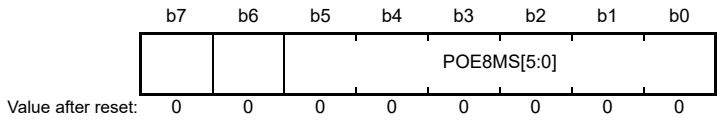
The IMCR1 register is an 8-bit readable/writable register used to mask the output disabling requests by the POE4# pin.

POE4MS[5:0] Bits (POE4 Mask Signal Select)

These bits are used to select a signal that masks the POE4# signal. When the selected pin drives high, the POE4# signal is masked.

23.2.59 Input Signal Mask Control Register 2 (IMCR2)

Address(es): POE.IMCR2 0009 E45Eh



Bit	Symbol	Bit Name	Description	R/W
b5 to b0	POE8MS[5:0]	POE8 Mask Signal Select	b5 b0 000000: Does not mask the POE8# signal 000001: Mask the POE8# signal with the MTIOC0A signal 000010: Mask the POE8# signal with the MTIOC0B signal 000011: Mask the POE8# signal with the MTIOC0C signal 000100: Mask the POE8# signal with the MTIOC0D signal 000101: Mask the POE8# signal with the MTIOC1A signal 000110: Mask the POE8# signal with the MTIOC1B signal 000111: Mask the POE8# signal with the MTIOC2A signal 001000: Mask the POE8# signal with the MTIOC2B signal 001001: Mask the POE8# signal with the MTIOC3A signal 001010: Mask the POE8# signal with the MTIOC3B signal 001011: Mask the POE8# signal with the MTIOC3C signal 001100: Mask the POE8# signal with the MTIOC3D signal 001101: Mask the POE8# signal with the MTIOC4A signal 001110: Mask the POE8# signal with the MTIOC4B signal 001111: Mask the POE8# signal with the MTIOC4C signal 010000: Mask the POE8# signal with the MTIOC4D signal 010001: Mask the POE8# signal with the MTIOC6A signal 010010: Mask the POE8# signal with the MTIOC6B signal 010011: Mask the POE8# signal with the MTIOC6C signal 010100: Mask the POE8# signal with the MTIOC6D signal 010101: Mask the POE8# signal with the MTIOC7A signal 010110: Mask the POE8# signal with the MTIOC7B signal 010111: Mask the POE8# signal with the MTIOC7C signal 011000: Mask the POE8# signal with the MTIOC7D signal 011001: Mask the POE8# signal with the MTIOC9A signal 011010: Mask the POE8# signal with the MTIOC9B signal 011011: Mask the POE8# signal with the MTIOC9C signal 011100: Mask the POE8# signal with the MTIOC9D signal 011101: Mask the POE8# signal with the GTIOC0A signal 011110: Mask the POE8# signal with the GTIOC0B signal 011111: Mask the POE8# signal with the GTIOC1A signal 100000: Mask the POE8# signal with the GTIOC1B signal 100001: Mask the POE8# signal with the GTIOC2A signal 100010: Mask the POE8# signal with the GTIOC2B signal 100011: Mask the POE8# signal with the GTIOC3A signal 100100: Mask the POE8# signal with the GTIOC3B signal 100101: Mask the POE8# signal with the GTIOC4A signal 100110: Mask the POE8# signal with the GTIOC4B signal 100111: Mask the POE8# signal with the GTIOC5A signal 101000: Mask the POE8# signal with the GTIOC5B signal 101001: Mask the POE8# signal with the GTIOC6A signal 101010: Mask the POE8# signal with the GTIOC6B signal 101011: Mask the POE8# signal with the GTIOC7A signal 101100: Mask the POE8# signal with the GTIOC7B signal Settings other than above are prohibited.	R/W*1
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

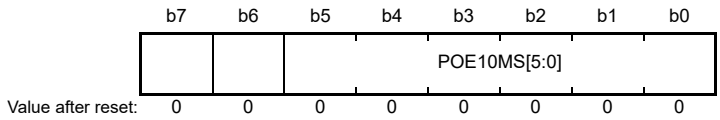
The IMCR2 register is an 8-bit readable/writable register used to mask the output disabling requests by the POE8# pin.

POE8MS[5:0] Bits (POE8 Mask Signal Select)

These bits are used to select a signal that masks the POE8# signal. When the selected pin drives high, the POE8# signal is masked.

23.2.60 Input Signal Mask Control Register 3 (IMCR3)

Address(es): POE.IMCR3 0009 E45Fh



Bit	Symbol	Bit Name	Description	R/W
b5 to b0	POE10MS[5:0]	POE10 Mask Signal Select	b5 b0 000000: Does not mask the POE10# signal 000001: Mask the POE10# signal with the MTIOC0A signal 000010: Mask the POE10# signal with the MTIOC0B signal 000011: Mask the POE10# signal with the MTIOC0C signal 000100: Mask the POE10# signal with the MTIOC0D signal 000101: Mask the POE10# signal with the MTIOC1A signal 000110: Mask the POE10# signal with the MTIOC1B signal 000111: Mask the POE10# signal with the MTIOC2A signal 001000: Mask the POE10# signal with the MTIOC2B signal 001001: Mask the POE10# signal with the MTIOC3A signal 001010: Mask the POE10# signal with the MTIOC3B signal 001011: Mask the POE10# signal with the MTIOC3C signal 001100: Mask the POE10# signal with the MTIOC3D signal 001101: Mask the POE10# signal with the MTIOC4A signal 001110: Mask the POE10# signal with the MTIOC4B signal 001111: Mask the POE10# signal with the MTIOC4C signal 010000: Mask the POE10# signal with the MTIOC4D signal 010001: Mask the POE10# signal with the MTIOC6A signal 010010: Mask the POE10# signal with the MTIOC6B signal 010011: Mask the POE10# signal with the MTIOC6C signal 010100: Mask the POE10# signal with the MTIOC6D signal 010101: Mask the POE10# signal with the MTIOC7A signal 010110: Mask the POE10# signal with the MTIOC7B signal 010111: Mask the POE10# signal with the MTIOC7C signal 011000: Mask the POE10# signal with the MTIOC7D signal 011001: Mask the POE10# signal with the MTIOC9A signal 011010: Mask the POE10# signal with the MTIOC9B signal 011011: Mask the POE10# signal with the MTIOC9C signal 011100: Mask the POE10# signal with the MTIOC9D signal 011101: Mask the POE10# signal with the GTIOC0A signal 011110: Mask the POE10# signal with the GTIOC0B signal 011111: Mask the POE10# signal with the GTIOC1A signal 100000: Mask the POE10# signal with the GTIOC1B signal 100001: Mask the POE10# signal with the GTIOC2A signal 100010: Mask the POE10# signal with the GTIOC2B signal 100011: Mask the POE10# signal with the GTIOC3A signal 100100: Mask the POE10# signal with the GTIOC3B signal 100101: Mask the POE10# signal with the GTIOC4A signal 100110: Mask the POE10# signal with the GTIOC4B signal 100111: Mask the POE10# signal with the GTIOC5A signal 101000: Mask the POE10# signal with the GTIOC5B signal 101001: Mask the POE10# signal with the GTIOC6A signal 101010: Mask the POE10# signal with the GTIOC6B signal 101011: Mask the POE10# signal with the GTIOC7A signal 101100: Mask the POE10# signal with the GTIOC7B signal Settings other than above are prohibited.	R/W*1
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

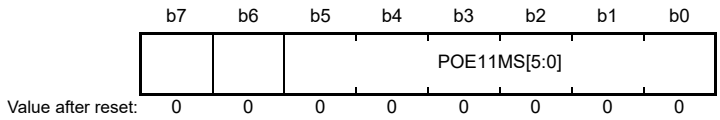
The IMCR3 register is an 8-bit readable/writable register used to mask the output disabling requests by the POE10# pin.

POE10MS[5:0] Bits (POE10 Mask Signal Select)

These bits are used to select a signal that masks the POE10# signal. When the selected pin drives high, the POE10# signal is masked.

23.2.61 Input Signal Mask Control Register 4 (IMCR4)

Address(es): POE.IMCR4 0009 E474h



Bit	Symbol	Bit Name	Description	R/W
b5 to b0	POE11MS[5:0]	POE11 Mask Signal Select	b5 b0 000000: Does not mask the POE11# signal 000001: Mask the POE11# signal with the MTIOC0A signal 000010: Mask the POE11# signal with the MTIOC0B signal 000011: Mask the POE11# signal with the MTIOC0C signal 000100: Mask the POE11# signal with the MTIOC0D signal 000101: Mask the POE11# signal with the MTIOC1A signal 000110: Mask the POE11# signal with the MTIOC1B signal 000111: Mask the POE11# signal with the MTIOC2A signal 001000: Mask the POE11# signal with the MTIOC2B signal 001001: Mask the POE11# signal with the MTIOC3A signal 001010: Mask the POE11# signal with the MTIOC3B signal 001011: Mask the POE11# signal with the MTIOC3C signal 001100: Mask the POE11# signal with the MTIOC3D signal 001101: Mask the POE11# signal with the MTIOC4A signal 001110: Mask the POE11# signal with the MTIOC4B signal 001111: Mask the POE11# signal with the MTIOC4C signal 010000: Mask the POE11# signal with the MTIOC4D signal 010001: Mask the POE11# signal with the MTIOC6A signal 010010: Mask the POE11# signal with the MTIOC6B signal 010011: Mask the POE11# signal with the MTIOC6C signal 010100: Mask the POE11# signal with the MTIOC6D signal 010101: Mask the POE11# signal with the MTIOC7A signal 010110: Mask the POE11# signal with the MTIOC7B signal 010111: Mask the POE11# signal with the MTIOC7C signal 011000: Mask the POE11# signal with the MTIOC7D signal 011001: Mask the POE11# signal with the MTIOC9A signal 011010: Mask the POE11# signal with the MTIOC9B signal 011011: Mask the POE11# signal with the MTIOC9C signal 011100: Mask the POE11# signal with the MTIOC9D signal 011101: Mask the POE11# signal with the GTIOC0A signal 011110: Mask the POE11# signal with the GTIOC0B signal 011111: Mask the POE11# signal with the GTIOC1A signal 100000: Mask the POE11# signal with the GTIOC1B signal 100001: Mask the POE11# signal with the GTIOC2A signal 100010: Mask the POE11# signal with the GTIOC2B signal 100011: Mask the POE11# signal with the GTIOC3A signal 100100: Mask the POE11# signal with the GTIOC3B signal 100101: Mask the POE11# signal with the GTIOC4A signal 100110: Mask the POE11# signal with the GTIOC4B signal 100111: Mask the POE11# signal with the GTIOC5A signal 101000: Mask the POE11# signal with the GTIOC5B signal 101001: Mask the POE11# signal with the GTIOC6A signal 101010: Mask the POE11# signal with the GTIOC6B signal 101011: Mask the POE11# signal with the GTIOC7A signal 101100: Mask the POE11# signal with the GTIOC7B signal Settings other than above are prohibited.	R/W*1
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

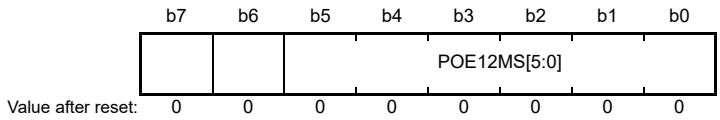
The IMCR4 register is an 8-bit readable/writable register used to mask the output disabling requests by the POE11# pin.

POE11MS[5:0] Bits (POE11 Mask Signal Select)

These bits are used to select a signal that masks the POE11# signal. When the selected pin drives high, the POE11# signal is masked.

23.2.62 Input Signal Mask Control Register 5 (IMCR5)

Address(es): POE.IMCR5 0009 E475h



Bit	Symbol	Bit Name	Description	R/W
b5 to b0	POE12MS[5:0]	POE12 Mask Signal Select	b5 b0 000000: Does not mask the POE12# signal 000001: Mask the POE12# signal with the MTIOC0A signal 000010: Mask the POE12# signal with the MTIOC0B signal 000011: Mask the POE12# signal with the MTIOC0C signal 000100: Mask the POE12# signal with the MTIOC0D signal 000101: Mask the POE12# signal with the MTIOC1A signal 000110: Mask the POE12# signal with the MTIOC1B signal 000111: Mask the POE12# signal with the MTIOC2A signal 001000: Mask the POE12# signal with the MTIOC2B signal 001001: Mask the POE12# signal with the MTIOC3A signal 001010: Mask the POE12# signal with the MTIOC3B signal 001011: Mask the POE12# signal with the MTIOC3C signal 001100: Mask the POE12# signal with the MTIOC3D signal 001101: Mask the POE12# signal with the MTIOC4A signal 001110: Mask the POE12# signal with the MTIOC4B signal 001111: Mask the POE12# signal with the MTIOC4C signal 010000: Mask the POE12# signal with the MTIOC4D signal 010001: Mask the POE12# signal with the MTIOC6A signal 010010: Mask the POE12# signal with the MTIOC6B signal 010011: Mask the POE12# signal with the MTIOC6C signal 010100: Mask the POE12# signal with the MTIOC6D signal 010101: Mask the POE12# signal with the MTIOC7A signal 010110: Mask the POE12# signal with the MTIOC7B signal 010111: Mask the POE12# signal with the MTIOC7C signal 011000: Mask the POE12# signal with the MTIOC7D signal 011001: Mask the POE12# signal with the MTIOC9A signal 011010: Mask the POE12# signal with the MTIOC9B signal 011011: Mask the POE12# signal with the MTIOC9C signal 011100: Mask the POE12# signal with the MTIOC9D signal 011101: Mask the POE12# signal with the GTIOC0A signal 011110: Mask the POE12# signal with the GTIOC0B signal 011111: Mask the POE12# signal with the GTIOC1A signal 100000: Mask the POE12# signal with the GTIOC1B signal 100001: Mask the POE12# signal with the GTIOC2A signal 100010: Mask the POE12# signal with the GTIOC2B signal 100011: Mask the POE12# signal with the GTIOC3A signal 100100: Mask the POE12# signal with the GTIOC3B signal 100101: Mask the POE12# signal with the GTIOC4A signal 100110: Mask the POE12# signal with the GTIOC4B signal 100111: Mask the POE12# signal with the GTIOC5A signal 101000: Mask the POE12# signal with the GTIOC5B signal 101001: Mask the POE12# signal with the GTIOC6A signal 101010: Mask the POE12# signal with the GTIOC6B signal 101011: Mask the POE12# signal with the GTIOC7A signal 101100: Mask the POE12# signal with the GTIOC7B signal Settings other than above are prohibited.	R/W*1
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

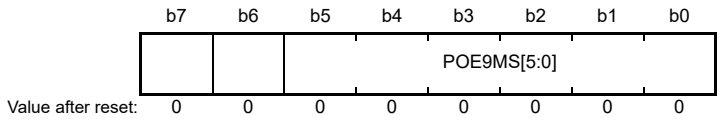
The IMCR5 register is an 8-bit readable/writable register used to mask the output disabling requests by the POE12# pin.

POE12MS[5:0] Bits (POE12 Mask Signal Select)

These bits are used to select a signal that masks the POE12# signal. When the selected pin drives high, the POE12# signal is masked.

23.2.63 Input Signal Mask Control Register 6 (IMCR6)

Address(es): POE.IMCR6 0009 E476h



Bit	Symbol	Bit Name	Description	R/W
b5 to b0	POE9MS[5:0]	POE9 Mask Signal Select	b5 b0 000000: Does not mask the POE9# signal 000001: Mask the POE9# signal with the MTIOC0A signal 000010: Mask the POE9# signal with the MTIOC0B signal 000011: Mask the POE9# signal with the MTIOC0C signal 000100: Mask the POE9# signal with the MTIOC0D signal 000101: Mask the POE9# signal with the MTIOC1A signal 000110: Mask the POE9# signal with the MTIOC1B signal 000111: Mask the POE9# signal with the MTIOC2A signal 001000: Mask the POE9# signal with the MTIOC2B signal 001001: Mask the POE9# signal with the MTIOC3A signal 001010: Mask the POE9# signal with the MTIOC3B signal 001011: Mask the POE9# signal with the MTIOC3C signal 001100: Mask the POE9# signal with the MTIOC3D signal 001101: Mask the POE9# signal with the MTIOC4A signal 001110: Mask the POE9# signal with the MTIOC4B signal 001111: Mask the POE9# signal with the MTIOC4C signal 010000: Mask the POE9# signal with the MTIOC4D signal 010001: Mask the POE9# signal with the MTIOC6A signal 010010: Mask the POE9# signal with the MTIOC6B signal 010011: Mask the POE9# signal with the MTIOC6C signal 010100: Mask the POE9# signal with the MTIOC6D signal 010101: Mask the POE9# signal with the MTIOC7A signal 010110: Mask the POE9# signal with the MTIOC7B signal 010111: Mask the POE9# signal with the MTIOC7C signal 011000: Mask the POE9# signal with the MTIOC7D signal 011001: Mask the POE9# signal with the MTIOC9A signal 011010: Mask the POE9# signal with the MTIOC9B signal 011011: Mask the POE9# signal with the MTIOC9C signal 011100: Mask the POE9# signal with the MTIOC9D signal 011101: Mask the POE9# signal with the GTIOC0A signal 011110: Mask the POE9# signal with the GTIOC0B signal 011111: Mask the POE9# signal with the GTIOC1A signal 100000: Mask the POE9# signal with the GTIOC1B signal 100001: Mask the POE9# signal with the GTIOC2A signal 100010: Mask the POE9# signal with the GTIOC2B signal 100011: Mask the POE9# signal with the GTIOC3A signal 100100: Mask the POE9# signal with the GTIOC3B signal 100101: Mask the POE9# signal with the GTIOC4A signal 100110: Mask the POE9# signal with the GTIOC4B signal 100111: Mask the POE9# signal with the GTIOC5A signal 101000: Mask the POE9# signal with the GTIOC5B signal 101001: Mask the POE9# signal with the GTIOC6A signal 101010: Mask the POE9# signal with the GTIOC6B signal 101011: Mask the POE9# signal with the GTIOC7A signal 101100: Mask the POE9# signal with the GTIOC7B signal Settings other than above are prohibited.	R/W*1
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

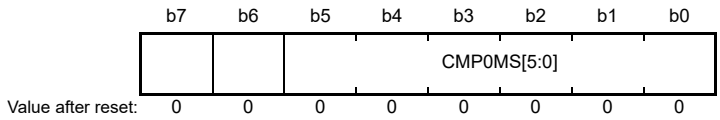
The IMCR6 register is an 8-bit readable/writable register used to mask the output disabling requests by the POE9# pin.

POE9MS[5:0] Bits (POE9 Mask Signal Select)

These bits are used to select a signal that masks the POE9# signal. When the selected pin drives high, the POE9# signal is masked.

23.2.64 Input Signal Mask Control Register 9 (IMCR9)

Address(es): POE.IMCR9 0009 E479h



Bit	Symbol	Bit Name	Description	R/W
b5 to b0	CMP0MS[5:0]	COMP0 Mask Signal Select	b5 b0 000000: Does not mask the COMP0 level detection signal 000001: Mask COMP0 level detection signal with MTIOC0A signal 000010: Mask COMP0 level detection signal with MTIOC0B signal 000011: Mask COMP0 level detection signal with MTIOC0C signal 000100: Mask COMP0 level detection signal with MTIOC0D signal 000101: Mask COMP0 level detection signal with MTIOC1A signal 000110: Mask COMP0 level detection signal with MTIOC1B signal 000111: Mask COMP0 level detection signal with MTIOC2A signal 001000: Mask COMP0 level detection signal with MTIOC2B signal 001001: Mask COMP0 level detection signal with MTIOC3A signal 001010: Mask COMP0 level detection signal with MTIOC3B signal 001011: Mask COMP0 level detection signal with MTIOC3C signal 001100: Mask COMP0 level detection signal with MTIOC3D signal 001101: Mask COMP0 level detection signal with MTIOC4A signal 001110: Mask COMP0 level detection signal with MTIOC4B signal 001111: Mask COMP0 level detection signal with MTIOC4C signal 010000: Mask COMP0 level detection signal with MTIOC4D signal 010001: Mask COMP0 level detection signal with MTIOC6A signal 010010: Mask COMP0 level detection signal with MTIOC6B signal 010011: Mask COMP0 level detection signal with MTIOC6C signal 010100: Mask COMP0 level detection signal with MTIOC6D signal 010101: Mask COMP0 level detection signal with MTIOC7A signal 010110: Mask COMP0 level detection signal with MTIOC7B signal 010111: Mask COMP0 level detection signal with MTIOC7C signal 011000: Mask COMP0 level detection signal with MTIOC7D signal 011001: Mask COMP0 level detection signal with MTIOC9A signal 011010: Mask COMP0 level detection signal with MTIOC9B signal 011011: Mask COMP0 level detection signal with MTIOC9C signal 011100: Mask COMP0 level detection signal with MTIOC9D signal 011101: Mask COMP0 level detection signal with GTIOC0A signal 011110: Mask COMP0 level detection signal with GTIOC0B signal 011111: Mask COMP0 level detection signal with GTIOC1A signal 100000: Mask COMP0 level detection signal with GTIOC1B signal 100001: Mask COMP0 level detection signal with GTIOC2A signal 100010: Mask COMP0 level detection signal with GTIOC2B signal 100011: Mask COMP0 level detection signal with GTIOC3A signal 100100: Mask COMP0 level detection signal with GTIOC3B signal 100101: Mask COMP0 level detection signal with GTIOC4A signal 100110: Mask COMP0 level detection signal with GTIOC4B signal 100111: Mask COMP0 level detection signal with GTIOC5A signal 101000: Mask COMP0 level detection signal with GTIOC5B signal 101001: Mask COMP0 level detection signal with GTIOC6A signal 101010: Mask COMP0 level detection signal with GTIOC6B signal 101011: Mask COMP0 level detection signal with GTIOC7A signal 101100: Mask COMP0 level detection signal with GTIOC7B signal Settings other than above are prohibited.	R/W*1
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

The IMCR9 register is an 8-bit readable/writable register used to mask the output disabling requests by the COMP0 level detection signal.

CMP0MS[5:0] Bits (COMP0 Mask Signal Select)

These bits are used to select a signal that masks the COMP0 level detection signal. When the selected pin drives high, the COMP0 level detection signal is masked.

23.2.65 Input Signal Mask Control Register 10 (IMCR10)

Address(es): POE.IMCR10 0009 E47Ah

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	CMP1MS[5:0]	COMP1 Mask Signal Select	b5 b0 000000: Does not mask the COMP1 level detection signal 000001: Mask COMP1 level detection signal with MTIOC0A signal 000010: Mask COMP1 level detection signal with MTIOC0B signal 000011: Mask COMP1 level detection signal with MTIOC0C signal 000100: Mask COMP1 level detection signal with MTIOC0D signal 000101: Mask COMP1 level detection signal with MTIOC1A signal 000110: Mask COMP1 level detection signal with MTIOC1B signal 000111: Mask COMP1 level detection signal with MTIOC2A signal 001000: Mask COMP1 level detection signal with MTIOC2B signal 001001: Mask COMP1 level detection signal with MTIOC3A signal 001010: Mask COMP1 level detection signal with MTIOC3B signal 001011: Mask COMP1 level detection signal with MTIOC3C signal 001100: Mask COMP1 level detection signal with MTIOC3D signal 001101: Mask COMP1 level detection signal with MTIOC4A signal 001110: Mask COMP1 level detection signal with MTIOC4B signal 001111: Mask COMP1 level detection signal with MTIOC4C signal 010000: Mask COMP1 level detection signal with MTIOC4D signal 010001: Mask COMP1 level detection signal with MTIOC6A signal 010010: Mask COMP1 level detection signal with MTIOC6B signal 010011: Mask COMP1 level detection signal with MTIOC6C signal 010100: Mask COMP1 level detection signal with MTIOC6D signal 010101: Mask COMP1 level detection signal with MTIOC7A signal 010110: Mask COMP1 level detection signal with MTIOC7B signal 010111: Mask COMP1 level detection signal with MTIOC7C signal 011000: Mask COMP1 level detection signal with MTIOC7D signal 011001: Mask COMP1 level detection signal with MTIOC9A signal 011010: Mask COMP1 level detection signal with MTIOC9B signal 011011: Mask COMP1 level detection signal with MTIOC9C signal 011100: Mask COMP1 level detection signal with MTIOC9D signal 011101: Mask COMP1 level detection signal with GTIOC0A signal 011110: Mask COMP1 level detection signal with GTIOC0B signal 011111: Mask COMP1 level detection signal with GTIOC1A signal 100000: Mask COMP1 level detection signal with GTIOC1B signal 100001: Mask COMP1 level detection signal with GTIOC2A signal 100010: Mask COMP1 level detection signal with GTIOC2B signal 100011: Mask COMP1 level detection signal with GTIOC3A signal 100100: Mask COMP1 level detection signal with GTIOC3B signal 100101: Mask COMP1 level detection signal with GTIOC4A signal 100110: Mask COMP1 level detection signal with GTIOC4B signal 100111: Mask COMP1 level detection signal with GTIOC5A signal 101000: Mask COMP1 level detection signal with GTIOC5B signal 101001: Mask COMP1 level detection signal with GTIOC6A signal 101010: Mask COMP1 level detection signal with GTIOC6B signal 101011: Mask COMP1 level detection signal with GTIOC7A signal 101100: Mask COMP1 level detection signal with GTIOC7B signal Settings other than above are prohibited.	R/W*1
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

The IMCR10 register is an 8-bit readable/writable register used to mask the output disabling requests by the COMP1 level detection signal.

CMP1MS[5:0] Bits (COMP1 Mask Signal Select)

These bits are used to select a signal that masks the COMP1 level detection signal. When the selected pin drives high, the COMP1 level detection signal is masked.

23.2.66 Input Signal Mask Control Register 11 (IMCR11)

Address(es): POE.IMCR11 0009 E47Bh

	b7	b6	b5	b4	b3	b2	b1	b0
			CMP2MS[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	CMP2MS[5:0]	COMP2 Mask Signal Select	b5 b0 000000: Does not mask the COMP2 level detection signal 000001: Mask COMP2 level detection signal with MTIOC0A signal 000010: Mask COMP2 level detection signal with MTIOC0B signal 000011: Mask COMP2 level detection signal with MTIOC0C signal 000100: Mask COMP2 level detection signal with MTIOC0D signal 000101: Mask COMP2 level detection signal with MTIOC1A signal 000110: Mask COMP2 level detection signal with MTIOC1B signal 000111: Mask COMP2 level detection signal with MTIOC2A signal 001000: Mask COMP2 level detection signal with MTIOC2B signal 001001: Mask COMP2 level detection signal with MTIOC3A signal 001010: Mask COMP2 level detection signal with MTIOC3B signal 001011: Mask COMP2 level detection signal with MTIOC3C signal 001100: Mask COMP2 level detection signal with MTIOC3D signal 001101: Mask COMP2 level detection signal with MTIOC4A signal 001110: Mask COMP2 level detection signal with MTIOC4B signal 001111: Mask COMP2 level detection signal with MTIOC4C signal 010000: Mask COMP2 level detection signal with MTIOC4D signal 010001: Mask COMP2 level detection signal with MTIOC6A signal 010010: Mask COMP2 level detection signal with MTIOC6B signal 010011: Mask COMP2 level detection signal with MTIOC6C signal 010100: Mask COMP2 level detection signal with MTIOC6D signal 010101: Mask COMP2 level detection signal with MTIOC7A signal 010110: Mask COMP2 level detection signal with MTIOC7B signal 010111: Mask COMP2 level detection signal with MTIOC7C signal 011000: Mask COMP2 level detection signal with MTIOC7D signal 011001: Mask COMP2 level detection signal with MTIOC9A signal 011010: Mask COMP2 level detection signal with MTIOC9B signal 011011: Mask COMP2 level detection signal with MTIOC9C signal 011100: Mask COMP2 level detection signal with MTIOC9D signal 011101: Mask COMP2 level detection signal with GTIOC0A signal 011110: Mask COMP2 level detection signal with GTIOC0B signal 011111: Mask COMP2 level detection signal with GTIOC1A signal 100000: Mask COMP2 level detection signal with GTIOC1B signal 100001: Mask COMP2 level detection signal with GTIOC2A signal 100010: Mask COMP2 level detection signal with GTIOC2B signal 100011: Mask COMP2 level detection signal with GTIOC3A signal 100100: Mask COMP2 level detection signal with GTIOC3B signal 100101: Mask COMP2 level detection signal with GTIOC4A signal 100110: Mask COMP2 level detection signal with GTIOC4B signal 100111: Mask COMP2 level detection signal with GTIOC5A signal 101000: Mask COMP2 level detection signal with GTIOC5B signal 101001: Mask COMP2 level detection signal with GTIOC6A signal 101010: Mask COMP2 level detection signal with GTIOC6B signal 101011: Mask COMP2 level detection signal with GTIOC7A signal 101100: Mask COMP2 level detection signal with GTIOC7B signal Settings other than above are prohibited.	R/W*1
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

The IMCR11 register is an 8-bit readable/writable register used to mask the output disabling requests by the COMP2 level detection signal.

CMP2MS[5:0] Bits (COMP2 Mask Signal Select)

These bits are used to select a signal that masks the COMP2 level detection signal. When the selected pin drives high, the COMP2 level detection signal is masked.

23.2.67 Input Signal Mask Control Register 12 (IMCR12)

Address(es): POE.IMCR12 0009 E47Ch

	b7	b6	b5	b4	b3	b2	b1	b0
			CMP3MS[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	CMP3MS[5:0]	COMP3 Mask Signal Select	b5 b0 000000: Does not mask the COMP3 level detection signal 000001: Mask COMP3 level detection signal with MTIOC0A signal 000010: Mask COMP3 level detection signal with MTIOC0B signal 000011: Mask COMP3 level detection signal with MTIOC0C signal 000100: Mask COMP3 level detection signal with MTIOC0D signal 000101: Mask COMP3 level detection signal with MTIOC1A signal 000110: Mask COMP3 level detection signal with MTIOC1B signal 000111: Mask COMP3 level detection signal with MTIOC2A signal 001000: Mask COMP3 level detection signal with MTIOC2B signal 001001: Mask COMP3 level detection signal with MTIOC3A signal 001010: Mask COMP3 level detection signal with MTIOC3B signal 001011: Mask COMP3 level detection signal with MTIOC3C signal 001100: Mask COMP3 level detection signal with MTIOC3D signal 001101: Mask COMP3 level detection signal with MTIOC4A signal 001110: Mask COMP3 level detection signal with MTIOC4B signal 001111: Mask COMP3 level detection signal with MTIOC4C signal 010000: Mask COMP3 level detection signal with MTIOC4D signal 010001: Mask COMP3 level detection signal with MTIOC6A signal 010010: Mask COMP3 level detection signal with MTIOC6B signal 010011: Mask COMP3 level detection signal with MTIOC6C signal 010100: Mask COMP3 level detection signal with MTIOC6D signal 010101: Mask COMP3 level detection signal with MTIOC7A signal 010110: Mask COMP3 level detection signal with MTIOC7B signal 010111: Mask COMP3 level detection signal with MTIOC7C signal 011000: Mask COMP3 level detection signal with MTIOC7D signal 011001: Mask COMP3 level detection signal with MTIOC9A signal 011010: Mask COMP3 level detection signal with MTIOC9B signal 011011: Mask COMP3 level detection signal with MTIOC9C signal 011100: Mask COMP3 level detection signal with MTIOC9D signal 011101: Mask COMP3 level detection signal with GTIOC0A signal 011110: Mask COMP3 level detection signal with GTIOC0B signal 011111: Mask COMP3 level detection signal with GTIOC1A signal 100000: Mask COMP3 level detection signal with GTIOC1B signal 100001: Mask COMP3 level detection signal with GTIOC2A signal 100010: Mask COMP3 level detection signal with GTIOC2B signal 100011: Mask COMP3 level detection signal with GTIOC3A signal 100100: Mask COMP3 level detection signal with GTIOC3B signal 100101: Mask COMP3 level detection signal with GTIOC4A signal 100110: Mask COMP3 level detection signal with GTIOC4B signal 100111: Mask COMP3 level detection signal with GTIOC5A signal 101000: Mask COMP3 level detection signal with GTIOC5B signal 101001: Mask COMP3 level detection signal with GTIOC6A signal 101010: Mask COMP3 level detection signal with GTIOC6B signal 101011: Mask COMP3 level detection signal with GTIOC7A signal 101100: Mask COMP3 level detection signal with GTIOC7B signal Settings other than above are prohibited.	R/W*1
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

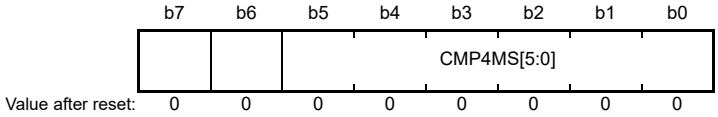
The IMCR12 register is an 8-bit readable/writable register used to mask the output disabling requests by the COMP3 level detection signal.

CMP3MS[5:0] Bits (COMP3 Mask Signal Select)

These bits are used to select a signal that masks the COMP3 level detection signal. When the selected pin drives high, the COMP3 level detection signal is masked.

23.2.68 Input Signal Mask Control Register 13 (IMCR13)

Address(es): POE.IMCR13 0009 E47Dh



Bit	Symbol	Bit Name	Description	R/W
b5 to b0	CMP4MS[5:0]	COMP4 Mask Signal Select	b5 b0 000000: Does not mask the COMP4 level detection signal 000001: Mask COMP4 level detection signal with MTIOC0A signal 000010: Mask COMP4 level detection signal with MTIOC0B signal 000011: Mask COMP4 level detection signal with MTIOC0C signal 000100: Mask COMP4 level detection signal with MTIOC0D signal 000101: Mask COMP4 level detection signal with MTIOC1A signal 000110: Mask COMP4 level detection signal with MTIOC1B signal 000111: Mask COMP4 level detection signal with MTIOC2A signal 001000: Mask COMP4 level detection signal with MTIOC2B signal 001001: Mask COMP4 level detection signal with MTIOC3A signal 001010: Mask COMP4 level detection signal with MTIOC3B signal 001011: Mask COMP4 level detection signal with MTIOC3C signal 001100: Mask COMP4 level detection signal with MTIOC3D signal 001101: Mask COMP4 level detection signal with MTIOC4A signal 001110: Mask COMP4 level detection signal with MTIOC4B signal 001111: Mask COMP4 level detection signal with MTIOC4C signal 010000: Mask COMP4 level detection signal with MTIOC4D signal 010001: Mask COMP4 level detection signal with MTIOC6A signal 010010: Mask COMP4 level detection signal with MTIOC6B signal 010011: Mask COMP4 level detection signal with MTIOC6C signal 010100: Mask COMP4 level detection signal with MTIOC6D signal 010101: Mask COMP4 level detection signal with MTIOC7A signal 010110: Mask COMP4 level detection signal with MTIOC7B signal 010111: Mask COMP4 level detection signal with MTIOC7C signal 011000: Mask COMP4 level detection signal with MTIOC7D signal 011001: Mask COMP4 level detection signal with MTIOC9A signal 011010: Mask COMP4 level detection signal with MTIOC9B signal 011011: Mask COMP4 level detection signal with MTIOC9C signal 011100: Mask COMP4 level detection signal with MTIOC9D signal 011101: Mask COMP4 level detection signal with GTIOC0A signal 011110: Mask COMP4 level detection signal with GTIOC0B signal 011111: Mask COMP4 level detection signal with GTIOC1A signal 100000: Mask COMP4 level detection signal with GTIOC1B signal 100001: Mask COMP4 level detection signal with GTIOC2A signal 100010: Mask COMP4 level detection signal with GTIOC2B signal 100011: Mask COMP4 level detection signal with GTIOC3A signal 100100: Mask COMP4 level detection signal with GTIOC3B signal 100101: Mask COMP4 level detection signal with GTIOC4A signal 100110: Mask COMP4 level detection signal with GTIOC4B signal 100111: Mask COMP4 level detection signal with GTIOC5A signal 101000: Mask COMP4 level detection signal with GTIOC5B signal 101001: Mask COMP4 level detection signal with GTIOC6A signal 101010: Mask COMP4 level detection signal with GTIOC6B signal 101011: Mask COMP4 level detection signal with GTIOC7A signal 101100: Mask COMP4 level detection signal with GTIOC7B signal Settings other than above are prohibited.	R/W*1
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

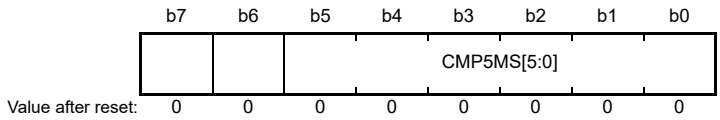
The IMCR13 register is an 8-bit readable/writable register used to mask the output disabling requests by the COMP4 level detection signal.

CMP4MS[5:0] Bits (COMP4 Mask Signal Select)

These bits are used to select a signal that masks the COMP4 level detection signal. When the selected pin drives high, the COMP4 level detection signal is masked.

23.2.69 Input Signal Mask Control Register 14 (IMCR14)

Address(es): POE.IMCR14 0009 E47Eh



Bit	Symbol	Bit Name	Description	R/W
b5 to b0	CMP5MS[5:0]	COMP5 Mask Signal Select	b5 b0 000000: Does not mask the COMP5 level detection signal 000001: Mask COMP5 level detection signal with MTIOC0A signal 000010: Mask COMP5 level detection signal with MTIOC0B signal 000011: Mask COMP5 level detection signal with MTIOC0C signal 000100: Mask COMP5 level detection signal with MTIOC0D signal 000101: Mask COMP5 level detection signal with MTIOC1A signal 000110: Mask COMP5 level detection signal with MTIOC1B signal 000111: Mask COMP5 level detection signal with MTIOC2A signal 001000: Mask COMP5 level detection signal with MTIOC2B signal 001001: Mask COMP5 level detection signal with MTIOC3A signal 001010: Mask COMP5 level detection signal with MTIOC3B signal 001011: Mask COMP5 level detection signal with MTIOC3C signal 001100: Mask COMP5 level detection signal with MTIOC3D signal 001101: Mask COMP5 level detection signal with MTIOC4A signal 001110: Mask COMP5 level detection signal with MTIOC4B signal 001111: Mask COMP5 level detection signal with MTIOC4C signal 010000: Mask COMP5 level detection signal with MTIOC4D signal 010001: Mask COMP5 level detection signal with MTIOC6A signal 010010: Mask COMP5 level detection signal with MTIOC6B signal 010011: Mask COMP5 level detection signal with MTIOC6C signal 010100: Mask COMP5 level detection signal with MTIOC6D signal 010101: Mask COMP5 level detection signal with MTIOC7A signal 010110: Mask COMP5 level detection signal with MTIOC7B signal 010111: Mask COMP5 level detection signal with MTIOC7C signal 011000: Mask COMP5 level detection signal with MTIOC7D signal 011001: Mask COMP5 level detection signal with MTIOC9A signal 011010: Mask COMP5 level detection signal with MTIOC9B signal 011011: Mask COMP5 level detection signal with MTIOC9C signal 011100: Mask COMP5 level detection signal with MTIOC9D signal 011101: Mask COMP5 level detection signal with GTIOC0A signal 011110: Mask COMP5 level detection signal with GTIOC0B signal 011111: Mask COMP5 level detection signal with GTIOC1A signal 100000: Mask COMP5 level detection signal with GTIOC1B signal 100001: Mask COMP5 level detection signal with GTIOC2A signal 100010: Mask COMP5 level detection signal with GTIOC2B signal 100011: Mask COMP5 level detection signal with GTIOC3A signal 100100: Mask COMP5 level detection signal with GTIOC3B signal 100101: Mask COMP5 level detection signal with GTIOC4A signal 100110: Mask COMP5 level detection signal with GTIOC4B signal 100111: Mask COMP5 level detection signal with GTIOC5A signal 101000: Mask COMP5 level detection signal with GTIOC5B signal 101001: Mask COMP5 level detection signal with GTIOC6A signal 101010: Mask COMP5 level detection signal with GTIOC6B signal 101011: Mask COMP5 level detection signal with GTIOC7A signal 101100: Mask COMP5 level detection signal with GTIOC7B signal Settings other than above are prohibited.	R/W*1
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

The IMCR13 register is an 8-bit readable/writable register used to mask the output disabling requests by the COMP5 level detection signal.

CMP5MS[5:0] Bits (COMP5 Mask Signal Select)

These bits are used to select a signal that masks the COMP5 level detection signal. When the selected pin drives high, the COMP5 level detection signal is masked.

23.3 Operation

The following shows the target pins and conditions for output disabling control.

(1) MTU3 pins (MTIOC3B, MTIOC3D)

When one of the following conditions is satisfied while the POECR2.MTU3BDZE bit is 1, the pins become high-impedance. Furthermore, when the POECR2.MTU3BDZE bit is 0 and the PMMCR1.MTU3BME and MTU3DME bits are 1 and when one of the following conditions is satisfied, the pins are switched to general I/O port pins.

- Operation for comparison of the output levels on the MTIOC3B and MTIOC3D signals
  When the OCSR1.OSF1 flag becomes 1 while the OCSR1.OCE1 bit is 1.
- SPOER setting
  When the SPOER.MTUCH34HIZ bit is set to 1.
- Conditions added by POECR4
  When the ICSR1.POE0F flag becomes 1 while the POECR4.IC1ADDMT34ZE bit is 1.
  When the ICSR2.POE4F flag becomes 1 while the POECR4.IC2ADDMT34ZE bit is 1.
  When the ICSR3.POE8F flag becomes 1 while the POECR4.IC3ADDMT34ZE bit and the ICSR3.POE8E bit are 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR4.IC4ADDMT34ZE bit and the ICSR4.POE10E bit are 1.
  When the ICSR5.POE11F flag becomes 1 while the POECR4.IC5ADDMT34ZE bit and the ICSR5.POE11E bit are 1.
  When the ICSR7.POE12F flag becomes 1 while the POECR4.IC6ADDMT34ZE bit and the ICSR7.POE12E bit are 1.
  When the ICSR8.POE9F flag becomes 1 while the POECR4.IC8ADDMT34ZE bit and the ICSR8.POE9E bit are 1.
- Comparator output detection
  When the POECMPFR.C0FLAG flag becomes 1 while the POECR4.CMADDMT34ZE bit is 1 and the POECMPEX0.POEREQ0 bit or the POECMPSEL.POEREQ0 bit is 1.
  When the POECMPFR.C1FLAG flag becomes 1 while the POECR4.CMADDMT34ZE bit is 1 and the POECMPEX0.POEREQ1 bit or the POECMPSEL.POEREQ1 bit is 1.
  When the POECMPFR.C2FLAG flag becomes 1 while the POECR4.CMADDMT34ZE bit is 1 and the POECMPEX0.POEREQ2 bit or the POECMPSEL.POEREQ2 bit is 1.
  When the POECMPFR.C3FLAG flag becomes 1 while the POECR4.CMADDMT34ZE bit is 1 and the POECMPEX0.POEREQ3 bit or the POECMPSEL.POEREQ3 bit is 1.
  When the POECMPFR.C4FLAG flag becomes 1 while the POECR4.CMADDMT34ZE bit is 1 and the POECMPEX0.POEREQ4 bit or the POECMPSEL.POEREQ4 bit is 1.
  When the POECMPFR.C5FLAG flag becomes 1 while the POECR4.CMADDMT34ZE bit is 1 and the POECMPEX0.POEREQ5 bit or the POECMPSEL.POEREQ5 bit is 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(2) MTU4 pins (MTIOC4A, MTIOC4C)

When one of the following conditions is satisfied while the POECR2.MTU4ACZE bit is 1, the pins become high-impedance. Furthermore, when the POECR2.MTU4ACZE bit is 0 and the PMMCR1.MTU4AME and MTU4CME bits are 1 and when one of the following conditions is satisfied, the pins are switched to general I/O port pins.

- Operation for comparison of the output levels on the MTIOC4A and MTIOC4C signals
  When the OCSR1.OSF1 flag becomes 1 while the OCSR1.OCE1 bit is 1.
- SPOER setting
  When the SPOER.MTUCH34HIZ bit is set to 1.
- Conditions added by POECR4
  When the ICSR1.POE0F flag becomes 1 while the POECR4.IC1ADDMT34ZE bit is 1.

When the ICSR2.POE4F flag becomes 1 while the POECR4.IC2ADDMT34ZE bit is 1.
When the ICSR3.POE8F flag becomes 1 while the POECR4.IC3ADDMT34ZE bit and the ICSR3.POE8E bit are 1.
When the ICSR4.POE10F flag becomes 1 while the POECR4.IC4ADDMT34ZE bit and the ICSR4.POE10E bit are 1.
When the ICSR5.POE11F flag becomes 1 while the POECR4.IC5ADDMT34ZE bit and the ICSR5.POE11E bit are 1.
When the ICSR7.POE12F flag becomes 1 while the POECR4.IC6ADDMT34ZE bit and the ICSR7.POE12E bit are 1.
When the ICSR8.POE9F flag becomes 1 while the POECR4.IC8ADDMT34ZE bit and the ICSR8.POE9E bit are 1.

- Comparator output detection
  When the POECMPFR.C0FLAG flag becomes 1 while the POECR4.CMADDMT34ZE bit is 1 and the POECMPEX0.POEREQ0 bit or the POECMPSEL.POEREQ0 bit is 1.
  When the POECMPFR.C1FLAG flag becomes 1 while the POECR4.CMADDMT34ZE bit is 1 and the POECMPEX0.POEREQ1 bit or the POECMPSEL.POEREQ1 bit is 1.
  When the POECMPFR.C2FLAG flag becomes 1 while the POECR4.CMADDMT34ZE bit is 1 and the POECMPEX0.POEREQ2 bit or the POECMPSEL.POEREQ2 bit is 1.
  When the POECMPFR.C3FLAG flag becomes 1 while the POECR4.CMADDMT34ZE bit is 1 and the POECMPEX0.POEREQ3 bit or the POECMPSEL.POEREQ3 bit is 1.
  When the POECMPFR.C4FLAG flag becomes 1 while the POECR4.CMADDMT34ZE bit is 1 and the POECMPEX0.POEREQ4 bit or the POECMPSEL.POEREQ4 bit is 1.
  When the POECMPFR.C5FLAG flag becomes 1 while the POECR4.CMADDMT34ZE bit is 1 and the POECMPEX0.POEREQ5 bit or the POECMPSEL.POEREQ5 bit is 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(3) MTU4 pins (MTIOC4B, MTIOC4D)

When one of the following conditions is satisfied while the POECR2.MTU4BDZE bit is 1, the pins become high-impedance. Furthermore, when the POECR2.MTU4BDZE bit is 0 and the PMMCR1.MTU4BME and MTU4DME bits are 1 and when one of the following conditions is satisfied, the pins are switched to general I/O port pins.

- Operation for comparison of the output levels on the MTIOC4B and MTIOC4D signals
  When the OCSR1.OSF1 flag becomes 1 while the OCSR1.OCE1 bit is 1.
- SPOER setting
  When the SPOER.MTUCH34HIZ bit is set to 1.
- Conditions added by POECR4
  When the ICSR1.POE0F flag becomes 1 while the POECR4.IC1ADDMT34ZE bit is 1.
  When the ICSR2.POE4F flag becomes 1 while the POECR4.IC2ADDMT34ZE bit is 1.
  When the ICSR3.POE8F flag becomes 1 while the POECR4.IC3ADDMT34ZE bit and the ICSR3.POE8E bit are 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR4.IC4ADDMT34ZE bit and the ICSR4.POE10E bit are 1.
  When the ICSR5.POE11F flag becomes 1 while the POECR4.IC5ADDMT34ZE bit and the ICSR5.POE11E bit are 1.
  When the ICSR7.POE12F flag becomes 1 while the POECR4.IC6ADDMT34ZE bit and the ICSR7.POE12E bit are 1.
  When the ICSR8.POE9F flag becomes 1 while the POECR4.IC8ADDMT34ZE bit and the ICSR8.POE9E bit are 1.
- Comparator output detection
  When the POECMPFR.C0FLAG flag becomes 1 while the POECR4.CMADDMT34ZE bit is 1 and the POECMPEX0.POEREQ0 bit or the POECMPSEL.POEREQ0 bit is 1.
  When the POECMPFR.C1FLAG flag becomes 1 while the POECR4.CMADDMT34ZE bit is 1 and the POECMPEX0.POEREQ1 bit or the POECMPSEL.POEREQ1 bit is 1.

When the POECMPFR.C2FLAG flag becomes 1 while the POECR4.CMADDMT34ZE bit is 1 and the POECMPEX0.POEREQ2 bit or the POECMPSEL.POEREQ2 bit is 1.
When the POECMPFR.C3FLAG flag becomes 1 while the POECR4.CMADDMT34ZE bit is 1 and the POECMPEX0.POEREQ3 bit or the POECMPSEL.POEREQ3 bit is 1.
When the POECMPFR.C4FLAG flag becomes 1 while the POECR4.CMADDMT34ZE bit is 1 and the POECMPEX0.POEREQ4 bit or the POECMPSEL.POEREQ4 bit is 1.
When the POECMPFR.C5FLAG flag becomes 1 while the POECR4.CMADDMT34ZE bit is 1 and the POECMPEX0.POEREQ5 bit or the POECMPSEL.POEREQ5 bit is 1.

- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(4) MTU6 pins (MTIOC6B, MTIOC6D)

When one of the following conditions is satisfied while the POECR2.MTU6BDZE bit is 1, the pins become high-impedance. Furthermore, when the POECR2.MTU6BDZE bit is 0 and the PMMCR1.MTU6BME and MTU6DME bits are 1 and when one of the following conditions is satisfied, the pins are switched to general I/O port pins.

- Operation for comparison of the output levels on the MTIOC6B and MTIOC6D signals
  When the OCSR2.OSF2 flag becomes 1 while the OCSR2.OCE2 bit is 1.
- SPOER setting
  When the SPOER.MTUCH67HIZ bit is set to 1.
- Conditions added by POECR4B
  When the ICSR1.POE0F flag becomes 1 while the POECR4B.IC1ADDMT67ZE bit is 1.
  When the ICSR2.POE4F flag becomes 1 while the POECR4B.IC2ADDMT67ZE bit is 1.
  When the ICSR3.POE8F flag becomes 1 while the POECR4B.IC3ADDMT67ZE bit and the ICSR3.POE8E bit are 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR4B.IC4ADDMT67ZE bit and the ICSR4.POE10E bit are 1.
  When the ICSR5.POE11F flag becomes 1 while the POECR4B.IC5ADDMT67ZE bit and the ICSR5.POE11E bit are 1.
  When the ICSR7.POE12F flag becomes 1 while the POECR4B.IC6ADDMT67ZE bit and the ICSR7.POE12E bit are 1.
  When the ICSR8.POE9F flag becomes 1 while the POECR4B.IC8ADDMT67ZE bit and the ICSR8.POE9E bit are 1.
- Comparator output detection
  When the POECMPFR.C0FLAG flag becomes 1 while the POECR4B.CMADDMT67ZE bit is 1 and the POECMPEX1.POEREQ0 bit or the POECMPSEL.POEREQ0 bit is 1.
  When the POECMPFR.C1FLAG flag becomes 1 while the POECR4B.CMADDMT67ZE bit is 1 and the POECMPEX1.POEREQ1 bit or the POECMPSEL.POEREQ1 bit is 1.
  When the POECMPFR.C2FLAG flag becomes 1 while the POECR4B.CMADDMT67ZE bit is 1 and the POECMPEX1.POEREQ2 bit or the POECMPSEL.POEREQ2 bit is 1.
  When the POECMPFR.C3FLAG flag becomes 1 while the POECR4B.CMADDMT67ZE bit is 1 and the POECMPEX1.POEREQ3 bit or the POECMPSEL.POEREQ3 bit is 1.
  When the POECMPFR.C4FLAG flag becomes 1 while the POECR4B.CMADDMT67ZE bit is 1 and the POECMPEX1.POEREQ4 bit or the POECMPSEL.POEREQ4 bit is 1.
  When the POECMPFR.C5FLAG flag becomes 1 while the POECR4B.CMADDMT67ZE bit is 1 and the POECMPEX1.POEREQ5 bit or the POECMPSEL.POEREQ5 bit is 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(5) MTU7 pins (MTIOC7A, MTIOC7C)

When one of the following conditions is satisfied while the POECR2.MTU7ACZE bit is 1, the pins become high-impedance. Furthermore, when the POECR2.MTU7ACZE bit is 0 and the PMMCR1.MTU7AME and MTU7CME bits are 1 and when one of the following conditions is satisfied, the pins are switched to general I/O port pins.

- Operation for comparison of the output levels on the MTIOC7A and MTIOC7C signals
  When the OCSR2.OSF2 flag becomes 1 while the OCSR2.OCE2 bit is 1.
- SPOER setting
  When the SPOER.MTUCH67HIZ bit is set to 1.
- Conditions added by POECR4B
  When the ICSR1.POE0F flag becomes 1 while the POECR4B.IC1ADDMT67ZE bit is 1.
  When the ICSR2.POE4F flag becomes 1 while the POECR4B.IC2ADDMT67ZE bit is 1.
  When the ICSR3.POE8F flag becomes 1 while the POECR4B.IC3ADDMT67ZE bit and the ICSR3.POE8E bit are 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR4B.IC4ADDMT67ZE bit and the ICSR4.POE10E bit are 1.
  When the ICSR5.POE11F flag becomes 1 while the POECR4B.IC5ADDMT67ZE bit and the ICSR5.POE11E bit are 1.
  When the ICSR7.POE12F flag becomes 1 while the POECR4B.IC6ADDMT67ZE bit and the ICSR7.POE12E bit are 1.
  When the ICSR8.POE9F flag becomes 1 while the POECR4B.IC8ADDMT67ZE bit and the ICSR8.POE9E bit are 1.
- Comparator output detection
  When the POECMPFR.C0FLAG flag becomes 1 while the POECR4B.CMADDMT67ZE bit is 1 and the POECMPEX1.POEREQ0 bit or the POECMPSEL.POEREQ0 bit is 1.
  When the POECMPFR.C1FLAG flag becomes 1 while the POECR4B.CMADDMT67ZE bit is 1 and the POECMPEX1.POEREQ1 bit or the POECMPSEL.POEREQ1 bit is 1.
  When the POECMPFR.C2FLAG flag becomes 1 while the POECR4B.CMADDMT67ZE bit is 1 and the POECMPEX1.POEREQ2 bit or the POECMPSEL.POEREQ2 bit is 1.
  When the POECMPFR.C3FLAG flag becomes 1 while the POECR4B.CMADDMT67ZE bit is 1 and the POECMPEX1.POEREQ3 bit or the POECMPSEL.POEREQ3 bit is 1.
  When the POECMPFR.C4FLAG flag becomes 1 while the POECR4B.CMADDMT67ZE bit is 1 and the POECMPEX1.POEREQ4 bit or the POECMPSEL.POEREQ4 bit is 1.
  When the POECMPFR.C5FLAG flag becomes 1 while the POECR4B.CMADDMT67ZE bit is 1 and the POECMPEX1.POEREQ5 bit or the POECMPSEL.POEREQ5 bit is 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(6) MTU7 pins (MTIOC7B, MTIOC7D)

When one of the following conditions is satisfied while the POECR2.MTU7BDZE bit is 1, the pins become high-impedance. Furthermore, when the POECR2.MTU7BDZE bit is 0 and the PMMCR1.MTU7BME and MTU7DME bits are 1 and when one of the following conditions is satisfied, the pins are switched to general I/O port pins.

- Operation for comparison of the output levels on the MTIOC7B and MTIOC7D signals
  When the OCSR2.OSF2 flag becomes 1 while the OCSR2.OCE2 bit is 1.
- SPOER setting
  When the SPOER.MTUCH67HIZ bit is set to 1.
- Conditions added by POECR4B
  When the ICSR1.POE0F flag becomes 1 while the POECR4B.IC1ADDMT67ZE bit is 1.
  When the ICSR2.POE4F flag becomes 1 while the POECR4B.IC2ADDMT67ZE bit is 1.

When the ICSR3.POE8F flag becomes 1 while the POECR4B.IC3ADDMT67ZE bit and the ICSR3.POE8E bit are 1.
When the ICSR4.POE10F flag becomes 1 while the POECR4B.IC4ADDMT67ZE bit and the ICSR4.POE10E bit are 1.
When the ICSR5.POE11F flag becomes 1 while the POECR4B.IC5ADDMT67ZE bit and the ICSR5.POE11E bit are 1.
When the ICSR7.POE12F flag becomes 1 while the POECR4B.IC6ADDMT67ZE bit and the ICSR7.POE12E bit are 1.
When the ICSR8.POE9F flag becomes 1 while the POECR4B.IC8ADDMT67ZE bit and the ICSR8.POE9E bit are 1.

- Comparator output detection
  When the POECMPFR.C0FLAG flag becomes 1 while the POECR4B.CMADDMT67ZE bit is 1 and the POECMPEX1.POEREQ0 bit or the POECMPSEL.POEREQ0 bit is 1.
  When the POECMPFR.C1FLAG flag becomes 1 while the POECR4B.CMADDMT67ZE bit is 1 and the POECMPEX1.POEREQ1 bit or the POECMPSEL.POEREQ1 bit is 1.
  When the POECMPFR.C2FLAG flag becomes 1 while the POECR4B.CMADDMT67ZE bit is 1 and the POECMPEX1.POEREQ2 bit or the POECMPSEL.POEREQ2 bit is 1.
  When the POECMPFR.C3FLAG flag becomes 1 while the POECR4B.CMADDMT67ZE bit is 1 and the POECMPEX1.POEREQ3 bit or the POECMPSEL.POEREQ3 bit is 1.
  When the POECMPFR.C4FLAG flag becomes 1 while the POECR4B.CMADDMT67ZE bit is 1 and the POECMPEX1.POEREQ4 bit or the POECMPSEL.POEREQ4 bit is 1.
  When the POECMPFR.C5FLAG flag becomes 1 while the POECR4B.CMADDMT67ZE bit is 1 and the POECMPEX1.POEREQ5 bit or the POECMPSEL.POEREQ5 bit is 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(7) MTU0 pin (MTIOC0A)

When one of the following conditions is satisfied while the POECR1.MTU0AZE bit is 1, the pins become high-impedance. Furthermore, when the POECR1.MTU0AZE bit is 0 and the PMMCR0.MTU0AME bit is 1 and when one of the following conditions is satisfied, the pins are switched to general I/O port pins.

- SPOER setting
  When the SPOER.MTUCH0HIZ bit is set to 1.
- Conditions added by POECR5
  When the ICSR1.POE0F flag becomes 1 while the POECR5.IC1ADDMT0ZE bit is 1.
  When the ICSR2.POE4F flag becomes 1 while the POECR5.IC2ADDMT0ZE bit is 1.
  When the ICSR3.POE8F flag becomes 1 while the POECR5.IC3ADDMT0ZE bit and the ICSR3.POE8E bit are 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR5.IC4ADDMT0ZE bit and the ICSR4.POE10E bit are 1.
  When the ICSR5.POE11F flag becomes 1 while the POECR5.IC5ADDMT0ZE bit and the ICSR5.POE11E bit are 1.
  When the ICSR7.POE12F flag becomes 1 while the POECR5.IC6ADDMT0ZE bit and the ICSR7.POE12E bit are 1.
  When the ICSR8.POE9F flag becomes 1 while the POECR5.IC8ADDMT0ZE bit and the ICSR8.POE9E bit are 1.
- Comparator output detection
  When the POECMPFR.C0FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the POECMPEX2.POEREQ0 bit or the POECMPSEL.POEREQ0 bit is 1.
  When the POECMPFR.C1FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the POECMPEX2.POEREQ1 bit or the POECMPSEL.POEREQ1 bit is 1.
  When the POECMPFR.C2FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the

POECMPEX2.POEREQ2 bit or the POECMPSEL.POEREQ2 bit is 1.
When the POECMPFR.C3FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the POECMPEX2.POEREQ3 bit or the POECMPSEL.POEREQ3 bit is 1.
When the POECMPFR.C4FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the POECMPEX2.POEREQ4 bit or the POECMPSEL.POEREQ4 bit is 1.
When the POECMPFR.C5FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the POECMPEX2.POEREQ5 bit or the POECMPSEL.POEREQ5 bit is 1.

- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(8) MTU0 pin (MTIOC0B)

When one of the following conditions is satisfied while the POECR1.MTU0BZE bit is 1, the pins become high-impedance. Furthermore, when the POECR1.MTU0BZE bit is 0 and the PMMCR0.MTU0BME bit is 1 and when one of the following conditions is satisfied, the pins are switched to general I/O port pins.

- SPOER setting
  When the SPOER.MTUCH0HIZ bit is set to 1.
- Conditions added by POECR5
  When the ICSR1.POE0F flag becomes 1 while the POECR5.IC1ADDMT0ZE bit is 1.
  When the ICSR2.POE4F flag becomes 1 while the POECR5.IC2ADDMT0ZE bit is 1.
  When the ICSR3.POE8F flag becomes 1 while the POECR5.IC3ADDMT0ZE bit and the ICSR3.POE8E bit are 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR5.IC4ADDMT0ZE bit and the ICSR4.POE10E bit are 1.
  When the ICSR5.POE11F flag becomes 1 while the POECR5.IC5ADDMT0ZE bit and the ICSR5.POE11E bit are 1.
  When the ICSR7.POE12F flag becomes 1 while the POECR5.IC6ADDMT0ZE bit and the ICSR7.POE12E bit are 1.
  When the ICSR8.POE9F flag becomes 1 while the POECR5.IC8ADDMT0ZE bit and the ICSR8.POE9E bit are 1.
- Comparator output detection
  When the POECMPFR.C0FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the POECMPEX2.POEREQ0 bit or the POECMPSEL.POEREQ0 bit is 1.
  When the POECMPFR.C1FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the POECMPEX2.POEREQ1 bit or the POECMPSEL.POEREQ1 bit is 1.
  When the POECMPFR.C2FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the POECMPEX2.POEREQ2 bit or the POECMPSEL.POEREQ2 bit is 1.
  When the POECMPFR.C3FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the POECMPEX2.POEREQ3 bit or the POECMPSEL.POEREQ3 bit is 1.
  When the POECMPFR.C4FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the POECMPEX2.POEREQ4 bit or the POECMPSEL.POEREQ4 bit is 1.
  When the POECMPFR.C5FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the POECMPEX2.POEREQ5 bit or the POECMPSEL.POEREQ5 bit is 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(9) MTU0 pin (MTIOC0C)

When one of the following conditions is satisfied while the POECR1.MTU0CZE bit is 1, the pins become high-impedance. Furthermore, when the POECR1.MTU0CZE bit is 0 and the PMMCR0.MTU0CME bit is 1 and when one of the following conditions is satisfied, the pins are switched to general I/O port pins.

- SPOER setting
  When the SPOER.MTUCH0HIZ bit is set to 1.

- Conditions added by POECR5
  When the ICSR1.POE0F flag becomes 1 while the POECR5.IC1ADDMT0ZE bit is 1.
  When the ICSR2.POE4F flag becomes 1 while the POECR5.IC2ADDMT0ZE bit is 1.
  When the ICSR3.POE8F flag becomes 1 while the POECR5.IC3ADDMT0ZE bit and the ICSR3.POE8E bit are 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR5.IC4ADDMT0ZE bit and the ICSR4.POE10E bit are 1.
  When the ICSR5.POE11F flag becomes 1 while the POECR5.IC5ADDMT0ZE bit and the ICSR5.POE11E bit are 1.
  When the ICSR7.POE12F flag becomes 1 while the POECR5.IC6ADDMT0ZE bit and the ICSR7.POE12E bit are 1.
  When the ICSR8.POE9F flag becomes 1 while the POECR5.IC8ADDMT0ZE bit and the ICSR8.POE9E bit are 1.
- Comparator output detection
  When the POECMPFR.C0FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the POECMPEX2.POEREQ0 bit or the POECMPSEL.POEREQ0 bit is 1.
  When the POECMPFR.C1FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the POECMPEX2.POEREQ1 bit or the POECMPSEL.POEREQ1 bit is 1.
  When the POECMPFR.C2FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the POECMPEX2.POEREQ2 bit or the POECMPSEL.POEREQ2 bit is 1.
  When the POECMPFR.C3FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the POECMPEX2.POEREQ3 bit or the POECMPSEL.POEREQ3 bit is 1.
  When the POECMPFR.C4FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the POECMPEX2.POEREQ4 bit or the POECMPSEL.POEREQ4 bit is 1.
  When the POECMPFR.C5FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the POECMPEX2.POEREQ5 bit or the POECMPSEL.POEREQ5 bit is 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(10) MTU0 pin (MTIOC0D)

When one of the following conditions is satisfied while the POECR1.MTU0DZE bit is 1, the pins become high-impedance. Furthermore, when the POECR1.MTU0DZE bit is 0 and the PMMCR0.MTU0DME bit is 1 and when one of the following conditions is satisfied, the pins are switched to general I/O port pins.

- SPOER setting
  When the SPOER.MTUCH0HIZ bit is set to 1.
- Conditions added by POECR5
  When the ICSR1.POE0F flag becomes 1 while the POECR5.IC1ADDMT0ZE bit is 1.
  When the ICSR2.POE4F flag becomes 1 while the POECR5.IC2ADDMT0ZE bit is 1.
  When the ICSR3.POE8F flag becomes 1 while the POECR5.IC3ADDMT0ZE bit and the ICSR3.POE8E bit are 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR5.IC4ADDMT0ZE bit and the ICSR4.POE10E bit are 1.
  When the ICSR5.POE11F flag becomes 1 while the POECR5.IC5ADDMT0ZE bit and the ICSR5.POE11E bit are 1.
  When the ICSR7.POE12F flag becomes 1 while the POECR5.IC6ADDMT0ZE bit and the ICSR7.POE12E bit are 1.
  When the ICSR8.POE9F flag becomes 1 while the POECR5.IC8ADDMT0ZE bit and the ICSR8.POE9E bit are 1.
- Comparator output detection
  When the POECMPFR.C0FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the POECMPEX2.POEREQ0 bit or the POECMPSEL.POEREQ0 bit is 1.
  When the POECMPFR.C1FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the POECMPEX2.POEREQ1 bit or the POECMPSEL.POEREQ1 bit is 1.

When the POECMPFR.C2FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the POECMPEX2.POEREQ2 bit or the POECMPSEL.POEREQ2 bit is 1.
When the POECMPFR.C3FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the POECMPEX2.POEREQ3 bit or the POECMPSEL.POEREQ3 bit is 1.
When the POECMPFR.C4FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the POECMPEX2.POEREQ4 bit or the POECMPSEL.POEREQ4 bit is 1.
When the POECMPFR.C5FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the POECMPEX2.POEREQ5 bit or the POECMPSEL.POEREQ5 bit is 1.

- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(11) MTU9 pin (MTIOC9A)

When one of the following conditions is satisfied while the POECR7.MTU9AZE bit is 1, the pins become high-impedance. Furthermore, when the POECR7.MTU9AZE bit is 0 and the PMMCR0.MTU9AME bit is 1 and when one of the following conditions is satisfied, the pins are switched to general I/O port pins.

- SPOER setting
  When the SPOER.MTUCH9HIZ bit is set to 1.
- Conditions added by POECR8
  When the ICSR1.POE0F flag becomes 1 while the POECR8.IC1ADDMT9ZE bit is 1.
  When the ICSR2.POE4F flag becomes 1 while the POECR8.IC2ADDMT9ZE bit is 1.
  When the ICSR3.POE8F flag becomes 1 while the POECR8.IC3ADDMT9ZE bit and the ICSR3.POE8E bit are 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR8.IC4ADDMT9ZE bit and the ICSR4.POE10E bit are 1.
  When the ICSR5.POE11F flag becomes 1 while the POECR8.IC5ADDMT9ZE bit and the ICSR5.POE11E bit are 1.
  When the ICSR7.POE12F flag becomes 1 while the POECR8.IC6ADDMT9ZE bit and the ICSR7.POE12E bit are 1.
  When the ICSR8.POE9F flag becomes 1 while the POECR8.IC8ADDMT9ZE bit and the ICSR8.POE9E bit are 1.
- Comparator output detection
  When the POECMPFR.C0FLAG flag becomes 1 while the POECR8.CMADDMT9ZE bit is 1 and the POECMPEX3.POEREQ0 bit or the POECMPSEL.POEREQ0 bit is 1.
  When the POECMPFR.C1FLAG flag becomes 1 while the POECR8.CMADDMT9ZE bit is 1 and the POECMPEX3.POEREQ1 bit or the POECMPSEL.POEREQ1 bit is 1.
  When the POECMPFR.C2FLAG flag becomes 1 while the POECR8.CMADDMT9ZE bit is 1 and the POECMPEX3.POEREQ2 bit or the POECMPSEL.POEREQ2 bit is 1.
  When the POECMPFR.C3FLAG flag becomes 1 while the POECR8.CMADDMT9ZE bit is 1 and the POECMPEX3.POEREQ3 bit or the POECMPSEL.POEREQ3 bit is 1.
  When the POECMPFR.C4FLAG flag becomes 1 while the POECR8.CMADDMT9ZE bit is 1 and the POECMPEX3.POEREQ4 bit or the POECMPSEL.POEREQ4 bit is 1.
  When the POECMPFR.C5FLAG flag becomes 1 while the POECR8.CMADDMT9ZE bit is 1 and the POECMPEX3.POEREQ5 bit or the POECMPSEL.POEREQ5 bit is 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(12) MTU9 pin (MTIOC9B)

When one of the following conditions is satisfied while the POECR7.MTU9BZE bit is 1, the pins become high-impedance. Furthermore, when the POECR7.MTU9BZE bit is 0 and the PMMCR0.MTU9BME bit is 1 and when one of the following conditions is satisfied, the pins are switched to general I/O port pins.

- SPOER setting

When the SPOER.MTUCH9HIZ bit is set to 1.

- Conditions added by POECR8
  When the ICSR1.POE0F flag becomes 1 while the POECR8.IC1ADDMT9ZE bit is 1.
  When the ICSR2.POE4F flag becomes 1 while the POECR8.IC2ADDMT9ZE bit is 1.
  When the ICSR3.POE8F flag becomes 1 while the POECR8.IC3ADDMT9ZE bit and the ICSR3.POE8E bit are 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR8.IC4ADDMT9ZE bit and the ICSR4.POE10E bit are 1.
  When the ICSR5.POE11F flag becomes 1 while the POECR8.IC5ADDMT9ZE bit and the ICSR5.POE11E bit are 1.
  When the ICSR7.POE12F flag becomes 1 while the POECR8.IC6ADDMT9ZE bit and the ICSR7.POE12E bit are 1.
  When the ICSR8.POE9F flag becomes 1 while the POECR8.IC8ADDMT9ZE bit and the ICSR8.POE9E bit are 1.
- Comparator output detection
  When the POECMPFR.C0FLAG flag becomes 1 while the POECR8.CMADDMT9ZE bit is 1 and the POECMPEX3.POEREQ0 bit or the POECMPSEL.POEREQ0 bit is 1.
  When the POECMPFR.C1FLAG flag becomes 1 while the POECR8.CMADDMT9ZE bit is 1 and the POECMPEX3.POEREQ1 bit or the POECMPSEL.POEREQ1 bit is 1.
  When the POECMPFR.C2FLAG flag becomes 1 while the POECR8.CMADDMT9ZE bit is 1 and the POECMPEX3.POEREQ2 bit or the POECMPSEL.POEREQ2 bit is 1.
  When the POECMPFR.C3FLAG flag becomes 1 while the POECR8.CMADDMT9ZE bit is 1 and the POECMPEX3.POEREQ3 bit or the POECMPSEL.POEREQ3 bit is 1.
  When the POECMPFR.C4FLAG flag becomes 1 while the POECR8.CMADDMT9ZE bit is 1 and the POECMPEX3.POEREQ4 bit or the POECMPSEL.POEREQ4 bit is 1.
  When the POECMPFR.C5FLAG flag becomes 1 while the POECR8.CMADDMT9ZE bit is 1 and the POECMPEX3.POEREQ5 bit or the POECMPSEL.POEREQ5 bit is 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(13) MTU9 pin (MTIOC9C)

When one of the following conditions is satisfied while the POECR7.MTU9CZE bit is 1, the pins become high-impedance. Furthermore, when the POECR7.MTU9CZE bit is 0 and the PMMCR0.MTU9CME bit is 1 and when one of the following conditions is satisfied, the pins are switched to general I/O port pins.

- SPOER setting
  When the SPOER.MTUCH9HIZ bit is set to 1.
- Conditions added by POECR8
  When the ICSR1.POE0F flag becomes 1 while the POECR8.IC1ADDMT9ZE bit is 1.
  When the ICSR2.POE4F flag becomes 1 while the POECR8.IC2ADDMT9ZE bit is 1.
  When the ICSR3.POE8F flag becomes 1 while the POECR8.IC3ADDMT9ZE bit and the ICSR3.POE8E bit are 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR8.IC4ADDMT9ZE bit and the ICSR4.POE10E bit are 1.
  When the ICSR5.POE11F flag becomes 1 while the POECR8.IC5ADDMT9ZE bit and the ICSR5.POE11E bit are 1.
  When the ICSR7.POE12F flag becomes 1 while the POECR8.IC6ADDMT9ZE bit and the ICSR7.POE12E bit are 1.
  When the ICSR8.POE9F flag becomes 1 while the POECR8.IC8ADDMT9ZE bit and the ICSR8.POE9E bit are 1.
- Comparator output detection
  When the POECMPFR.C0FLAG flag becomes 1 while the POECR8.CMADDMT9ZE bit is 1 and the POECMPEX3.POEREQ0 bit or the POECMPSEL.POEREQ0 bit is 1.
  When the POECMPFR.C1FLAG flag becomes 1 while the POECR8.CMADDMT9ZE bit is 1 and the

POECMPEX3.POEREQ1 bit or the POECMPSEL.POEREQ1 bit is 1.
When the POECMPFR.C2FLAG flag becomes 1 while the POECR8.CMADDMT9ZE bit is 1 and the POECMPEX3.POEREQ2 bit or the POECMPSEL.POEREQ2 bit is 1.
When the POECMPFR.C3FLAG flag becomes 1 while the POECR8.CMADDMT9ZE bit is 1 and the POECMPEX3.POEREQ3 bit or the POECMPSEL.POEREQ3 bit is 1.
When the POECMPFR.C4FLAG flag becomes 1 while the POECR8.CMADDMT9ZE bit is 1 and the POECMPEX3.POEREQ4 bit or the POECMPSEL.POEREQ4 bit is 1.
When the POECMPFR.C5FLAG flag becomes 1 while the POECR8.CMADDMT9ZE bit is 1 and the POECMPEX3.POEREQ5 bit or the POECMPSEL.POEREQ5 bit is 1.

- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(14) MTU9 pin (MTIOC9D)

When one of the following conditions is satisfied while the POECR7.MTU9DZE bit is 1, the pins become high-impedance. Furthermore, when the POECR7.MTU9DZE bit is 0 and the PMMCR0.MTU9DME bit is 1 and when one of the following conditions is satisfied, the pins are switched to general I/O port pins.

- SPOER setting
  When the SPOER.MTUCH9HIZ bit is set to 1.
- Conditions added by POECR8
  When the ICSR1.POE0F flag becomes 1 while the POECR8.IC1ADDMT9ZE bit is 1.
  When the ICSR2.POE4F flag becomes 1 while the POECR8.IC2ADDMT9ZE bit is 1.
  When the ICSR3.POE8F flag becomes 1 while the POECR8.IC3ADDMT9ZE bit and the ICSR3.POE8E bit are 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR8.IC4ADDMT9ZE bit and the ICSR4.POE10E bit are 1.
  When the ICSR5.POE11F flag becomes 1 while the POECR8.IC5ADDMT9ZE bit and the ICSR5.POE11E bit are 1.
  When the ICSR7.POE12F flag becomes 1 while the POECR8.IC6ADDMT9ZE bit and the ICSR7.POE12E bit are 1.
  When the ICSR8.POE9F flag becomes 1 while the POECR8.IC8ADDMT9ZE bit and the ICSR8.POE9E bit are 1.
- Comparator output detection
  When the POECMPFR.C0FLAG flag becomes 1 while the POECR8.CMADDMT9ZE bit is 1 and the POECMPEX3.POEREQ0 bit or the POECMPSEL.POEREQ0 bit is 1.
  When the POECMPFR.C1FLAG flag becomes 1 while the POECR8.CMADDMT9ZE bit is 1 and the POECMPEX3.POEREQ1 bit or the POECMPSEL.POEREQ1 bit is 1.
  When the POECMPFR.C2FLAG flag becomes 1 while the POECR8.CMADDMT9ZE bit is 1 and the POECMPEX3.POEREQ2 bit or the POECMPSEL.POEREQ2 bit is 1.
  When the POECMPFR.C3FLAG flag becomes 1 while the POECR8.CMADDMT9ZE bit is 1 and the POECMPEX3.POEREQ3 bit or the POECMPSEL.POEREQ3 bit is 1.
  When the POECMPFR.C4FLAG flag becomes 1 while the POECR8.CMADDMT9ZE bit is 1 and the POECMPEX3.POEREQ4 bit or the POECMPSEL.POEREQ4 bit is 1.
  When the POECMPFR.C5FLAG flag becomes 1 while the POECR8.CMADDMT9ZE bit is 1 and the POECMPEX3.POEREQ5 bit or the POECMPSEL.POEREQ5 bit is 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(15) GPTW0 pins (GTIOC0A, GTIOC0B)

When one of the following conditions is satisfied while the POECR3.GPT0ABZE bit is 1, the pins become high-impedance. Furthermore, when the POECR3.GPT0ABZE bit is 0 and the PMMCR2.GPT0AME and GPT0BME bits are 1 and when one of the following conditions is satisfied, the pins are switched to general I/O port pins.

- Operation for comparison of the output levels on the GTIOC0A and GTIOC0B signals
  When the OCSR3.OSF3 flag becomes 1 while the OCSR3.OCE3 bit is 1.
- SPOER setting
  When the SPOER.GPT01HIZ bit is set to 1.
  When the SPOER.GPT02HIZ bit is set to 1.
- Conditions added by POECR6
  When the ICSR1.POE0F flag becomes 1 while the POECR6.IC1ADDGPT01ZE bit is 1.
  When the ICSR2.POE4F flag becomes 1 while the POECR6.IC2ADDGPT01ZE bit is 1.
  When the ICSR3.POE8F flag becomes 1 while the POECR6.IC3ADDGPT01ZE bit and the ICSR3.POE8E bit are 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR6.IC4ADDGPT01ZE bit and the ICSR4.POE10E bit are 1.
  When the ICSR5.POE11F flag becomes 1 while the POECR6.IC5ADDGPT01ZE bit and the ICSR5.POE11E bit are 1.
  When the ICSR7.POE12F flag becomes 1 while the POECR6.IC6ADDGPT01ZE bit and the ICSR7.POE12E bit are 1.
  When the ICSR8.POE9F flag becomes 1 while the POECR6.IC8ADDGPT01ZE bit and the ICSR8.POE9E bit are 1.
- Conditions added by POECR9
  When the ICSR1.POE0F flag becomes 1 while the POECR9.IC1ADDGPT02ZE bit is 1.
  When the ICSR2.POE4F flag becomes 1 while the POECR9.IC2ADDGPT02ZE bit is 1.
  When the ICSR3.POE8F flag becomes 1 while the POECR9.IC3ADDGPT02ZE bit and the ICSR3.POE8E bit are 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR9.IC4ADDGPT02ZE bit and the ICSR4.POE10E bit are 1.
  When the ICSR5.POE11F flag becomes 1 while the POECR9.IC5ADDGPT02ZE bit and the ICSR5.POE11E bit are 1.
  When the ICSR7.POE12F flag becomes 1 while the POECR9.IC6ADDGPT02ZE bit and the ICSR7.POE12E bit are 1.
  When the ICSR8.POE9F flag becomes 1 while the POECR9.IC8ADDGPT02ZE bit and the ICSR8.POE9E bit are 1.
- Comparator output detection
  When the POECMPFR.C0FLAG flag becomes 1 while the POECR6.CMADDGPT01ZE bit is 1 and the POECMPEX4.POEREQ0 bit or the POECMPSEL.POEREQ0 bit is 1, or while the POECR9.CMADDGPT02ZE bit is 1 and the POECMPEX6.POEREQ0 bit or the POECMPSEL.POEREQ0 bit is 1.
  When the POECMPFR.C1FLAG flag becomes 1 while the POECR6.CMADDGPT01ZE bit is 1 and the POECMPEX4.POEREQ1 bit or the POECMPSEL.POEREQ1 bit is 1, or while the POECR9.CMADDGPT02ZE bit is 1 and the POECMPEX6.POEREQ1 bit or the POECMPSEL.POEREQ1 bit is 1.
  When the POECMPFR.C2FLAG flag becomes 1 while the POECR6.CMADDGPT01ZE bit is 1 and the POECMPEX4.POEREQ2 bit or the POECMPSEL.POEREQ2 bit is 1, or while the POECR9.CMADDGPT02ZE bit is 1 and the POECMPEX6.POEREQ2 bit or the POECMPSEL.POEREQ2 bit is 1.
  When the POECMPFR.C3FLAG flag becomes 1 while the POECR6.CMADDGPT01ZE bit is 1 and the POECMPEX4.POEREQ3 bit or the POECMPSEL.POEREQ3 bit is 1, or while the POECR9.CMADDGPT02ZE bit is 1 and the POECMPEX6.POEREQ3 bit or the POECMPSEL.POEREQ3 bit is 1.
  When the POECMPFR.C4FLAG flag becomes 1 while the POECR6.CMADDGPT01ZE bit is 1 and the POECMPEX4.POEREQ4 bit or the POECMPSEL.POEREQ4 bit is 1, or while the POECR9.CMADDGPT02ZE bit is 1 and the POECMPEX6.POEREQ4 bit or the POECMPSEL.POEREQ4 bit is 1.
  When the POECMPFR.C5FLAG flag becomes 1 while the POECR6.CMADDGPT01ZE bit is 1 and the POECMPEX4.POEREQ5 bit or the POECMPSEL.POEREQ5 bit is 1, or while the POECR9.CMADDGPT02ZE

bit is 1 and the POECMPEX6.POEREQ5 bit or the POECMPSEL.POEREQ5 bit is 1.

- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(16) GPTW1 pins (GTIOC1A, GTIOC1B)

When one of the following conditions is satisfied while the POECR3.GPT1ABZE bit is 1, the pins become high-impedance. Furthermore, when the POECR3.GPT1ABZE bit is 0 and the PMMCR2.GPT1AME and GPT1BME bits are 1 and when one of the following conditions is satisfied, the pins are switched to general I/O port pins.

- Operation for comparison of the output levels on the GTIOC1A and GTIOC1B signals
  When the OCSR3.OSF3 flag becomes 1 while the OCSR3.OCE3 bit is 1.
- SPOER setting
  When the SPOER.GPT01HIZ bit is set to 1.
  When the SPOER.GPT02HIZ bit is set to 1.
- Conditions added by POECR6
  When the ICSR1.POE0F flag becomes 1 while the POECR6.IC1ADDGPT01ZE bit is 1.
  When the ICSR2.POE4F flag becomes 1 while the POECR6.IC2ADDGPT01ZE bit is 1.
  When the ICSR3.POE8F flag becomes 1 while the POECR6.IC3ADDGPT01ZE bit and the ICSR3.POE8E bit are 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR6.IC4ADDGPT01ZE bit and the ICSR4.POE10E bit are 1.
  When the ICSR5.POE11F flag becomes 1 while the POECR6.IC5ADDGPT01ZE bit and the ICSR5.POE11E bit are 1.
  When the ICSR7.POE12F flag becomes 1 while the POECR6.IC6ADDGPT01ZE bit and the ICSR7.POE12E bit are 1.
  When the ICSR8.POE9F flag becomes 1 while the POECR6.IC8ADDGPT01ZE bit and the ICSR8.POE9E bit are 1.
- Conditions added by POECR9
  When the ICSR1.POE0F flag becomes 1 while the POECR9.IC1ADDGPT02ZE bit is 1.
  When the ICSR2.POE4F flag becomes 1 while the POECR9.IC2ADDGPT02ZE bit is 1.
  When the ICSR3.POE8F flag becomes 1 while the POECR9.IC3ADDGPT02ZE bit and the ICSR3.POE8E bit are 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR9.IC4ADDGPT02ZE bit and the ICSR4.POE10E bit are 1.
  When the ICSR5.POE11F flag becomes 1 while the POECR9.IC5ADDGPT02ZE bit and the ICSR5.POE11E bit are 1.
  When the ICSR7.POE12F flag becomes 1 while the POECR9.IC6ADDGPT02ZE bit and the ICSR7.POE12E bit are 1.
  When the ICSR8.POE9F flag becomes 1 while the POECR9.IC8ADDGPT02ZE bit and the ICSR8.POE9E bit are 1.
- Comparator output detection
  When the POECMPFR.C0FLAG flag becomes 1 while the POECR6.CMADDGPT01ZE bit is 1 and the POECMPEX4.POEREQ0 bit or the POECMPSEL.POEREQ0 bit is 1, or while the POECR9.CMADDGPT02ZE bit is 1 and the POECMPEX6.POEREQ0 bit or the POECMPSEL.POEREQ0 bit is 1.
  When the POECMPFR.C1FLAG flag becomes 1 while the POECR6.CMADDGPT01ZE bit is 1 and the POECMPEX4.POEREQ1 bit or the POECMPSEL.POEREQ1 bit is 1, or while the POECR9.CMADDGPT02ZE bit is 1 and the POECMPEX6.POEREQ1 bit or the POECMPSEL.POEREQ1 bit is 1.
  When the POECMPFR.C2FLAG flag becomes 1 while the POECR6.CMADDGPT01ZE bit is 1 and the POECMPEX4.POEREQ2 bit or the POECMPSEL.POEREQ2 bit is 1, or while the POECR9.CMADDGPT02ZE bit is 1 and the POECMPEX6.POEREQ2 bit or the POECMPSEL.POEREQ2 bit is 1.

When the POECMPFR.C3FLAG flag becomes 1 while the POECR6.CMADDGPT01ZE bit is 1 and the POECMPEX4.POEREQ3 bit or the POECMPSEL.POEREQ3 bit is 1, or while the POECR9.CMADDGPT02ZE bit is 1 and the POECMPEX6.POEREQ3 bit or the POECMPSEL.POEREQ3 bit is 1.
When the POECMPFR.C4FLAG flag becomes 1 while the POECR6.CMADDGPT01ZE bit is 1 and the POECMPEX4.POEREQ4 bit or the POECMPSEL.POEREQ4 bit is 1, or while the POECR9.CMADDGPT02ZE bit is 1 and the POECMPEX6.POEREQ4 bit or the POECMPSEL.POEREQ4 bit is 1.
When the POECMPFR.C5FLAG flag becomes 1 while the POECR6.CMADDGPT01ZE bit is 1 and the POECMPEX4.POEREQ5 bit or the POECMPSEL.POEREQ5 bit is 1, or while the POECR9.CMADDGPT02ZE bit is 1 and the POECMPEX6.POEREQ5 bit or the POECMPSEL.POEREQ5 bit is 1.

- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(17) GPTW2 pins (GTIOC2A, GTIOC2B)

When one of the following conditions is satisfied while the POECR3.GPT2ABZE bit is 1, the pins become high-impedance. Furthermore, when the POECR3.GPT2ABZE bit is 0 and the PMMCR2.GPT2AME and GPT2BME bits are 1 and when one of the following conditions is satisfied, the pins are switched to general I/O port pins.

- Operation for comparison of the output levels on the GTIOC2A and GTIOC2B signals
  When the OCSR3.OSF3 flag becomes 1 while the OCSR3.OCE3 bit is 1.
- SPOER setting
  When the SPOER.GPT23HIZ bit is set to 1.
  When the SPOER.GPT02HIZ bit is set to 1.
- Conditions added by POECR6B
  When the ICSR1.POE0F flag becomes 1 while the POECR6B.IC1ADDGPT23ZE bit is 1.
  When the ICSR2.POE4F flag becomes 1 while the POECR6B.IC2ADDGPT23ZE bit is 1.
  When the ICSR3.POE8F flag becomes 1 while the POECR6B.IC3ADDGPT23ZE bit and the ICSR3.POE8E bit are 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR6B.IC4ADDGPT23ZE bit and the ICSR4.POE10E bit are 1.
  When the ICSR5.POE11F flag becomes 1 while the POECR6B.IC5ADDGPT23ZE bit and the ICSR5.POE11E bit are 1.
  When the ICSR7.POE12F flag becomes 1 while the POECR6B.IC6ADDGPT23ZE bit and the ICSR7.POE12E bit are 1.
  When the ICSR8.POE9F flag becomes 1 while the POECR6B.IC8ADDGPT23ZE bit and the ICSR8.POE9E bit are 1.
- Conditions added by POECR9
  When the ICSR1.POE0F flag becomes 1 while the POECR9.IC1ADDGPT02ZE bit is 1.
  When the ICSR2.POE4F flag becomes 1 while the POECR9.IC2ADDGPT02ZE bit is 1.
  When the ICSR3.POE8F flag becomes 1 while the POECR9.IC3ADDGPT02ZE bit and the ICSR3.POE8E bit are 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR9.IC4ADDGPT02ZE bit and the ICSR4.POE10E bit are 1.
  When the ICSR5.POE11F flag becomes 1 while the POECR9.IC5ADDGPT02ZE bit and the ICSR5.POE11E bit are 1.
  When the ICSR7.POE12F flag becomes 1 while the POECR9.IC6ADDGPT02ZE bit and the ICSR7.POE12E bit are 1.
  When the ICSR8.POE9F flag becomes 1 while the POECR9.IC8ADDGPT02ZE bit and the ICSR8.POE9E bit are 1.
- Comparator output detection
  When the POECMPFR.C0FLAG flag becomes 1 while the POECR6B.CMADDGPT23ZE bit is 1 and the

POECMPEX5.POEREQ0 bit or the POECMPSEL.POEREQ0 bit is 1, or while the POECR9.CMADDGPT02ZE bit is 1 and the POECMPEX6.POEREQ0 bit or the POECMPSEL.POEREQ0 bit is 1.
When the POECMPFR.C1FLAG flag becomes 1 while the POECR6B.CMADDGPT23ZE bit is 1 and the POECMPEX5.POEREQ1 bit or the POECMPSEL.POEREQ1 bit is 1, or while the POECR9.CMADDGPT02ZE bit is 1 and the POECMPEX6.POEREQ1 bit or the POECMPSEL.POEREQ1 bit is 1.
When the POECMPFR.C2FLAG flag becomes 1 while the POECR6B.CMADDGPT23ZE bit is 1 and the POECMPEX5.POEREQ2 bit or the POECMPSEL.POEREQ2 bit is 1, or while the POECR9.CMADDGPT02ZE bit is 1 and the POECMPEX6.POEREQ2 bit or the POECMPSEL.POEREQ2 bit is 1.
When the POECMPFR.C3FLAG flag becomes 1 while the POECR6B.CMADDGPT23ZE bit is 1 and the POECMPEX5.POEREQ3 bit or the POECMPSEL.POEREQ3 bit is 1, or while the POECR9.CMADDGPT02ZE bit is 1 and the POECMPEX6.POEREQ3 bit or the POECMPSEL.POEREQ3 bit is 1.
When the POECMPFR.C4FLAG flag becomes 1 while the POECR6B.CMADDGPT23ZE bit is 1 and the POECMPEX5.POEREQ4 bit or the POECMPSEL.POEREQ4 bit is 1, or while the POECR9.CMADDGPT02ZE bit is 1 and the POECMPEX6.POEREQ4 bit or the POECMPSEL.POEREQ4 bit is 1.
When the POECMPFR.C5FLAG flag becomes 1 while the POECR6B.CMADDGPT23ZE bit is 1 and the POECMPEX5.POEREQ5 bit or the POECMPSEL.POEREQ5 bit is 1, or while the POECR9.CMADDGPT02ZE bit is 1 and the POECMPEX6.POEREQ5 bit or the POECMPSEL.POEREQ5 bit is 1.

- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(18) GPTW3 pins (GTIOC3A, GTIOC3B)

When one of the following conditions is satisfied while the POECR3.GPT3ABZE bit is 1, the pins become high-impedance. Furthermore, when the POECR3.GPT3ABZE bit is 0 and the PMMCR2.GPT3AME and GPT3BME bits are 1 and when one of the following conditions is satisfied, the pins are switched to general I/O port pins.

- SPOER setting
  When the SPOER.GPT23HIZ bit is set to 1.
- Conditions added by POECR6B
  When the ICSR1.POE0F flag becomes 1 while the POECR6B.IC1ADDGPT23ZE bit is 1.
  When the ICSR2.POE4F flag becomes 1 while the POECR6B.IC2ADDGPT23ZE bit is 1.
  When the ICSR3.POE8F flag becomes 1 while the POECR6B.IC3ADDGPT23ZE bit and the ICSR3.POE8E bit are 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR6B.IC4ADDGPT23ZE bit and the ICSR4.POE10E bit are 1.
  When the ICSR5.POE11F flag becomes 1 while the POECR6B.IC5ADDGPT23ZE bit and the ICSR5.POE11E bit are 1.
  When the ICSR7.POE12F flag becomes 1 while the POECR6B.IC6ADDGPT23ZE bit and the ICSR7.POE12E bit are 1.
  When the ICSR8.POE9F flag becomes 1 while the POECR6B.IC8ADDGPT23ZE bit and the ICSR8.POE9E bit are 1.
- Comparator output detection
  When the POECMPFR.C0FLAG flag becomes 1 while the POECR6B.CMADDGPT23ZE bit is 1 and the POECMPEX5.POEREQ0 bit or the POECMPSEL.POEREQ0 bit is 1.
  When the POECMPFR.C1FLAG flag becomes 1 while the POECR6B.CMADDGPT23ZE bit is 1 and the POECMPEX5.POEREQ1 bit or the POECMPSEL.POEREQ1 bit is 1.
  When the POECMPFR.C2FLAG flag becomes 1 while the POECR6B.CMADDGPT23ZE bit is 1 and the POECMPEX5.POEREQ2 bit or the POECMPSEL.POEREQ2 bit is 1.
  When the POECMPFR.C3FLAG flag becomes 1 while the POECR6B.CMADDGPT23ZE bit is 1 and the POECMPEX5.POEREQ3 bit or the POECMPSEL.POEREQ3 bit is 1.
  When the POECMPFR.C4FLAG flag becomes 1 while the POECR6B.CMADDGPT23ZE bit is 1 and the

POECMPEX5.POEREQ4 bit or the POECMPSEL.POEREQ4 bit is 1.
When the POECMPFR.C5FLAG flag becomes 1 while the POECR6B.CMADDGPT23ZE bit is 1 and the POECMPEX5.POEREQ5 bit or the POECMPSEL.POEREQ5 bit is 1.

- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(19) GPTW4 pins (GTIOC4A, GTIOC4B)

When one of the following conditions is satisfied while the POECR3.GPT4ABZE bit is 1, the pins become high-impedance. Furthermore, when the POECR3.GPT4ABZE bit is 0 and the PMMCR2.GPT4AME and GPT4BME bits are 1 and when one of the following conditions is satisfied, the pins are switched to general I/O port pins.

- Operation for comparison of the output levels on the GTIOC4A and GTIOC4B signals
  When the OCSR4.OSF4 flag becomes 1 while the OCSR4.OCE4 bit is 1.
- SPOER setting
  When the SPOER.GPT46HIZ bit is set to 1.
- Conditions added by POECR10
  When the ICSR1.POE0F flag becomes 1 while the POECR10.IC1ADDGPT46ZE bit is 1.
  When the ICSR2.POE4F flag becomes 1 while the POECR10.IC2ADDGPT46ZE bit is 1.
  When the ICSR3.POE8F flag becomes 1 while the POECR10.IC3ADDGPT46ZE bit and the ICSR3.POE8E bit are 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR10.IC4ADDGPT46ZE bit and the ICSR4.POE10E bit are 1.
  When the ICSR5.POE11F flag becomes 1 while the POECR10.IC5ADDGPT46ZE bit and the ICSR5.POE11E bit are 1.
  When the ICSR7.POE12F flag becomes 1 while the POECR10.IC6ADDGPT46ZE bit and the ICSR7.POE12E bit are 1.
  When the ICSR8.POE9F flag becomes 1 while the POECR10.IC8ADDGPT46ZE bit and the ICSR8.POE9E bit are 1.
- Comparator output detection
  When the POECMPFR.C0FLAG flag becomes 1 while the POECR10.CMADDGPT46ZE bit is 1 and the POECMPEX7.POEREQ0 bit or the POECMPSEL.POEREQ0 bit is 1.
  When the POECMPFR.C1FLAG flag becomes 1 while the POECR10.CMADDGPT46ZE bit is 1 and the POECMPEX7.POEREQ1 bit or the POECMPSEL.POEREQ1 bit is 1.
  When the POECMPFR.C2FLAG flag becomes 1 while the POECR10.CMADDGPT46ZE bit is 1 and the POECMPEX7.POEREQ2 bit or the POECMPSEL.POEREQ2 bit is 1.
  When the POECMPFR.C3FLAG flag becomes 1 while the POECR10.CMADDGPT46ZE bit is 1 and the POECMPEX7.POEREQ3 bit or the POECMPSEL.POEREQ3 bit is 1.
  When the POECMPFR.C4FLAG flag becomes 1 while the POECR10.CMADDGPT46ZE bit is 1 and the POECMPEX7.POEREQ4 bit or the POECMPSEL.POEREQ4 bit is 1.
  When the POECMPFR.C5FLAG flag becomes 1 while the POECR10.CMADDGPT46ZE bit is 1 and the POECMPEX7.POEREQ5 bit or the POECMPSEL.POEREQ5 bit is 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(20) GPTW5 pins (GTIOC5A, GTIOC5B)

When one of the following conditions is satisfied while the POECR3.GPT5ABZE bit is 1, the pins become high-impedance. Furthermore, when the POECR3.GPT5ABZE bit is 0 and the PMMCR2.GPT5AME and GPT5BME bits are 1 and when one of the following conditions is satisfied, the pins are switched to general I/O port pins.

- Operation for comparison of the output levels on the GTIOC5A and GTIOC5B signals
  When the OCSR4.OSF4 flag becomes 1 while the OCSR4.OCE4 bit is 1.

- SPOER setting
  When the SPOER.GPT46HIZ bit is set to 1.
- Conditions added by POECR10
  When the ICSR1.POE0F flag becomes 1 while the POECR10.IC1ADDGPT46ZE bit is 1.
  When the ICSR2.POE4F flag becomes 1 while the POECR10.IC2ADDGPT46ZE bit is 1.
  When the ICSR3.POE8F flag becomes 1 while the POECR10.IC3ADDGPT46ZE bit and the ICSR3.POE8E bit are 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR10.IC4ADDGPT46ZE bit and the ICSR4.POE10E bit are 1.
  When the ICSR5.POE11F flag becomes 1 while the POECR10.IC5ADDGPT46ZE bit and the ICSR5.POE11E bit are 1.
  When the ICSR7.POE12F flag becomes 1 while the POECR10.IC6ADDGPT46ZE bit and the ICSR7.POE12E bit are 1.
  When the ICSR8.POE9F flag becomes 1 while the POECR10.IC8ADDGPT46ZE bit and the ICSR8.POE9E bit are 1.
- Comparator output detection
  When the POECMPFR.C0FLAG flag becomes 1 while the POECR10.CMADDGPT46ZE bit is 1 and the POECMPEX7.POEREQ0 bit or the POECMPSEL.POEREQ0 bit is 1.
  When the POECMPFR.C1FLAG flag becomes 1 while the POECR10.CMADDGPT46ZE bit is 1 and the POECMPEX7.POEREQ1 bit or the POECMPSEL.POEREQ1 bit is 1.
  When the POECMPFR.C2FLAG flag becomes 1 while the POECR10.CMADDGPT46ZE bit is 1 and the POECMPEX7.POEREQ2 bit or the POECMPSEL.POEREQ2 bit is 1.
  When the POECMPFR.C3FLAG flag becomes 1 while the POECR10.CMADDGPT46ZE bit is 1 and the POECMPEX7.POEREQ3 bit or the POECMPSEL.POEREQ3 bit is 1.
  When the POECMPFR.C4FLAG flag becomes 1 while the POECR10.CMADDGPT46ZE bit is 1 and the POECMPEX7.POEREQ4 bit or the POECMPSEL.POEREQ4 bit is 1.
  When the POECMPFR.C5FLAG flag becomes 1 while the POECR10.CMADDGPT46ZE bit is 1 and the POECMPEX7.POEREQ5 bit or the POECMPSEL.POEREQ5 bit is 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(21) GPTW6 pins (GTIOC6A, GTIOC6B)

When one of the following conditions is satisfied while the POECR3.GPT6ABZE bit is 1, the pins become high-impedance. Furthermore, when the POECR3.GPT6ABZE bit is 0 and the PMMCR2.GPT6AME and GPT6BME bits are 1 and when one of the following conditions is satisfied, the pins are switched to general I/O port pins.

- Operation for comparison of the output levels on the GTIOC6A and GTIOC6B signals
  When the OCSR4.OSF4 flag becomes 1 while the OCSR4.OCE4 bit is 1.
- SPOER setting
  When the SPOER.GPT46HIZ bit is set to 1.
- Conditions added by POECR10
  When the ICSR1.POE0F flag becomes 1 while the POECR10.IC1ADDGPT46ZE bit is 1.
  When the ICSR2.POE4F flag becomes 1 while the POECR10.IC2ADDGPT46ZE bit is 1.
  When the ICSR3.POE8F flag becomes 1 while the POECR10.IC3ADDGPT46ZE bit and the ICSR3.POE8E bit are 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR10.IC4ADDGPT46ZE bit and the ICSR4.POE10E bit are 1.
  When the ICSR5.POE11F flag becomes 1 while the POECR10.IC5ADDGPT46ZE bit and the ICSR5.POE11E bit are 1.
  When the ICSR7.POE12F flag becomes 1 while the POECR10.IC6ADDGPT46ZE bit and the ICSR7.POE12E bit

are 1.
When the ICSR8.POE9F flag becomes 1 while the POECR10.IC8ADDGPT46ZE bit and the ICSR8.POE9E bit are 1.

- Comparator output detection
When the POECMPFR.C0FLAG flag becomes 1 while the POECR10.CMADDGPT46ZE bit is 1 and the POECMPEX7.POEREQ0 bit or the POECMPSEL.POEREQ0 bit is 1.
When the POECMPFR.C1FLAG flag becomes 1 while the POECR10.CMADDGPT46ZE bit is 1 and the POECMPEX7.POEREQ1 bit or the POECMPSEL.POEREQ1 bit is 1.
When the POECMPFR.C2FLAG flag becomes 1 while the POECR10.CMADDGPT46ZE bit is 1 and the POECMPEX7.POEREQ2 bit or the POECMPSEL.POEREQ2 bit is 1.
When the POECMPFR.C3FLAG flag becomes 1 while the POECR10.CMADDGPT46ZE bit is 1 and the POECMPEX7.POEREQ3 bit or the POECMPSEL.POEREQ3 bit is 1.
When the POECMPFR.C4FLAG flag becomes 1 while the POECR10.CMADDGPT46ZE bit is 1 and the POECMPEX7.POEREQ4 bit or the POECMPSEL.POEREQ4 bit is 1.
When the POECMPFR.C5FLAG flag becomes 1 while the POECR10.CMADDGPT46ZE bit is 1 and the POECMPEX7.POEREQ5 bit or the POECMPSEL.POEREQ5 bit is 1.
- Detection of oscillation stop
When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(22) GPTW7 pins (GTIOC7A, GTIOC7B)

When one of the following conditions is satisfied while the POECR3.GPT7ABZE bit is 1, the pins become high-impedance. Furthermore, when the POECR3.GPT7ABZE bit is 0 and the PMMCR2.GPT7AME and GPT7BME bits are 1 and when one of the following conditions is satisfied, the pins are switched to general I/O port pins.

- Operation for comparison of the output levels on the GTIOC7A and GTIOC7B signals
When the OCSR5.OSF5 flag becomes 1 while the OCSR5.OCE5 bit is 1.
- SPOER setting
When the SPOER.GPT79HIZ bit is set to 1.
- Conditions added by POECR11
When the ICSR1.POE0F flag becomes 1 while the POECR11.IC1ADDGPT79ZE bit is 1.
When the ICSR2.POE4F flag becomes 1 while the POECR11.IC2ADDGPT79ZE bit is 1.
When the ICSR3.POE8F flag becomes 1 while the POECR11.IC3ADDGPT79ZE bit and the ICSR3.POE8E bit are 1.
When the ICSR4.POE10F flag becomes 1 while the POECR11.IC4ADDGPT79ZE bit and the ICSR4.POE10E bit are 1.
When the ICSR5.POE11F flag becomes 1 while the POECR11.IC5ADDGPT79ZE bit and the ICSR5.POE11E bit are 1.
When the ICSR7.POE12F flag becomes 1 while the POECR11.IC6ADDGPT79ZE bit and the ICSR7.POE12E bit are 1.
When the ICSR8.POE9F flag becomes 1 while the POECR11.IC8ADDGPT79ZE bit and the ICSR8.POE9E bit are 1.
- Comparator output detection
When the POECMPFR.C0FLAG flag becomes 1 while the POECR11.CMADDGPT79ZE bit is 1 and the POECMPEX8.POEREQ0 bit or the POECMPSEL.POEREQ0 bit is 1.
When the POECMPFR.C1FLAG flag becomes 1 while the POECR11.CMADDGPT79ZE bit is 1 and the POECMPEX8.POEREQ1 bit or the POECMPSEL.POEREQ1 bit is 1.
When the POECMPFR.C2FLAG flag becomes 1 while the POECR11.CMADDGPT79ZE bit is 1 and the POECMPEX8.POEREQ2 bit or the POECMPSEL.POEREQ2 bit is 1.
When the POECMPFR.C3FLAG flag becomes 1 while the POECR11.CMADDGPT79ZE bit is 1 and the POECMPEX8.POEREQ3 bit or the POECMPSEL.POEREQ3 bit is 1.

When the POECMPFR.C4FLAG flag becomes 1 while the POECR11.CMADDGPT79ZE bit is 1 and the POECMPEX8.POEREQ4 bit or the POECMPSEL.POEREQ4 bit is 1.
When the POECMPFR.C5FLAG flag becomes 1 while the POECR11.CMADDGPT79ZE bit is 1 and the POECMPEX8.POEREQ5 bit or the POECMPSEL.POEREQ5 bit is 1.

- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

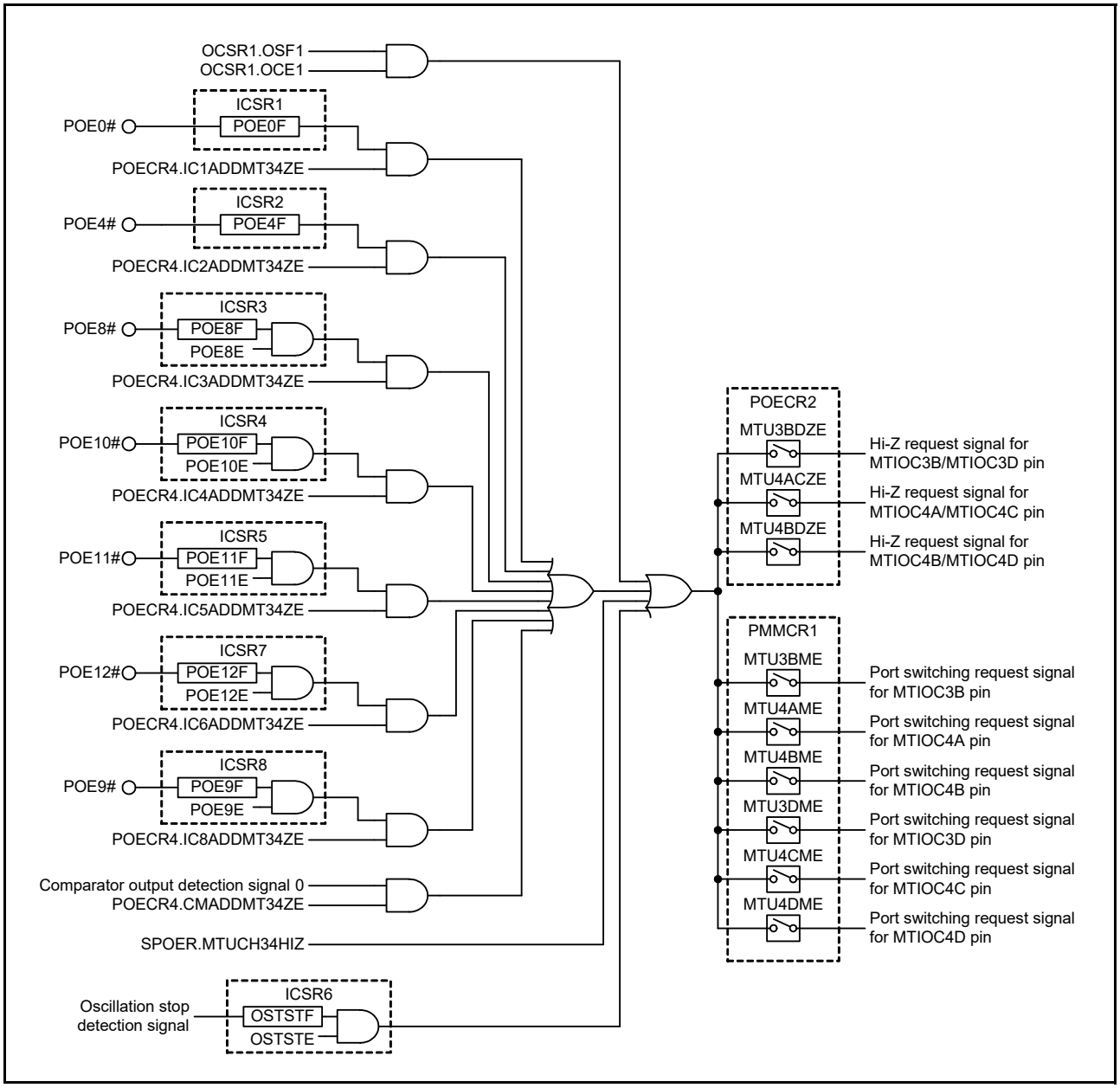


Figure 23.3 Target Pins and Conditions for High-Impedance Control (1)

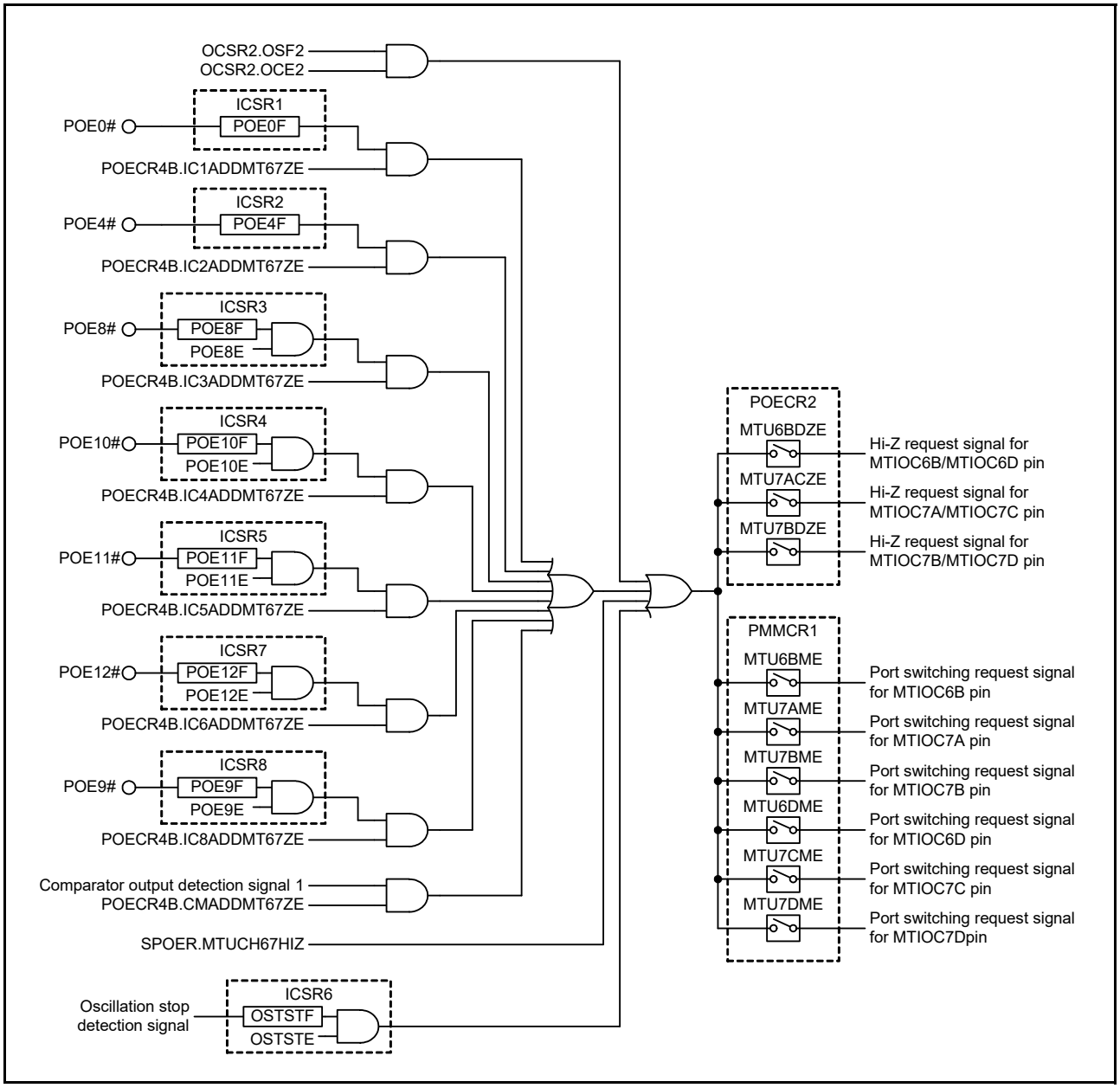


Figure 23.4 Target Pins and Conditions for High-Impedance Control (2)

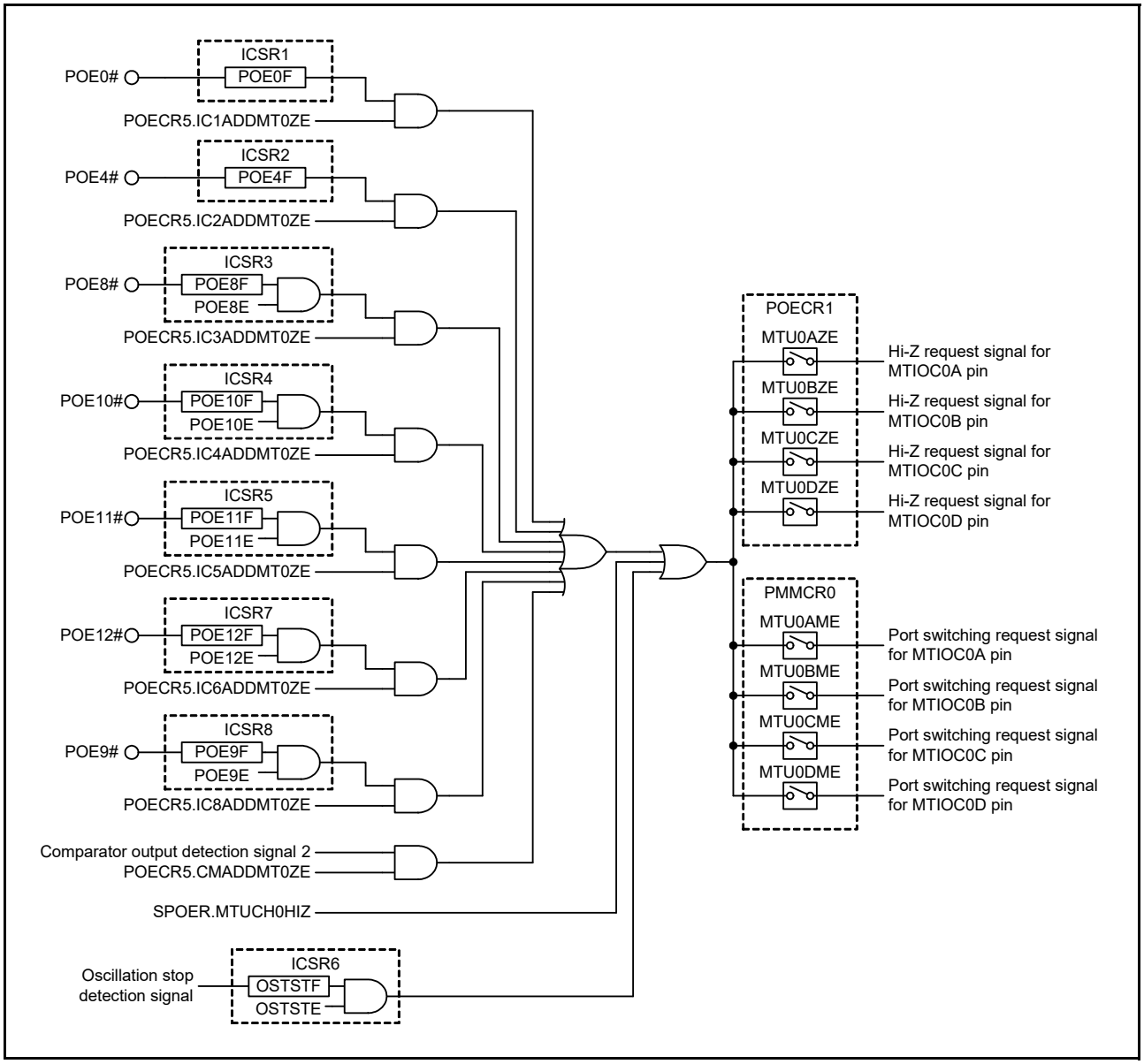


Figure 23.5 Target Pins and Conditions for High-Impedance Control (3)

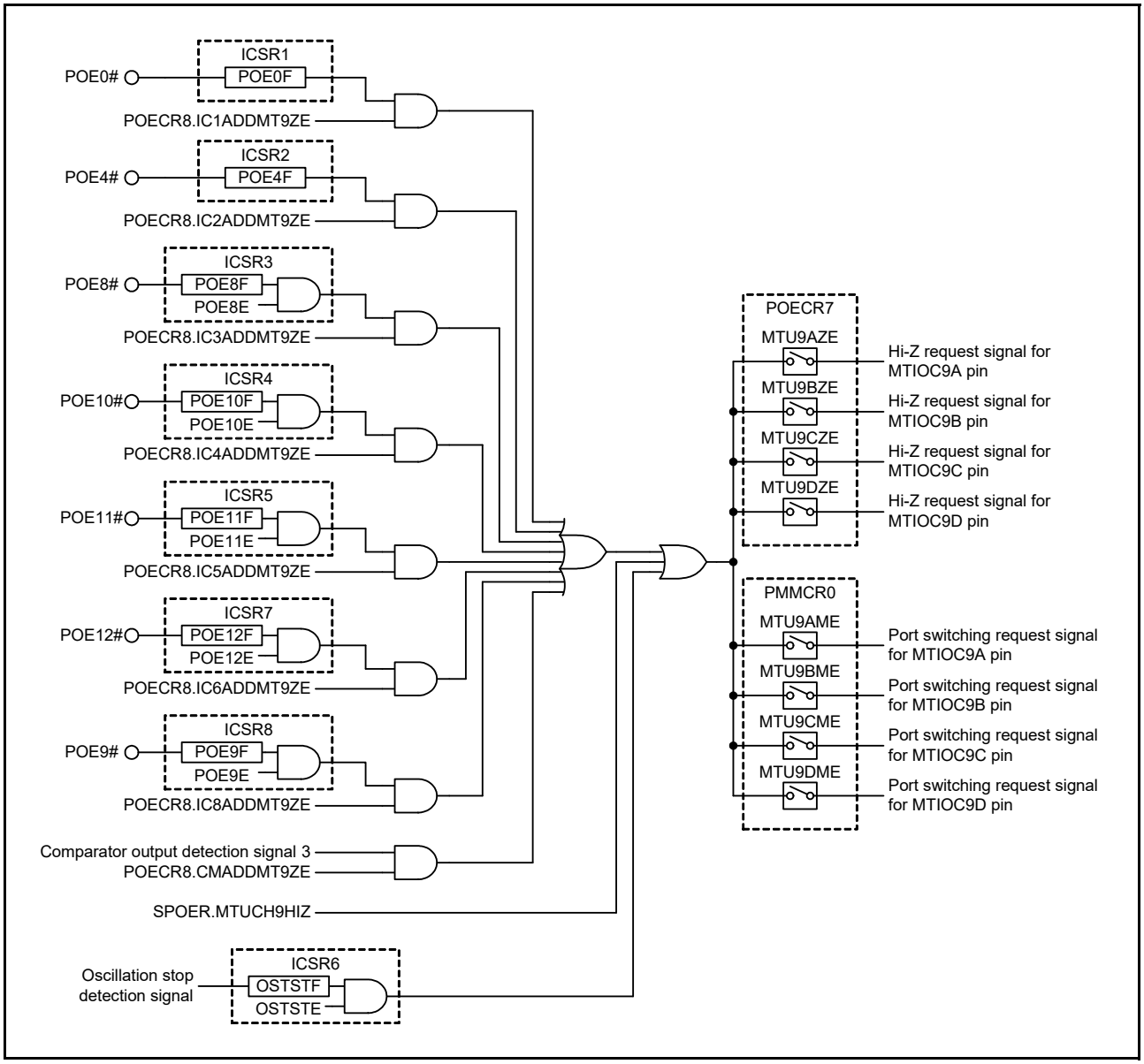


Figure 23.6 Target Pins and Conditions for High-Impedance Control (4)

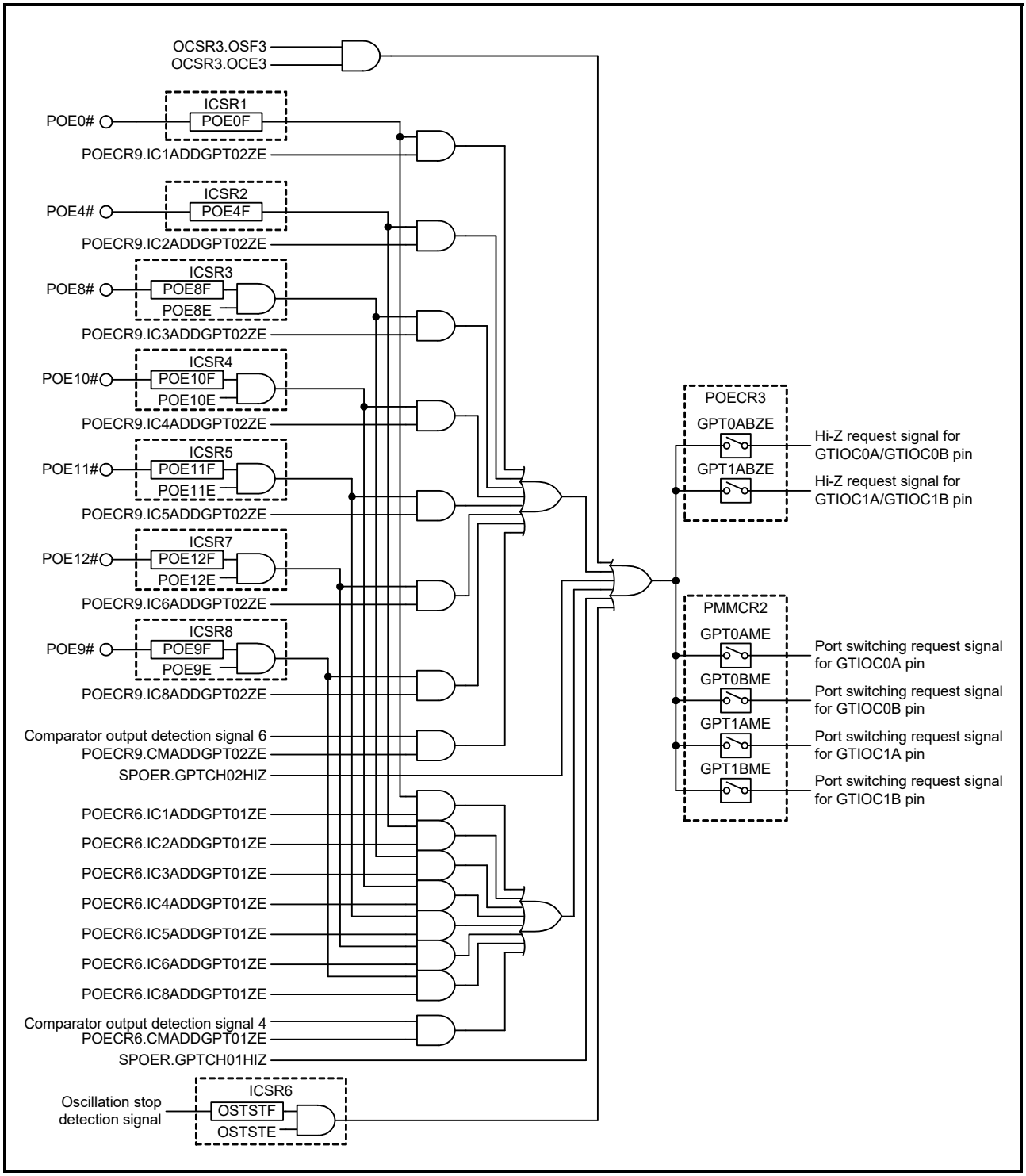


Figure 23.7 Target Pins and Conditions for High-Impedance Control (5)

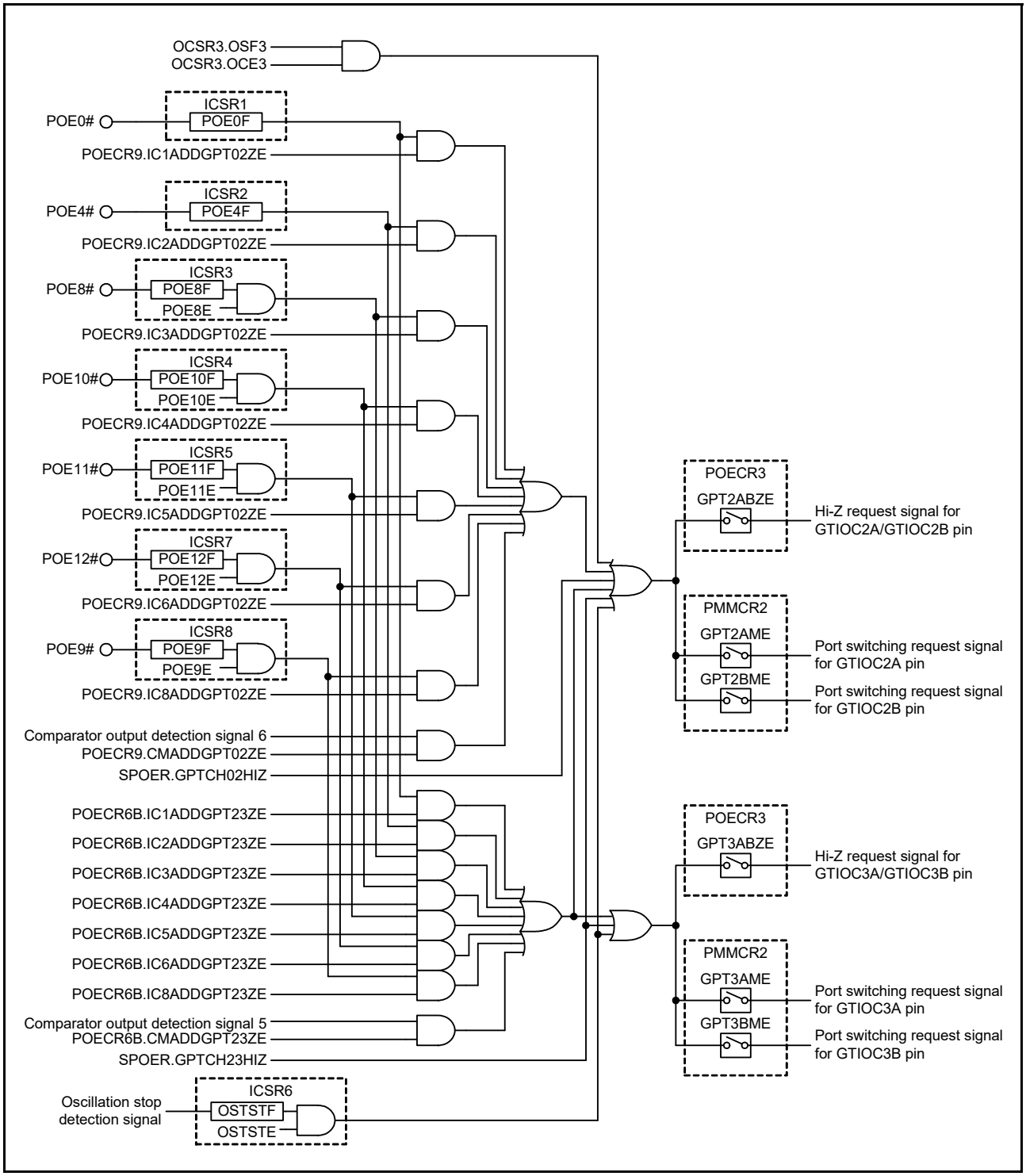


Figure 23.8 Target Pins and Conditions for High-Impedance Control (6)

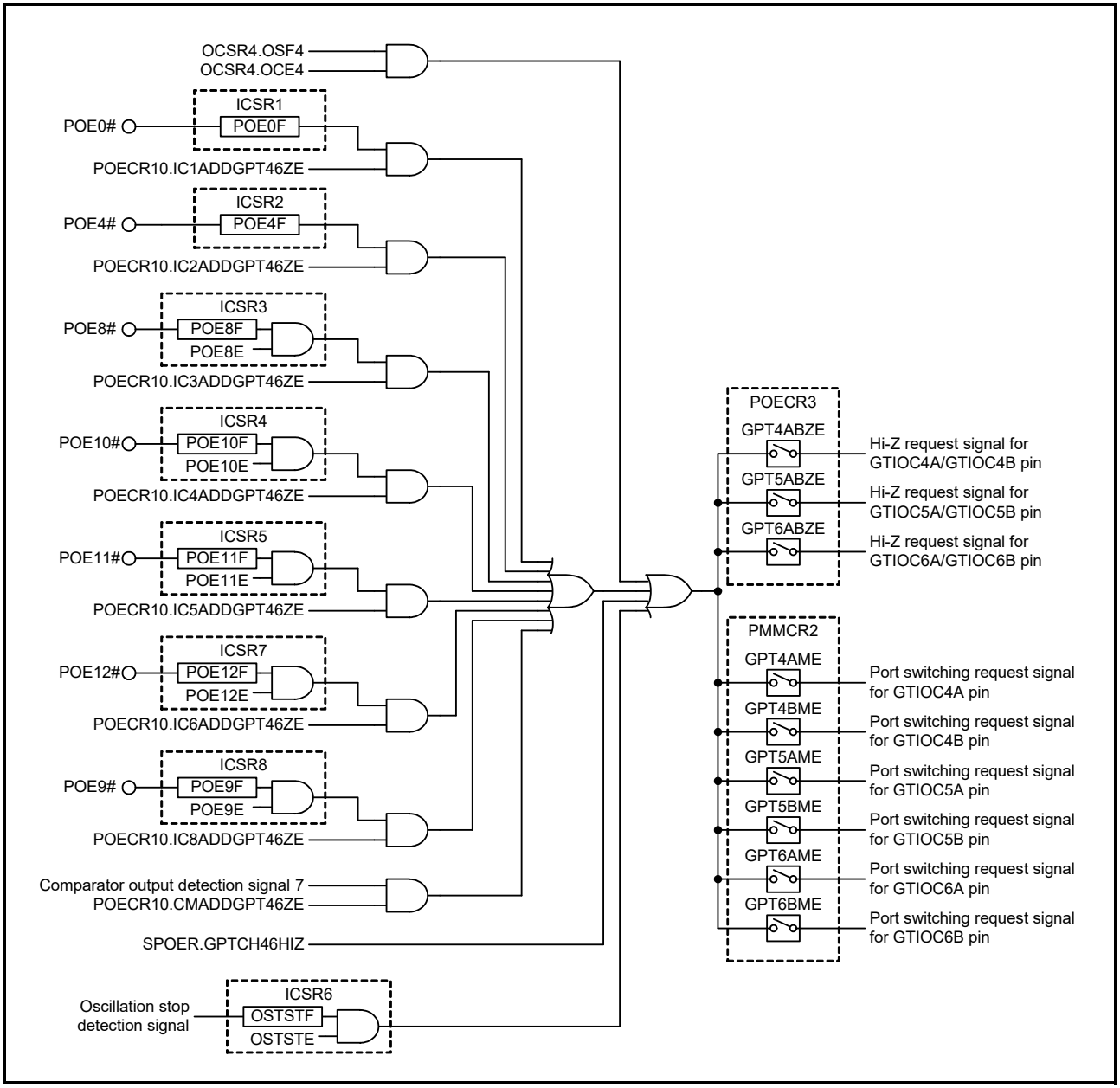


Figure 23.9 Target Pins and Conditions for High-Impedance Control (7)

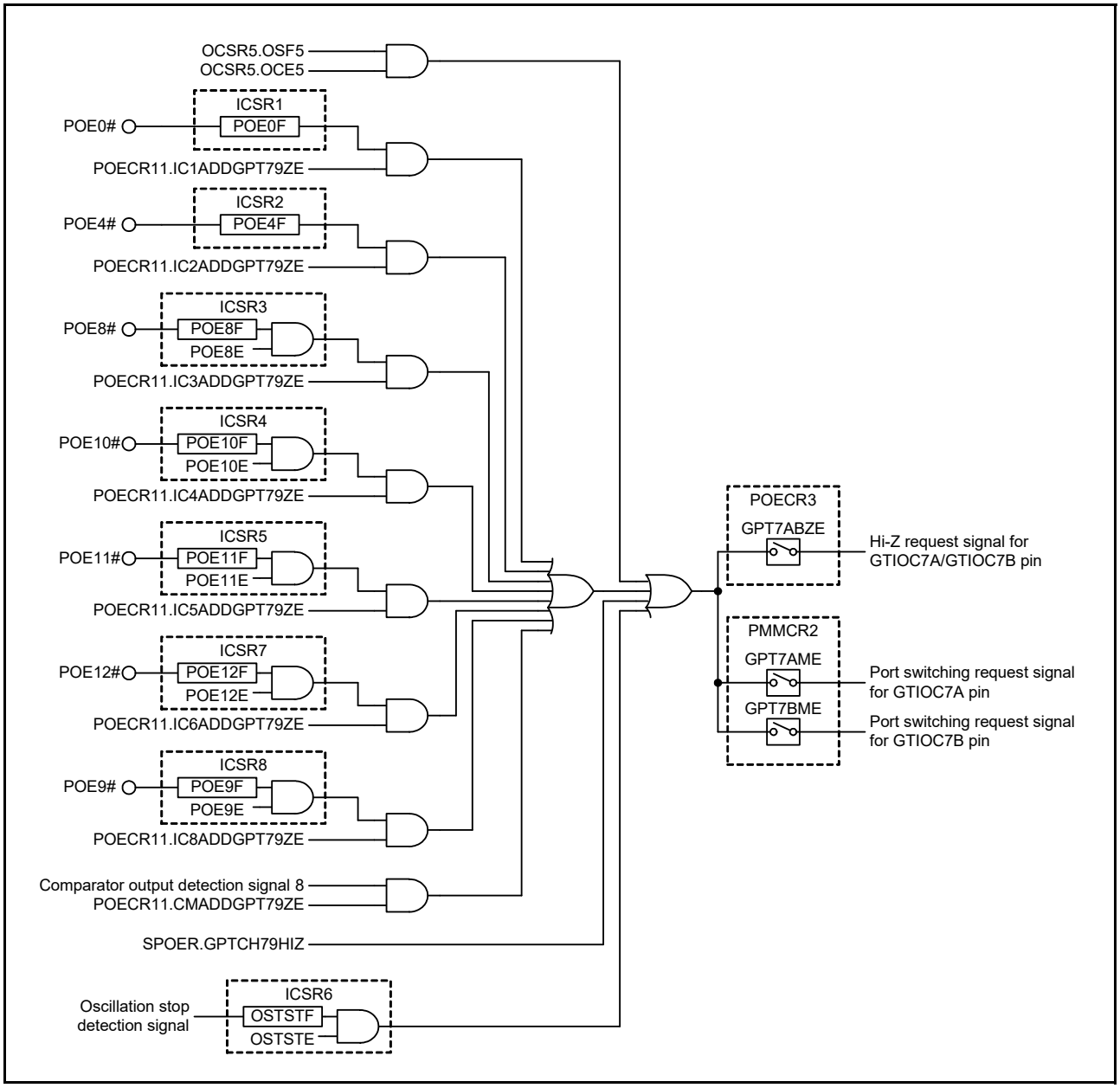


Figure 23.10 Target Pins and Conditions for High-Impedance Control (8)

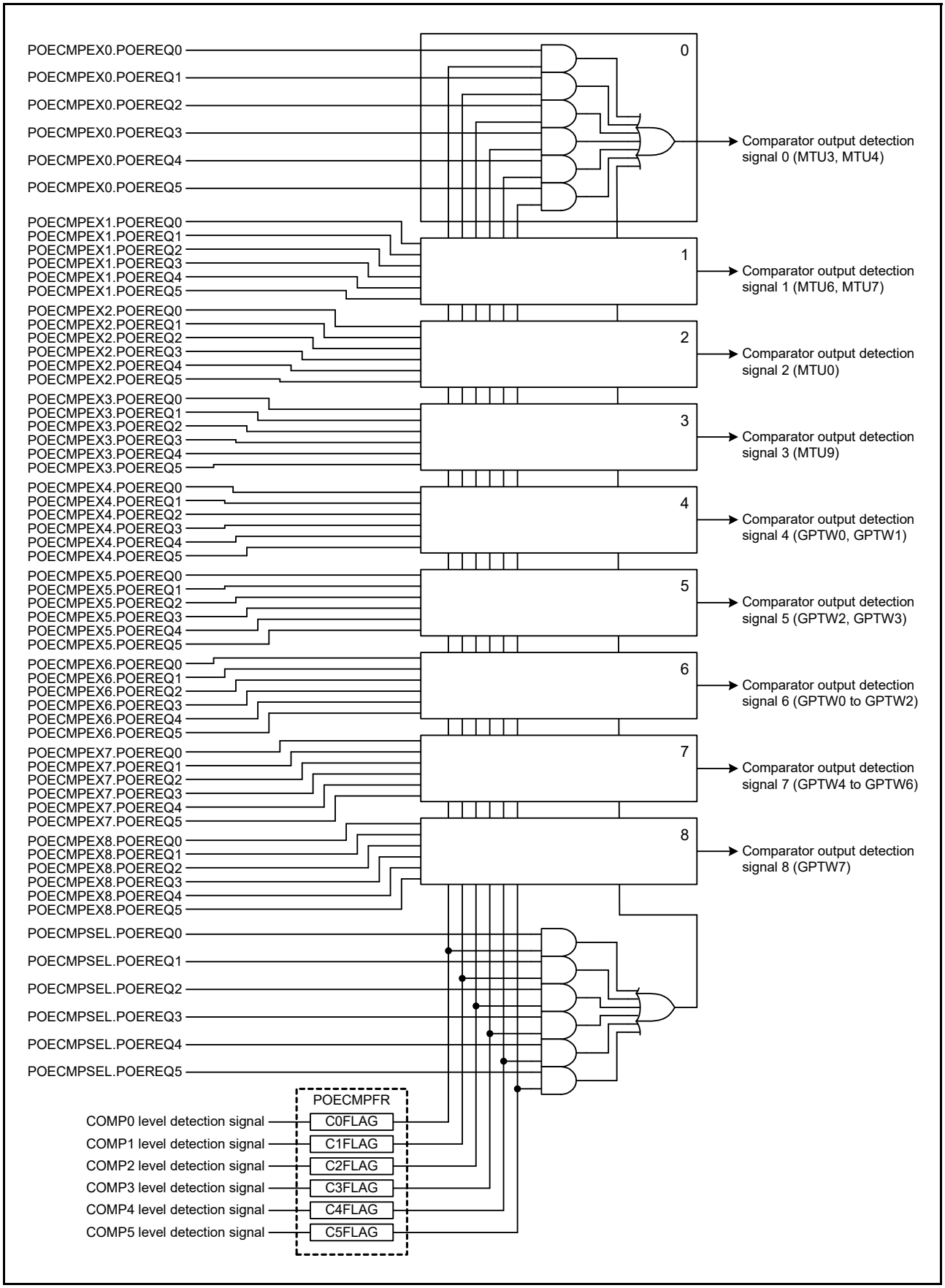


Figure 23.11 Comparator Output Detection Signal Generator Block Diagram

23.3.1 MTU/GPTW Pin Selection

In this MCU, the pin functions for MTU and GPTW are respectively multiplexed with multiple sets of port pins. The target pins for output disabling control can be selected by the pin select register in POE (M0SELR1, M0SELR2, M3SELR, M4SELR1, M4SELR2, M6SELR, M7SELR1, M7SELR2, M9SELR1, M9SELR2, G0SELR, G1SELR, G2SELR, G3SELR, G4SELR, G5SELR, G6SELR, or G7SELR register). Table 23.5 shows the correspondence between MTU pins and select registers. Table 23.6 shows the correspondence between GPTW pins and select registers. Note that settings for pins to be used as MTU or GPTW must be separately made in the registers of the multi-function pin controller (MPC). Take care so that there are no differences between the pins selected in the POE registers and the pins selected in the MPC registers.

Table 23.5 Correspondence between MTU Pins

MTU Pin Functions	Corresponding Ports	Select Registers
MTIOC0A	PB3	M0SELR1
	P31	
	P70	
MTIOC0B	PB2	
	P30	
MTIOC0C	PB1	M0SELR2
	P27	
MTIOC0D	PB0	
MTIOC3B	P71	M3SELR
MTIOC3D	P74	
MTIOC4A	P72	M4SELR1
MTIOC4C	P75	
MTIOC4B	P73	M4SELR2
MTIOC4D	P76	
MTIOC6B	P95	M6SELR
MTIOC6D	P92	
MTIOC7A	P94	M7SELR1
MTIOC7C	P91	
MTIOC7B	P93	M7SELR2
MTIOC7D	P90	

MTU Pin Functions	Corresponding Ports	Select Registers
MTIOC9A	PD7	M9SELR1
	P21	
	P00	
MTIOC9B	PE0	
	P10	
	P22	
MTIOC9C	PD6	M9SELR2
	P20	
	P01	
MTIOC9D	PE1	
	PN7	
	PE5	
	P11	

Table 23.6 Correspondence between GPTW Pins

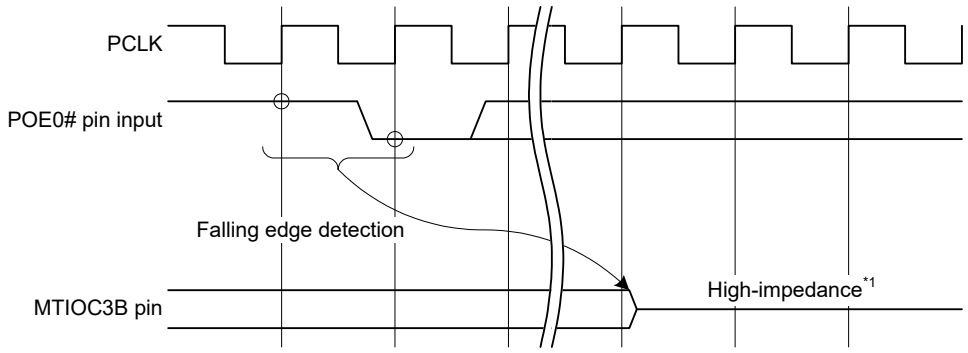
GPTW Pin Functions	Corresponding Ports	Select Registers
GTIOC0A	P71	G0SELR
	PD7	
	PD2	
GTIOC0B	P74	
	PD6	
	PD1	
GTIOC1A	P72	G1SELR
	PD5	
	PD0	
GTIOC1B	P75	
	PD4	
	PB7	
GTIOC2A	P73	G2SELR
	PD3	
	PB6	
GTIOC2B	P76	
	PD2	
	PB5	
GTIOC3A	P32	G3SELR
	PD1	
	PE5	
	PD7	
	PB6	
	P10	
GTIOC3B	P33	
	PD0	
	P11	
	PD6	
	PB5	
GTIOC4A	P95	G4SELR
	P71	
GTIOC4B	P92	
	P74	
GTIOC5A	P94	G5SELR
	P72	
GTIOC5B	P91	
	P75	
GTIOC6A	P93	G6SELR
	P73	
GTIOC6B	P90	
	P76	
GTIOC7A	P95	G7SELR
	PB2	
	P32	
	PD5	
GTIOC7B	P92	
	PB1	
	P33	
	PD3	

23.3.2 Input-Level Detection Operation

If the input conditions set by ICSR1 to ICSR5, ICSR7, and ICSR8 occur on the POE0#, POE4#, POE8#, POE10#, POE11#, POE12#, and POE9# pins, the outputs of the MTU complementary PWM output pins (MTU3 and MTU4 or MTU6 and MTU7), MTU0, MTU9 pins, and GPTW pins are disabled.

(1) Edge Detection

When the input levels to the POE0#, POE4#, POE8#, POE10#, POE11#, POE12#, and POE9# pins change from high to low (when the INV bit is 0) or low to high (when the INV bit is 1), outputs of the MTU complementary PWM output pins, MTU0 pins, MTU9 pins, and GPTW pins are disabled.
The edge is detected after the level is sampled with PCLK. Input a low or high level for at least one PCLK clock to the POE0#, POE4#, POE8#, POE10#, POE11#, POE12#, and POE9# pins.
Figure 23.12 shows a sample timing after the level changes in input to the POE0#, POE4#, POE8#, POE10#, POE11#, POE12#, and POE9# pins until the respective pins become high-impedance.



Note 1. Other pins also become high-impedance at the same time.

Figure 23.12 Operation when A Falling Edge Detection and High-Impedance are Selected

(2) Level Detection

Figure 23.13 shows an example of changing pins to high-impedance state while a low-level detection is selected (the INV bit is 0). When low is consecutively detected for the specified number of times with the sampling clock set in the ICSR1 to ICSR5, ICSR7, and ICSR8 registers, it is considered that low has been input, and the outputs of the MTU complementary PWM output pins, MTU0 pins, MTU9 pins, and GPTW pins are disabled. At this time, if a high is detected even once, it is not considered that low has been input.
The timing when the outputs of the MTU complementary PWM output pins, MTU0 pins, MTU9 pins, and GPTW pins are disabled after the sampling clock is input is the same in both edge detection and in level detection.

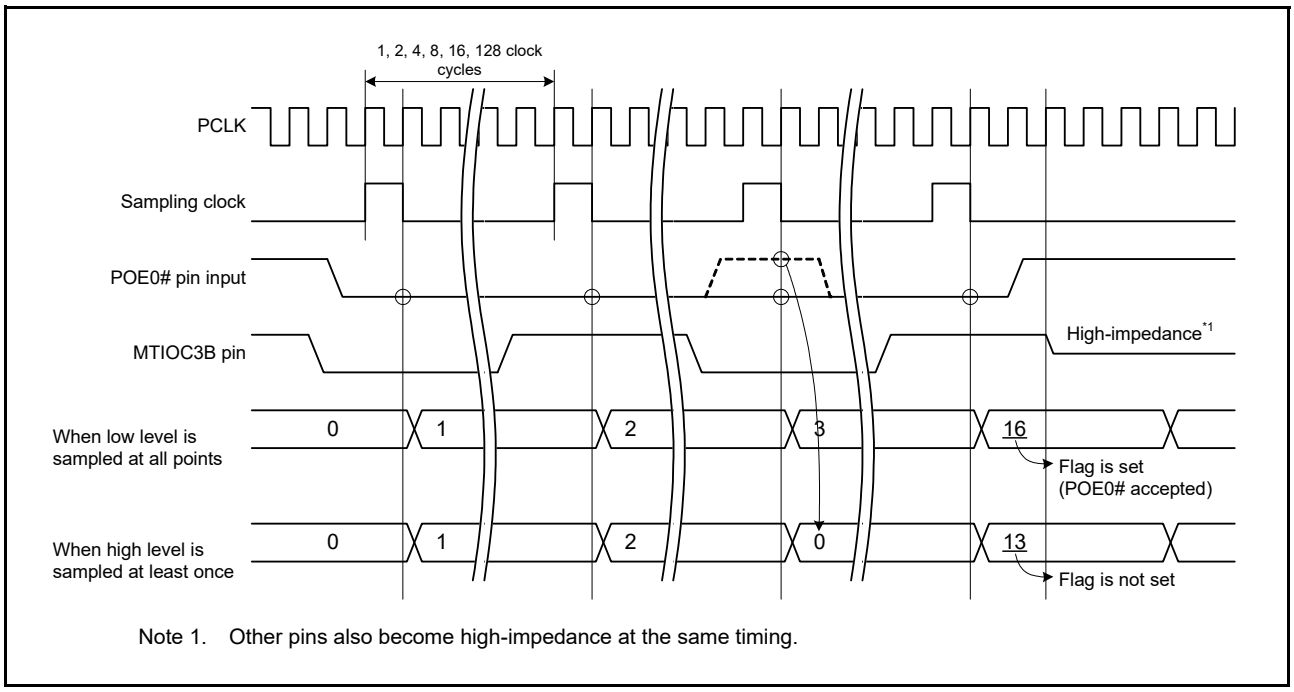


Figure 23.13 Operation when A Low-Level Detection and High-Impedance are Selected (16 samplings)

23.3.3 Output-Level Compare Operation

Figure 23.14 shows an example of the output-level compare operation for the combination of MTIOC3B and MTIOC3D. When the MTIOC3B and MTIOC3D signals simultaneously go to the active level for at least one cycle of PCLK, the outputs are disabled.
The operation is the same for the other signal combinations and switching to general I/O ports.

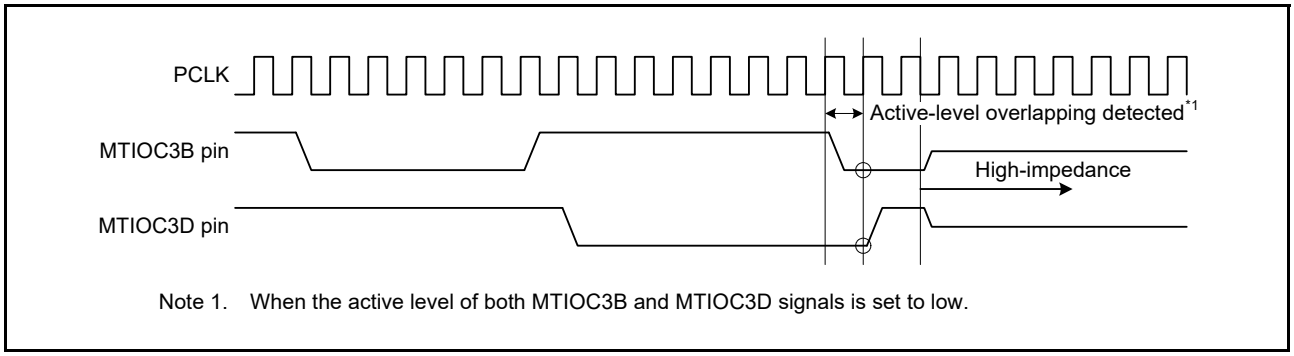


Figure 23.14 Output-Level Compare Operation

23.3.4 Output Disabling Control Using Registers

The output disabling request of the MTU pins (MTU0, MTU3, MTU4, MTU6, MTU7, and MTU9) and the GPTW pins (GPTW0 to GPTW7) can be directly controlled by using the SPOER register.
For instance, setting the SPOER.MTUCH34HIZ bit to 1 switches the MTU3 and MTU4 pins specified by the POECR2 register to the high-impedance state.
The output disabling request of other pins can also be controlled by setting the appropriate bits in the SPOER register.

23.3.5 Output Disabling Control through Detection of Oscillation Stop

When the oscillation stop detection function for the clock generation circuit detects the oscillation stop while the ICSR6.OSTSTE bit is 1, the MTU complementary PWM output pins, MTU0 pins, MTU9 pins, and GPTW pins specified by the registers POECR1 to POECR3 and POECR7 can be switched to high-impedance, and the MTU complementary PWM output pins, MTU0 pins, MTU9 pins, and GPTW pins specified by the registers PMMCR0 to PMMCR2 can be switched to a general I/O port.

23.3.6 Output Disabling Control through Detection of the Comparator Output

The outputs of the MTU complementary PWM output pins, MTU0 pins, MTU9 pins, and GPTW pins can be disabled in response to detection of the output of the comparator.
For instance, when the POECMPFR.CjFLAG flag (j = 0 to 5) is added to the output disabling control conditions for the MTU3 and MTU4 pins by setting the POECR4.CMADDMT34ZE bit to 1, the MTU3 and MTU4 pins specified by the POECR2 register become high-impedance and the MTU3 and MTU4 pins specified by the PMMCR1 register are switched to general I/O port pins on comparator output detection.
The output disabling control of other pins can be controlled by the POECR1 to POECR8 registers and the PMMCR0 to PMMCR2 registers.
When using the CjFLAG flag (j = 0 to 5) as a source for the output disabling control conditions, set the POECMPSEL or POECMPEXm register (m = 0 to 8). When the POECMPSEL.POEREQj bit is set to 1, the CjFLAG flag is used as a source for all target pins. When the POECMPEXm.POEREQj bit is set to 1, the CjFLAG flag is used as a source for the corresponding pins.

23.3.7 Additional Functions for Output Disabling Control

Output disabling control conditions for the MTU complementary PWM output pins, MTU0 pins, MTU9 pins, and GPTW pins can be added by setting the POECR4 to POECR6, POECR4B, POECR6B, and POECR8 to POECR11 registers.
For instance, the settings listed below can be added as output disabling control conditions for the MTU3 and MTU4 pins.

- Setting the POECR4.CMADDMT34ZE bit to 1 adds comparator output detection
- Setting the POECR4.IC1ADDMT34ZE bit to 1 and adds the input-level detection by the POE0# pin
- Setting the POECR4.IC2ADDMT34ZE bit to 1 and adds the input-level detection by the POE4# pin
- Setting the POECR4.IC3ADDMT34ZE bit to 1 and adds the input-level detection by the POE8# pin
- Setting the POECR4.IC4ADDMT34ZE bit to 1 and adds the input-level detection by the POE11# pin
- Setting the POECR4.IC5ADDMT34ZE bit to 1 and adds the input-level detection by the POE8# pin
- Setting the POECR4.IC6ADDMT34ZE bit to 1 and adds the input-level detection by the POE12# pin
- Setting the POECR4.IC8ADDMT34ZE bit to 1 and adds the input-level detection by the POE9# pin

The output disabling control of other pins can also be controlled by setting the appropriate bits in the POECR4 to POECR6, POECR4B, POECR6B, and POECR8 to POECR11 registers.

23.3.8 Masking of Output Disabling Control Signals

Output disabling requests from the POE0#, POE4#, POE8#, POE10#, POE11#, POE12#, and POE9# signals and the comparator level detection signals can be masked by the IMCRn registers (n = 0 to 6, 9 to 14).
When the signal selected by the IMCRn.POEmMS[5:0] bits (n = 0 to 6; m = 0, 4, 8, 10, 11, 12, 9) is high, the input signal from the POEm# pin is masked. When the signal selected by the IMCRn.CMPjMS[5:0] bits (n = 9 to 14; j = 0 to 5) is high, the comparator level detection signal is masked.
Figure 23.15 shows sample timing for input signal masking when falling-edge detection is selected.

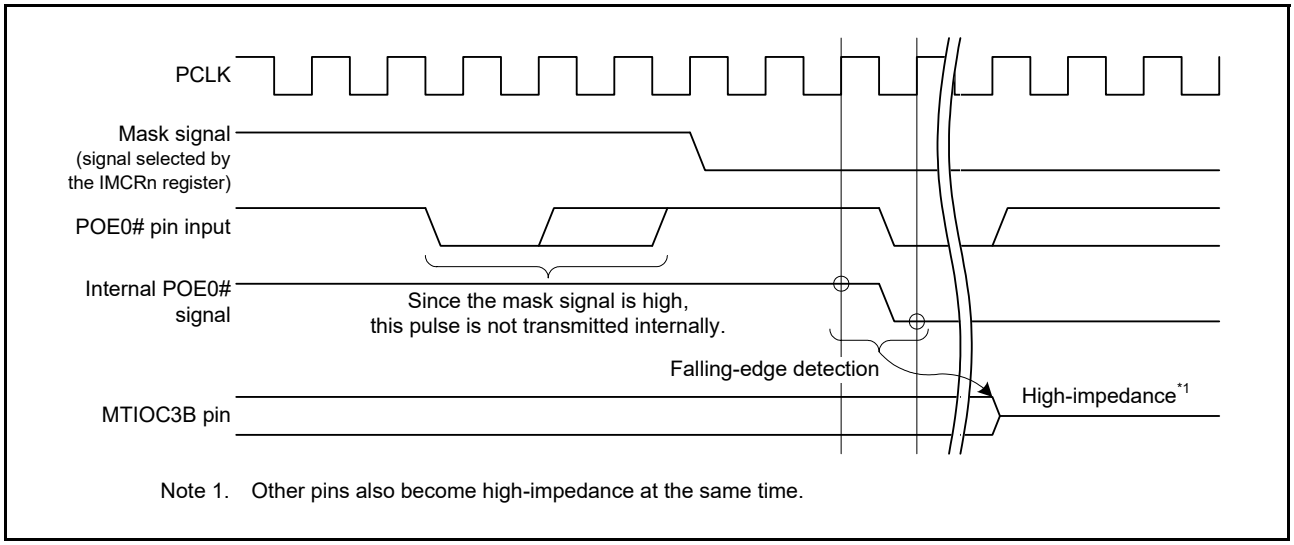


Figure 23.15 POEm# Signal Masking Operation (When Falling-Edge Detection is Selected)

23.3.9 Control in Response to an Output Disabling Request

When a request to disable outputs is generated, those pins for which the corresponding bits of the POECR1 to POECR3 and POECR7 registers is set to 1 become high-impedance and those pins for which the corresponding bits of the PMMCR0 to PMMCR2 registers are set to 1 are switched to general I/O port pins.
When both bits are set to 1 for the same pin, the settings of the POECR1 to POECR3 and POECR7 registers take priority and the pins become high-impedance.
After a pin is switched to a general I/O port pin, the settings of the corresponding bits in the PDR and PODR registers determine the state of the pin.

23.3.10 Recover from Output Disabled State

The outputs which have been disabled due to input-level detection can be recovered from the state either by returning them to their initial state with a reset, or by clearing all of the ICSR1.POE0F, ICSR2.POE4F, ICSR3.POE8F, ICSR4.POE10F, ICSR5.POE11F, ICSR7.POE12F, and ICSR8.POE9F flags. However, note that when low-level detection is selected with the ICSR1.POE0M[3:0], ICSR2.POE4M[3:0], ICSR3.POE8M[3:0], ICSR4.POE10M[3:0], ICSR5.POE11M[3:0], ICSR7.POE12M[3:0], and ICSR8.POE9M[3:0], just writing 0 to a flag is ignored (the flag is not set to 0); flags can be cleared by writing 0 to it only after a high level is input to the POE0#, POE4#, POE8#, POE10#, POE11#, POE12#, and POE9# pins and is detected. Similarly, when high-level detection is selected, the flag can be cleared by writing 0 to it only after a low level is input to the pin and is detected.
Pins whose outputs are disabled by detecting a simultaneous conduction can be released by resetting to return to the initial state or by writing 0 to the OCSR1.OSF1 flag, OCSR2.OSF2 flag, OCSR3.OSF3 flag, OCSR4.OSF4 flag, and OCSR5.OSF5 flag. However, note that just writing 0 to a flag is ignored (the flag is not set to 0); the flags can be cleared by writing 0 to it only after setting the signals to the inactive level. In the MTU, the inactive level (initial output level)

can be output by stopping the count operation. In the GPTW, the inactive level can be output in accordance with the procedure described in section 24.9.2, Pin Initialization Due to Error during Operation.

The outputs which have been disabled due to comparator output detection can be recovered from the state either by returning them to their initial state with a reset or by setting the POECMPFR.CjFLAG flag (j = 0 to 5) to 0.

When setting the POECMPFR.CjFLAG flag to 0, be sure to confirm that the analog input signal that triggered comparator output detection has returned to a normal value by performing A/D conversion and so on.

Note that the above POECMPFR.CjFLAG flag is not set to 1 again in the following cases:

- This flag is cleared without confirmation that the analog input signal has returned to a normal value, and
  1. the analog input signal remains above the reference voltage when the comparator is set to non-inverted output, or
  2. the analog input signal remains below the reference voltage when the comparator is set to inverted output.

The outputs which have been disabled due to oscillation stop detection can be recovered from the state either by returning them to their initial state with a reset or by setting the SYSTEM.OSTDSR.OSTDF flag to 0 to set the ICSR6.OSTSTF flag to 0.

23.4 POE Setting Procedure

Figure 23.16 shows the procedure for setting the POE. It illustrates an example of high-impedance control in response to comparison of the output levels on the MTU3 signals (MTIOC3B/MTIOC3D). In the figure, the MTIOC3B signal is assigned to P71 pin and MTIOC3D signal to P74 pin.

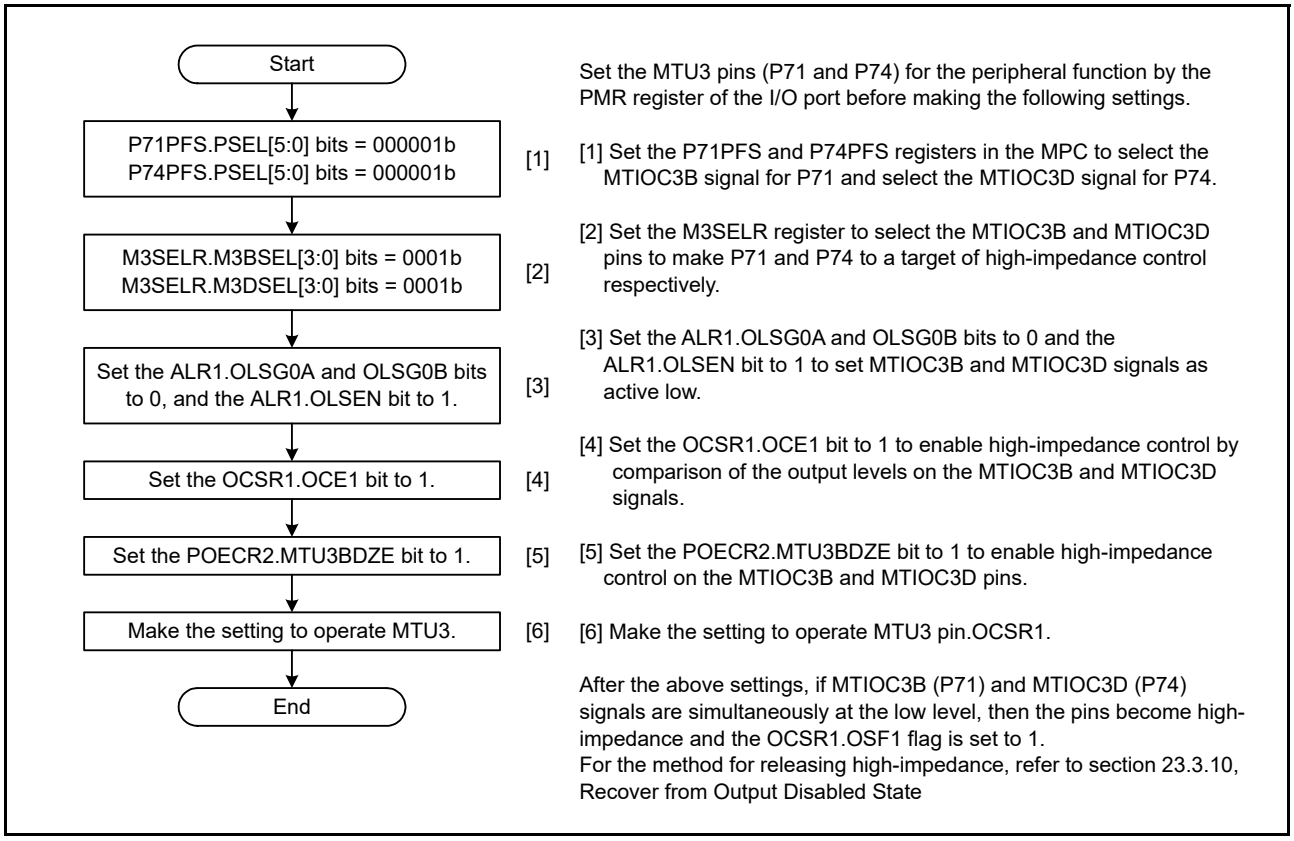


Figure 23.16 Procedure for Setting the POE

23.5 Interrupts

The POE issues a request to generate an interrupt when the specified condition is satisfied during input-level detection or output-level comparison. Table 23.7 shows the interrupt sources and their conditions.

Table 23.7 Interrupt Sources and Conditions

Name	Interrupt Source	Interrupt Flag	Condition
OEI1	Output enable interrupt 1	POE0F, OSF1	When the ICSR1.POE0F flag becomes 1 while the ICSR1.PIE1 bit is 1 or when the OCSR1.OSF1 flag becomes 1 while the OCSR1.OIE1 bit is 1
OEI2	Output enable interrupt 2	POE4F, OSF2	When the ICSR2.POE4F flag becomes 1 while the ICSR2.PIE2 bit is 1 or when the OCSR2.OSF2 flag becomes 1 while the OCSR2.OIE2 bit is 1
OEI3	Output enable interrupt 3	POE8F, POE9F	When the ICSR3.POE8F flag becomes 1 while the ICSR3.PIE3 bit is 1 or when the ICSR8.POE9F flag becomes 1 while the ICSR8.PIE8 bit is 1
OEI4	Output enable interrupt 4	POE10F, POE11F	When the ICSR4.POE10F flag becomes 1 while the ICSR4.PIE4 bit is 1 or when the ICSR5.POE11F flag becomes 1 while the ICSR5.PIE5 bit is 1
OEI5	Output enable interrupt 5	POE12F, OSF3, OSF4, OSF5	When the ICSR7.POE12F flag becomes 1 while the ICSR7.PIE7 bit is 1, when the OCSR3.OSF3 flag becomes 1 while the OCSR3.OIE3 bit is 1, when the OCSR4.OSF4 flag becomes 1 while the OCSR4.OIE4 bit is 1 or when the OCSR5.OSF5 flag becomes 1 while the OCSR5.OIE5 bit is 1

23.6 Usage Notes

23.6.1 Transition to Low Power Consumption Mode

When using the POE, do not transition to software standby mode. Output disabling control for pins cannot be performed as the POE operation is disabled in software standby mode.

23.6.2 Output Disabling Control When the MTU and GPTW Pins are Not Selected

When the control conditions are satisfied while the output disabling control for the MTU and GPTW pins are enabled by the POECR1 to POECR3 and POECR7 registers, and the PMMCR0 to PMMCR2 registers, outputs are disabled for the pins which have become controlled by the pin select register settings even when the MTU and GPTW functions are not selected by the PxyPFS register in the MPC.
To avoid unintended output disabling, ensure that there are no differences between the settings for MTU and GPTW pin selection in the PxyPFS registers of the MPC and for MTU and GPTW pin selection in the pin select register of the POE.

23.6.3 When the POE is Not Used

The output disabling control for the pins by the POE becomes enabled for some pins after resetting. Even when the POE is not used, write 0 to a relevant bit in the POECR2 register. Also write 0000b to relevant bits in the following pin select registers: M0SELR1, M0SELR2, M3SELR, M4SELR1, M4SELR2, M6SELR, M7SELR1, M7SELR2, M9SELR1, M9SELR2, G0SELR, G1SELR, G2SELR, G3SELR, G4SELR, G5SELR, G6SELR, and G7SELR.

23.6.4 Setting the Active Level When the Inverting Output Is Set for the MTU and GPTW

In this MCU, either of normal output or inverting output can be selected for an output of the MTU and GPTW by the MPC.PxyPFS register.
When inverting output of the MTU is selected, the active level specified by the MTU.TOCR1j and MTU.TOCR2j registers (j = A, B) and the active level of the signal which is output to the pins become inverted. At this time, to use detection of simultaneous conduction, specify, by the ALR1 and ALR2 registers, the active level with reference to signals which are output to the pins.
When the inverting output of the GPTW is selected, the active level of the signal which is output to the pins become inverted. At this time, to use detection of simultaneous conduction, specify, by the registers ALR3 to ALR5, the active level with reference to signals which are output to the pins.

23.6.5 Reading the Pins When in High-Impedance State

When the pins are in high-impedance state by the POE, the level of the relevant pins cannot be read. The value when read is indeterminable. To read the level of the pins, release from the high-impedance state.
When the port switching control is selected instead of high-impedance control, this is not applied.

23.6.6 Note for Using POE and POEG Together

When using the POE and POEG together, do not control the output disabling for the same GPTW output pin by both the POE and POEG.

23.6.7 Notes on Using POE with the INV Bit Set to 1

When using the POE with the ICSRn.INV bit (n = 1 to 5, 7, 8) set to 1 (rising edge or high-level detection), it is expected that low is input to the POEm# pin when the reset is released. When the POEm# signal is selected by the MPC.PxyPFS register in this state, the internal signal changes from high to low, and an output disabling request is generated. Set the ICSRn register first and then the PxyPFS register.

Depending on the settings of the ICSRn.POEmM[3:0] and POEmM2[3:0] bits, an output disabling request may be generated between the settings of the ICSRn register and the PxyPFS register. In this case, clear the corresponding ICSRn.POEmF flag according to section 23.3.10, Recover from Output Disabled State.

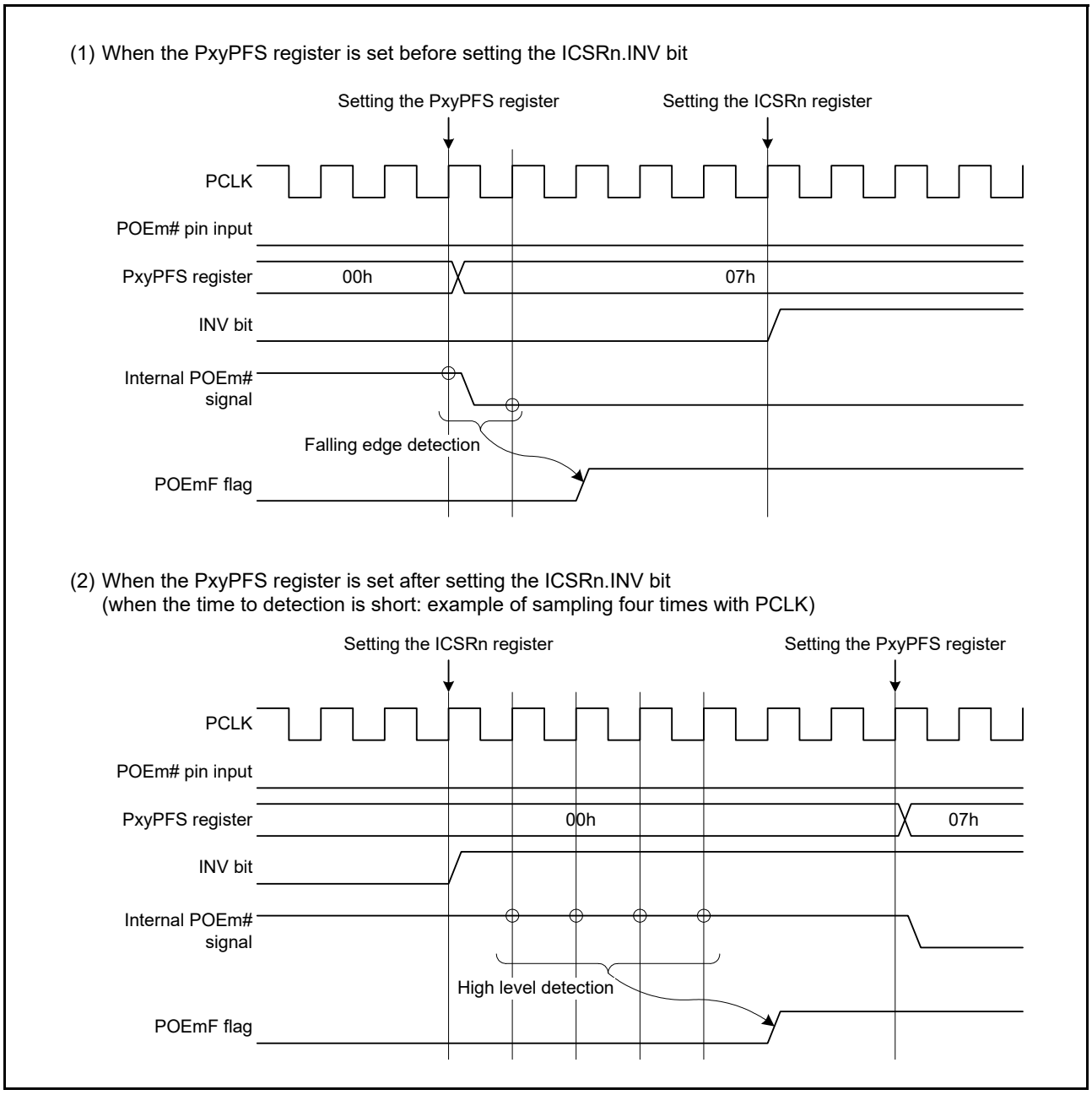


Figure 23.17 Timing to Set ICSRn.INV Bit and PxyPFS Register

24. General PWM Timer (GPTWa)

This MCU has a general PWM timer (GPTW) consisting of a eight-channel 32-bit timer.

24.1 Overview

Table 24.1 lists the specifications for the GPTW, and Table 24.2 shows the functions of the GPTW. Figure 24.1 and Figure 24.2 show block diagrams of the GPTW.

Table 24.1 GPTW Specifications

Item	Description
Functions	• 32 bits × 8 channels (products with 64 Kbytes of RAM) • 16 bits × 8 channels (products with 48 Kbytes of RAM) • Up-counting or down-counting (sawtooth waves) or up/down-counting (triangle waves) for each counter. • Clock sources independently selectable for each channel • Two I/O pins per channel • Two output compare/input capture registers per channel • For the two output compare/input capture registers of each channel, four registers are provided as buffer registers and are capable of operating as comparison registers when buffering is not in use. • In output compare operation, buffer switching can be at crests or troughs, enabling the generation of laterally asymmetric PWM waveforms. • Registers for setting up frame cycles in each channel (with capability for generating interrupts at overflow or underflow) • Generation of dead times in PWM operation • Generation of high accuracy duty in the vicinity of duty 0% and 100% PWM waveform • In output compare operation, setting compare register is immediately used to generate PWM waveform with dead times • Simultaneous start/stop/clearing of desired channel counters • Operation of count start/count stop/counter clearing/up-counting/down-counting/input capture by maximum of eight ELC events based on the ELC setting • Operation of count start/count stop/counter clearing/up-counting/down-counting/input capture by detecting two input signal conditions • Operation of count start/count stop/counter clearing/up-counting/down-counting/input capture by maximum of four external triggers • Function to control output negation by dead time error and requests for disabling of output from the POEG • A/D conversion start trigger generation function • Event signals for compare match A to F and for overflow/underflow can be output to the ELC • Input capture input can select noise filter function • Cycle count function • External pulse width measuring function • Logical operation between the channel output • Synchronous counter clearing/counter setting/input capture among channels • Bus clock: PCLKA, GPTW count reference clock: PCLKC, Frequency ratio between PCLKA : PCLKC = 1 : 1, 1 : 2

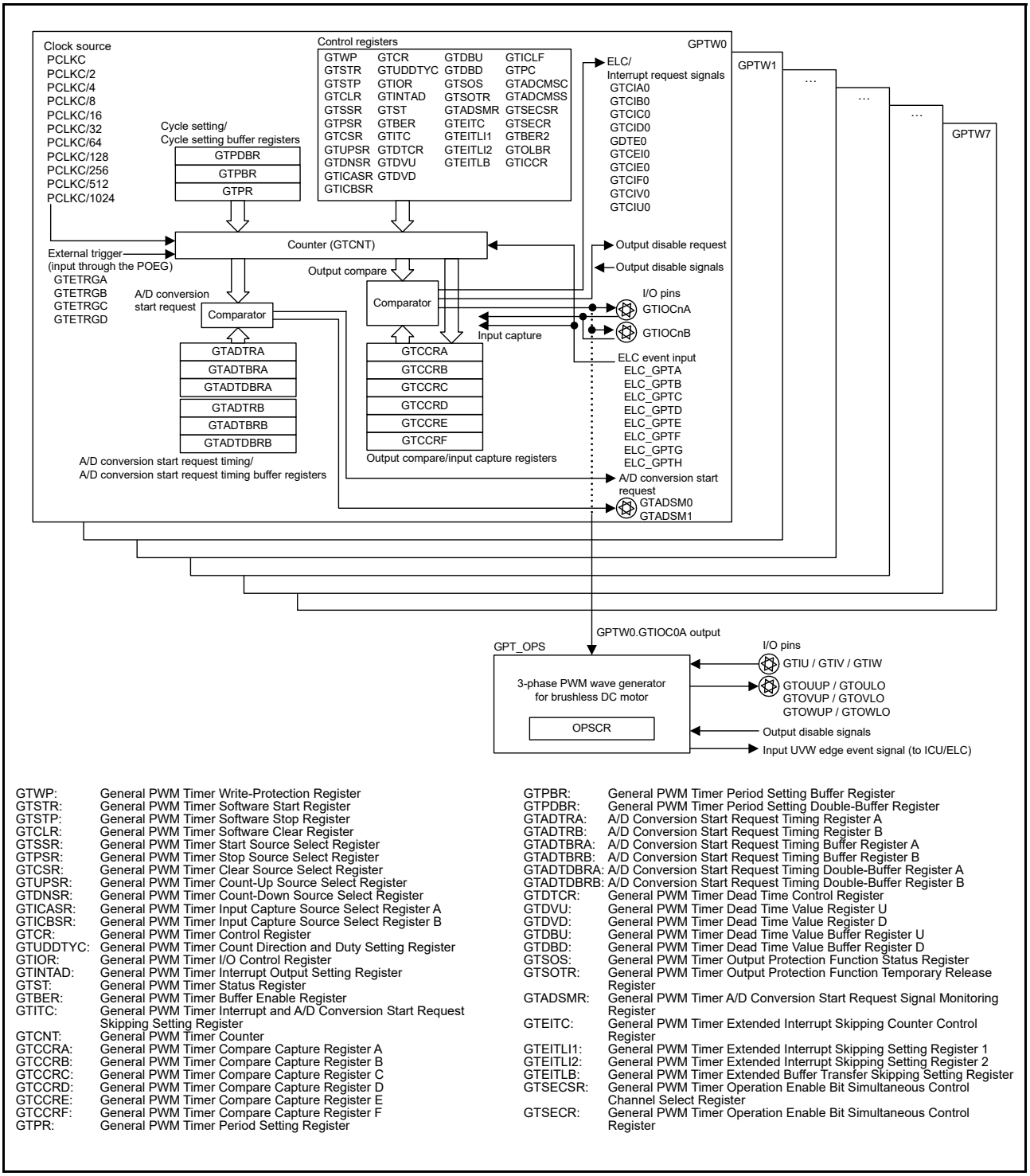
Table 24.2 GPTW Functions (1/2)

Item		GPTW0 to GPTW2	GPTW3	GPTW4 to GPTW6	GPTW7
Timer counter		32-bit / 16-bit			
Count clocks		PCLKC, PCLKC/2, PCLKC/4, PCLKC/8, PCLKC/16, PCLKC/32, PCLKC/64, PCLKC/128, PCLKC/256, PCLKC/512, PCLKC/1024, GTETRGA, GTETRGB, GTETRGC, GTETRGD			
Output compare/input capture registers (GTCCR)		GTCCRA, GTCCRB			
Compare/buffer registers		GTCCRC, GTCCRD, GTCCRE, GTCCRF			
PWM period setting register		GTPR			
PWM period setting buffer registers		GTPBR, GTPDBR			
Input capture input/ compare match output/ PWM output pin		GTIOCnA, GTIOCnB			
External trigger input pin (via the POEG)		GTETRGA, GTETRGB, GTETRGC, GTETRGD			
Counter clear sources		Compare match for the GTPR register, input capture, ELC input, the state of input pins, external trigger input, GTCCR register compare match, and Other channel's counter clear sources			
Cycle count function		✓			
Compare match output	Low output	✓			
	High output	✓			
	Toggle output	✓			
Input capture function		✓			
Automatic addition of dead time		✓			
PWM mode	Sawtooth-wave PWM mode 1	✓	✓	✓	✓
	Sawtooth-wave PWM mode 2	✓	—	✓	—
	Sawtooth-wave one-shot pulse mode	✓	✓	✓	✓
	Triangle-wave PWM mode 1/2/3	✓	✓	✓	✓
	Complementary PWM mode 1/2/3/4	✓	—	✓	—
Phase counting function		✓			
External pulse width measuring function		✓			
Buffer operation	Double buffer	✓			
	Simultaneous operation disable control for multiple channels	✓			
	Buffer transfer by counter clear / compare match	✓			
One-shot operation		✓			
DMAC/DTC activation		All interrupt sources			
A/D conversion start trigger		Compare match of GTADTRA or GTADTRB register			
3-phase PWM wave generator for brushless DC motor		✓			

Table 24.2 GPTW Functions (2/2)

Item	GPTW0 to GPTW2	GPTW3	GPTW4 to GPTW6	GPTW7
Interrupt sources	10 sources • GTCCRA register compare match/input capture (GTCIAn) • GTCCRB register compare match/input capture (GTCIBn) • GTCCRC register compare match (GTCICn) • GTCCRD register compare match (GTCIDn) • Dead time error (GDTEn) • GTCCRE register compare match (GTCIEn) • GTCCRF register compare match (GTCIFn) • GTCNT counter overflow (GTPR register compare match) (GTCIVn) • GTCNT counter underflow (GTCIUn) • Cycle count end (GTCEIn)			
Interrupt skipping function	• Skipping of interrupts of GTCNT counter overflow (GTPR register compare match) (GTCIVn) and GTCNT counter underflow (GTCIUn) (interlocked with other interrupts and A/D conversion start requests) • Skipping of GTADTRA and GTADTRB register compare match			
Event operation by the ELC	✓			
Noise filter function	✓			
Logical operation between the channel output	✓			
Synchronous counter clearing/counter setting/ input capture	✓			

✓: Possible



GTWP: General PWM Timer Write-Protection Register
GTSTR: General PWM Timer Software Start Register
GTSTP: General PWM Timer Software Stop Register
GTCLR: General PWM Timer Software Clear Register
GTSSR: General PWM Timer Start Source Select Register
GTPSR: General PWM Timer Stop Source Select Register
GTCSR: General PWM Timer Clear Source Select Register
GTUPSR: General PWM Timer Count-Up Source Select Register
GTDNSR: General PWM Timer Count-Down Source Select Register
GTICASR: General PWM Timer Input Capture Source Select Register A
GTICBSR: General PWM Timer Input Capture Source Select Register B
GTCR: General PWM Timer Control Register
GTUDDTYC: General PWM Timer Count Direction and Duty Setting Register
GTIOR: General PWM Timer I/O Control Register
GTINTAD: General PWM Timer Interrupt Output Setting Register
GTST: General PWM Timer Status Register
GTBER: General PWM Timer Buffer Enable Register
GTITC: General PWM Timer Interrupt and A/D Conversion Start Request Skipping Setting Register
GTCNT: General PWM Timer Counter
GTCCRA: General PWM Timer Compare Capture Register A
GTCCRB: General PWM Timer Compare Capture Register B
GTCCRC: General PWM Timer Compare Capture Register C
GTCCRD: General PWM Timer Compare Capture Register D
GTCCRE: General PWM Timer Compare Capture Register E
GTCCRF: General PWM Timer Compare Capture Register F
GTPR: General PWM Timer Period Setting Register

GTPBR: General PWM Timer Period Setting Buffer Register
GTPDBR: General PWM Timer Period Setting Double-Buffer Register
GTADTRA: A/D Conversion Start Request Timing Register A
GTADTRB: A/D Conversion Start Request Timing Register B
GTADTBRA: A/D Conversion Start Request Timing Buffer Register A
GTADTBRB: A/D Conversion Start Request Timing Buffer Register B
GTADTDBRA: A/D Conversion Start Request Timing Double-Buffer Register A
GTADTDBRB: A/D Conversion Start Request Timing Double-Buffer Register B
GTDTCR: General PWM Timer Dead Time Control Register
GTDVU: General PWM Timer Dead Time Value Register U
GTDVD: General PWM Timer Dead Time Value Register D
GTDBU: General PWM Timer Dead Time Value Buffer Register U
GTDBD: General PWM Timer Dead Time Value Buffer Register D
GTSOS: General PWM Timer Output Protection Function Status Register
GTSOTR: General PWM Timer Output Protection Function Temporary Release Register
GTADSMR: General PWM Timer A/D Conversion Start Request Signal Monitoring Register
GTEITC: General PWM Timer Extended Interrupt Skipping Counter Control Register
GTEITLI1: General PWM Timer Extended Interrupt Skipping Setting Register 1
GTEITLI2: General PWM Timer Extended Interrupt Skipping Setting Register 2
GTEITLB: General PWM Timer Extended Buffer Transfer Skipping Setting Register
GTSECSR: General PWM Timer Operation Enable Bit Simultaneous Control Channel Select Register
GTSECR: General PWM Timer Operation Enable Bit Simultaneous Control Register

Figure 24.1 GPTW Block Diagram (Sawtooth-wave PWM mode, Sawtooth-wave one-shot pulse mode, and Triangle-wave PWM mode 1/2/3)

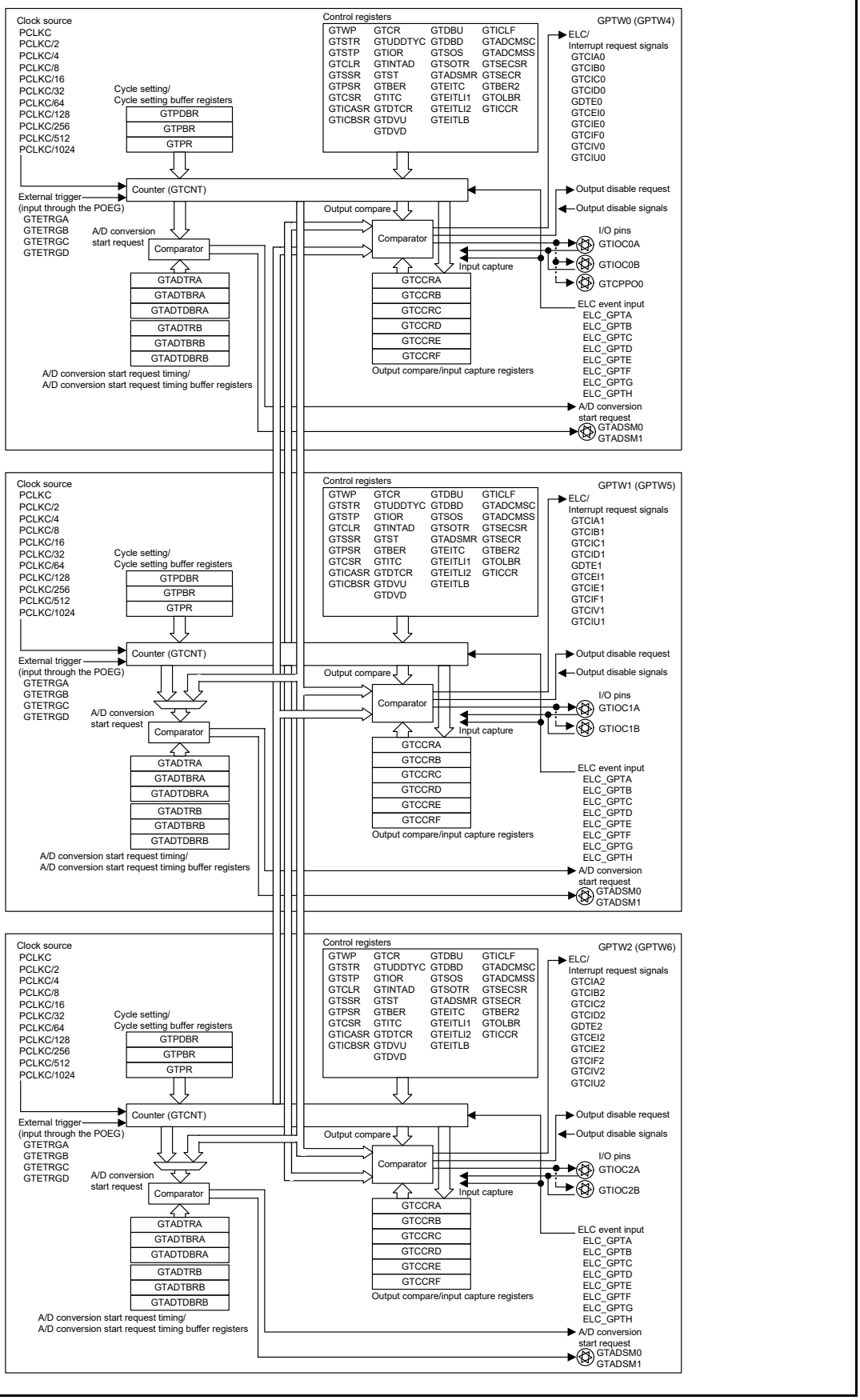


Figure 24.2 GPTW Block Diagram (Sawtooth-wave PWM mode 1/2/3, Sawtooth-wave one-shot pulse mode, Triangle-wave PWM mode 1/2/3, and Complementary PWM 1/2/3/4)

In this specification, three consecutive channels to configure complementary PWM mode is defined as complementary PWM mode channel group. The lowest position channel of complementary PWM mode channel group is defined as master channel. The second channel is defined as slave channel 1. The highest position channel is defined as slave channel 2.

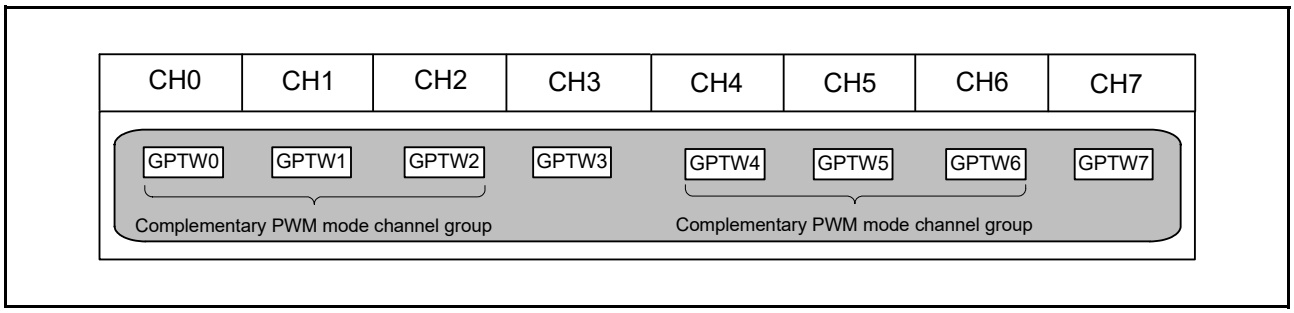


Figure 24.3 Association between GPTW channels and module names

Table 24.3 lists the I/O pins used in the GPTW.

Table 24.3 GPTW I/O Pins (n = 0 to 7)

Channel	Pin Name	I/O	Function
GPTW	GTETRGA	Input	External trigger input pin A (input via the POEG)
	GTETRGB	Input	External trigger input pin B (input via the POEG)
	GTETRGC	Input	External trigger input pin C (input via the POEG)
	GTETRGD	Input	External trigger input pin D (input via the POEG)
	GTADSM0	Output	A/D conversion start request monitor 0 output pin
	GTADSM1	Output	A/D conversion start request monitor 1 output pin
GPTWn	GTIOCnA	I/O	GTCCRA register input capture input/output compare output/PWM output pin
	GTIOCnB	I/O	GTCCRB register input capture input/output compare output/PWM output pin
	GTCPPOn	Output	Toggle output synchronized with PWM period
OPS	GTIU	Input	Hall sensor input pin U
	GTIV	Input	Hall sensor input pin V
	GTIW	Input	Hall sensor input pin W
	GTOUUP	Output	3-phase PWM output for BLDC motor control (positive U-phase)
	GTOULO	Output	3-phase PWM output for BLDC motor control (negative U-phase)
	GTOVUP	Output	3-phase PWM output for BLDC motor control (positive V-phase)
	GTOVLO	Output	3-phase PWM output for BLDC motor control (negative V-phase)
	GTOWUP	Output	3-phase PWM output for BLDC motor control (positive W-phase)
	GTOWLO	Output	3-phase PWM output for BLDC motor control (negative W-phase)

24.2 Register Descriptions

24.2.1 General PWM Timer Write-Protection Register (GTWP)

Address(es): GPTW0.GTWP 000C 2000h, GPTW1.GTWP 000C 2100h, GPTW2.GTWP 000C 2200h, GPTW3.GTWP 000C 2300h, GPTW4.GTWP 000C 2400h, GPTW5.GTWP 000C 2500h, GPTW6.GTWP 000C 2600h, GPTW7.GTWP 000C 2700h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PRKEY[7:0]								—	—	—	CMNWP	CLRWP	STPWP	STRWP	WP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	WP	Register Write Disabled	0: Write to the register is enabled 1: Write to the register is disabled	R/W
b1	STRWP	GTSTR.CSTRT Bit Write Disabled	0: Write to the bit is enabled 1: Write to the bit is disabled	R/W
b2	STPWP	GTSTP.CSTOP Bit Write Disabled	0: Write to the bit is enabled 1: Write to the bit is disabled	R/W
b3	CLRWP	GTCLR.CCLR Bit Write Disabled	0: Write to the bit is enabled 1: Write to the bit is disabled	R/W
b4	CMNWP	Common Register Write Disabled	0: Write to the register is enabled 1: Write to the register is disabled	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	PRKEY[7:0]	GTWP Key Code	These bits are read as 0. To modify the WP, STRWP, STPWP, CLRWP, and CMNWP bits, write A5h.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

GTWP enables or disables writing to registers to prevent accidental modification.
Protection by the GTWP register only covers the CPU write operation.
Such protection does not cover a register updating generated in coordination with the CPU write.
For registers that are write enabled or disabled depending on the setting of the GTWP register, see section 24.8.1, Write-Protection for Registers.

WP Bit (Register Write Disabled)

Write enable/disable to the GPTW can be selected with this bit.
Registers which are targets of write enable/disable are as follows:
GTSSR, GTPSR, GTCSR, GTUPSR, GTDNSR, GTICASR, GTICBSR, GTCR, GTUDDTYC, GTIOR, GTINTAD, GTST, GTBER, GTITC, GTCNT, GTCCRA, GTCCRB, GTCCRC, GTCCRD, GTCCRE, GTCCRF, GTPR, GTPBR, GTPDBR, GTADTRA, GTADTBRA, GTADTDBRA, GTADTRB, GTADTBRB, GTADTDBRB, GTDTCR, GTDVU, GTDVD, GTDBU, GTDBD, GTSOS, GTSOTR, GTADSMR, GTEITC, GTEITLI1, GTEITLI2, GTEITLB, GTICLF, GTPC, GTADCMSC, GTADCMSS, GTBER2, GTOLBR, GTICCR

STRWP Bit (GTSTR.CSTRT Bit Write Disabled)

The STRWP bits enable or disable starting the updating of counter values by writing to the CSTRTn bit corresponding to a channel number in the GTSTR register.

The bit position of each CSTRTn bit in the GTSTR registers is allocated to the channel with the corresponding number, and writing to the GTSTR register for any channel results in writing to the registers of all channels. The STRWP bit for each channel does not control writing but only controls updating of the CSTRT bit for the corresponding channel in the case of simultaneous writing to all channels.
Thus, in the case of writing to the CSTRT bits of a channel for which the setting of the STRWP bit is 1 (disabling writing), the CSTRT bit for the given channel will not be updated, but the CSTRT bits corresponding to channel for which the setting of the STRWP bit is 0 (enabling writing) will be updated. For example, when the setting of the GPTW0.GTWP.STRWP bit is 0 (enabling writing), writing 1 to the GPTW1.GTSTR.CSTRT0 bit when its current setting is 0 will lead to updating of the value, and the GPTW0.GTCNT counter starts running. When the setting of the GPTW0.GTWP.STRWP bit is 1 (disabling writing), writing 1 to the GPTW1.GTSTR.CSTRT0 bit when its current setting is 0 will leave the bit with the value 0, and the GPTW0.GTCNT counter will not start running.
If you want to protect all bits in the GTSTR register from being updated, set the STRWP bits of all channels to 1.

STPWP Bit (GTSTP.CSTOP Bit Write Disabled)

The STPWP bits enable or disable starting the updating of counter values by writing to the CSTOPn bit corresponding to a channel number in the GTSTP register.
The bit position of each CSTOPn bit in the GTSTP registers is allocated to the channel with the corresponding number, and writing to the GTSTP register for any channel results in writing to the registers of all channels. The STPWP bit for each channel does not control writing but only controls updating of the CSTOP bit for the corresponding channel in the case of simultaneous writing to all channels.
Thus, in the case of writing to the CSTOP bits of a channel for which the setting of the STPWP bit is 1 (disabling writing), the CSTOP bit for the given channel will not be updated, but the CSTOP bits corresponding to channel for which the setting of the STPWP bit is 0 (enabling writing) will be updated. For example, when the setting of the GPTW0.GTWP.STPWP bit is 0 (enabling writing), writing 1 to the GPTW1.GTSTP.CSTOP0 bit when its current setting is 0 will lead to updating of the value, and the GPTW0.GTCNT counter is stopped. When the setting of the GPTW0.GTWP.STPWP bit is 1 (disabling writing), writing 1 to the GPTW1.GTSTP.CSTOP0 bit when its current setting is 0 will leave the bit with the value 0, and the GPTW0.GTCNT counter will not be stopped.
If you want to protect all bits in the GTSTP register from being updated, set the STPWP bits of all channels to 1.

CLRWP Bit (GTCLR.CCLR Bit Write Disabled)

The CLRWP bits enable or disable starting the updating of counter values by writing to the CCLRn bit corresponding to a channel number in the GTCLR register.
The bit position of each CCLRn bit in the GTCLR registers is allocated to the channel with the corresponding number, and writing to the GTCLR register for any channel results in writing to the registers of all channels. The CLRWP bit for each channel does not control writing but only controls updating of the CCLR bit for the corresponding channel in the case of simultaneous writing to all channels.
Thus, in the case of writing to the CCLR bits of a channel for which the setting of the CLRWP bit is 1 (disabling writing), the CCLR bit for the given channel will not be updated, but the CCLR bits corresponding to channel for which the setting of the CLRWP bit is 0 (enabling writing) will be updated. For example, when the setting of the GPTW0.GTWP.CLRWP bit is 0 (enabling writing), writing 1 to the GPTW1.GTCLR.CCLR0 bit when its current setting is 0 will lead to updating of the value, and the GPTW0.GTCNT counter is cleared.
If you want to protect all bits in the GTCLR register from being updated, set the CLRWP bits of all channels to 1.

CMNWP Bit (Common Register Write Disabled)

The CMNWP bit enables or disables starting the updating of counter values by writing to the SECSELn bit (n = 0 to 7) corresponding to a channel number in the GTSECSR register or to the GTSECR register.
The bit position of each SECSEL bit in the GTSECSR registers is allocated to the channel with the corresponding number, and writing to the GTSECSR register for any channel results in writing to the registers of all channels. Writing to the GTSECR register of any channel leads to writing to the registers of all channels. The CMNWP bit for each channel

does not control writing but only controls updating of the SECSEL bit and the GTSECR register value for the corresponding channel in the case of simultaneous writing to all channels.
Thus, in the case of writing to the SECSEL bit and the GTSECR register value of a channel for which the setting of the CMNWP bit is 1 (disabling writing), the SECSEL bit and the GTSECR register value for the given channel will not be updated, but the SECSEL bit and the GTSECR register value corresponding to channel for which the setting of the CMNWP bit is 0 (enabling writing) will be updated.
For example, when the setting of the GPTW0.GTWP.CMNWP bit is 0 (enabling writing), writing to the GPTW1.GTSECSR.SECSEL0 bit will lead to updating of the value of the GPTW0.GTSECSR.SECSEL0 bit. In the same way, writing to the GPTW1.GTSECR register will update the value of the GPTW0.GTSECR register. When the setting of the GPTW0.GTWP.CMNWP bit is 1 (disabling writing), writing to the GPTW1.GTSECSR.SECSEL0 bit will not lead to updating of the value of the GPTW0.GTSECSR.SECSEL0 bit. In the same way, writing to the GPTW1.GTSECR register will not lead to updating of the value of the GPTW0.GTSECR register.
If you want to protect all bits in the GTSECSR and GTSECR registers from being updated, set the CMNWP bits of all channels to 1.

PRKEY[7:0] Bits (GTWP Key Code)

This bit controls whether the WP, STRWP, STPWP, CLRWP, and CMNWP bits can be overwritten.

24.2.2 General PWM Timer Software Start Register (GTSTR)

Address(es): GPTW0.GTSTR 000C 2004h, GPTW1.GTSTR 000C 2104h, GPTW2.GTSTR 000C 2204h, GPTW3.GTSTR 000C 2304h, GPTW4.GTSTR 000C 2404h, GPTW5.GTSTR 000C 2504h, GPTW6.GTSTR 000C 2604h, GPTW7.GTSTR 000C 2704h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	CSTRT7	CSTRT6	CSTRT5	CSTRT4	CSTRT3	CSTRT2	CSTRT1	CSTRT0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CSTRT0	Channel 0 Count Start	When reading 0: The counter is stopped. 1: The counter is running. When writing 0: Ignored. 1: The counter starts running.	R/W
b1	CSTRT1	Channel 1 Count Start		R/W
b2	CSTRT2	Channel 2 Count Start		R/W
b3	CSTRT3	Channel 3 Count Start		R/W
b4	CSTRT4	Channel 4 Count Start		R/W
b5	CSTRT5	Channel 5 Count Start		R/W
b6	CSTRT6	Channel 6 Count Start		R/W
b7	CSTRT7	Channel 7 Count Start		R/W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTSTR register starts the GTCNT counter operation.

A bit position of the GTSTR register indicates a channel number. The GTSTR register of each channel is a common register, and writing 1 to the GTSTR register in any channel and updating it can start operation of the GTCNT counter in all the channels related to the position of the bit written with 1. A change in counter operation and the register value is not generated by writing 0 to the bit.

In complementary PWM mode, writing to the CSTRTn bit of master channel is only valid. The bit on slave channels reflects the bit value of master channel.

CSTRTn Bit (Channel n Count Start) (n = 0 to 7)

This bit starts the GTCNT counter for channel n.

A read value indicates the state of the counter operation in a corresponding channel (GTCR.CST bit). The bit with 0 indicates that the counter is stopped, while the bit with 1 indicates that the counter is in operation.

24.2.3 General PWM Timer Software Stop Register (GTSTP)

Address(es): GPTW0.GTSTP 000C 2008h, GPTW1.GTSTP 000C 2108h, GPTW2.GTSTP 000C 2208h, GPTW3.GTSTP 000C 2308h, GPTW4.GTSTP 000C 2408h, GPTW5.GTSTP 000C 2508h, GPTW6.GTSTP 000C 2608h, GPTW7.GTSTP 000C 2708h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	CSTOP7	CSTOP6	CSTOP5	CSTOP4	CSTOP3	CSTOP2	CSTOP1	CSTOP0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	CSTOP0	Channel 0 Count Stop	When reading 0: The counter is operating. 1: The counter is stopped. When writing 0: Ignored. 1: The counter is stopped.	R/W
b1	CSTOP1	Channel 1 Count Stop		R/W
b2	CSTOP2	Channel 2 Count Stop		R/W
b3	CSTOP3	Channel 3 Count Stop		R/W
b4	CSTOP4	Channel 4 Count Stop		R/W
b5	CSTOP5	Channel 5 Count Stop		R/W
b6	CSTOP6	Channel 6 Count Stop		R/W
b7	CSTOP7	Channel 7 Count Stop		R/W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTSTP register stops the GTCNT counter operation.

A bit position of the GTSTP register indicates a channel number. The GTSTP register of each channel is a common register, and writing 1 to the GTSTP register in any channel and updating it can stop operation of the GTCNT counter in all the channels related to the position of the bit written with 1. A change in counter operation and the register value is not generated by writing 0 to the bit.

In complementary PWM mode, writing to the CSTOPn bit of master channel is only valid. The bit on slave channels reflects the bit value of master channel.

CSTOPn Bit (Channel n Count Stop) (n = 0 to 7)

This bit stops the GTCNT counter for channel n.

A read value indicates the state of the counter operation in a corresponding channel (inverted GTCR.CST bit). The bit with 0 indicates that the counter is in operation, while the bit with 1 indicates that the counter is stopped.

24.2.4 General PWM Timer Software Clear Register (GTCLR)

Address(es): GPTW0.GTCLR 000C 200Ch, GPTW1.GTCLR 000C 210Ch, GPTW2.GTCLR 000C 220Ch, GPTW3.GTCLR 000C 230Ch, GPTW4.GTCLR 000C 240Ch, GPTW5.GTCLR 000C 250Ch, GPTW6.GTCLR 000C 260Ch, GPTW7.GTCLR 000C 270Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	CCLR7	CCLR6	CCLR5	CCLR4	CCLR3	CCLR2	CCLR1	CCLR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CCLR0	Channel 0 Counter Clear	0: Ignored. 1: The counter is to be cleared.	W
b1	CCLR1	Channel 1 Counter Clear		W
b2	CCLR2	Channel 2 Counter Clear		W
b3	CCLR3	Channel 3 Counter Clear		W
b4	CCLR4	Channel 4 Counter Clear		W
b5	CCLR5	Channel 5 Counter Clear		W
b6	CCLR6	Channel 6 Counter Clear		W
b7	CCLR7	Channel 7 Counter Clear		W
b31 to b8	—	Reserved	The write value should be 0.	W

The GTCLR register is a write-only register which sets clearing of the GTCNT counter.
A bit position of the GTCLR register indicates a channel number. The GTCLR register of each channel is a common register, and writing 1 to the GTCLR register in any channel and updating it changes to 0 of the GTCNT counter in all the channels related to the position of the bit written with 1. A change in counter operation and the register value is not generated by writing 0 to the bit.
In complementary PWM mode, writing to the CCLRn bit of master channel is only valid. The bit on slave channels reflects the bit value of master channel.

CCLRn Bit (Channel n Counter Clear) (n = 0 to 7)

When the counting direction flag is set for decrementation (GTST.TUCF flag = 0) with sawtooth-wave mode selected in the GTCR.MD[2:0] or the GTCR.MD[3:0] bits, the value of the GTCNT counter becomes that of the corresponding GTPR register in response to writing 1 to the CCLRn bit. The value of the counter becomes 0000 0000h with other settings.

24.2.5 General PWM Timer Start Source Select Register (GTSSR)

Address(es): GPTW0.GTSSR 000C 2010h, GPTW1.GTSSR 000C 2110h, GPTW2.GTSSR 000C 2210h, GPTW3.GTSSR 000C 2310h, GPTW4.GTSSR 000C 2410h, GPTW5.GTSSR 000C 2510h, GPTW6.GTSSR 000C 2610h, GPTW7.GTSSR 000C 2710h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CSTRT	—	—	—	—	—	—	—	SSELCH	SSELCG	SSELCF	SSELCE	SSELCD	SSELCC	SSELCB	SSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SSCBFAH	SSCBFAL	SSCBRAH	SSCBRAL	SSCAFBH	SSCAFBL	SSCARBH	SSCARBL	SSGTRGDF	SSGTRGDR	SSGTRGCF	SSGTRGCR	SSGTRGBF	SSGTRGBR	SSGTRGAF	SSGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SSGTRGAR	GTETRGA Signal Edge Select	b1 b0 0 0: The GTETRGA signal is not used as a trigger to start counting. 0 1: The counter starts at a rising edge of the GTETRGA signal. 1 0: The counter starts at a falling edge of the GTETRGA signal. 1 1: The counter starts at both edges of the GTETRGA signal.	R/W
b1	SSGTRGAF			R/W
b2	SSGTRGBR	GTETRGB Signal Edge Select	b3 b2 0 0: The GTETRGB signal is not used as a trigger to start counting. 0 1: The counter starts at a rising edge of the GTETRGB signal. 1 0: The counter starts at a falling edge of the GTETRGB signal. 1 1: The counter starts at both edges of the GTETRGB signal.	R/W
b3	SSGTRGBF			R/W
b4	SSGTRGCR	GTETRGC Signal Edge Select	b5 b4 0 0: The GTETRGC signal is not used as a trigger to start counting. 0 1: The counter starts at a rising edge of the GTETRGC signal. 1 0: The counter starts at a falling edge of the GTETRGC signal. 1 1: The counter starts at both edges of the GTETRGC signal.	R/W
b5	SSGTRGCF			R/W
b6	SSGTRGDR	GTETRGD Signal Edge Select	b7 b6 0 0: The GTETRGD signal is not used as a trigger to start counting. 0 1: The counter starts at a rising edge of the GTETRGD signal. 1 0: The counter starts at a falling edge of the GTETRGD signal. 1 1: The counter starts at both edges of the GTETRGD signal.	R/W
b7	SSGTRGDF			R/W
b8	SSCARBL	GTIOCnA Signal Rising Edge Applying Condition Select*1	b9 b8 0 0: Rising edge of the GTIOCnA signal is not used as a trigger to start counting. 0 1: The counter starts at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven low. 1 0: The counter starts at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven high. 1 1: The counter starts at a rising edge of the GTIOCnA signal.	R/W
b9	SSCARBH			R/W

Bit	Symbol	Bit Name	Description	R/W
b10	SSCAFBL	GTIOCnA Signal Falling Edge Applying Condition Select*1	b11 b10 0 0: Falling edge of the GTIOCnA signal is not used as a trigger to start counting. 0 1: The counter starts at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven low. 1 0: The counter starts at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven high. 1 1: The counter starts at a falling edge of the GTIOCnA signal.	R/W
b11	SSCAFBH			R/W
b12	SSCBRAL	GTIOCnB Signal Rising Edge Applying Condition Select*1	b13 b12 0 0: Rising edge of the GTIOCnB signal is not used as a trigger to start counting. 0 1: The counter starts at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven low. 1 0: The counter starts at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven high. 1 1: The counter starts at a rising edge of the GTIOCnB signal.	R/W
b13	SSCBRAH			R/W
b14	SSCBFAL	GTIOCnB Signal Falling Edge Applying Condition Select*1	b15 b14 0 0: Falling edge of the GTIOCnB signal is not used as a trigger to start counting. 0 1: The counter starts at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven low. 1 0: The counter starts at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven high. 1 1: The counter starts at a falling edge of the GTIOCnB signal.	R/W
b15	SSCBFAH			R/W
b16	SSELCA	ELCA Event Source Count Start Enable	0: Disables count start by the ELCA event input 1: Enables count start by the ELCA event input	R/W *2
b17	SSELCB	ELCB Event Source Count Start Enable	0: Disables count start by the ELCB event input 1: Enables count start by the ELCB event input	R/W *2
b18	SSELCC	ELCC Event Source Count Start Enable	0: Disables count start by the ELCC event input 1: Enables count start by the ELCC event input	R/W *2
b19	SSELCD	ELCD Event Source Count Start Enable	0: Disables count start by the ELCD event input 1: Enables count start by the ELCD event input	R/W *2
b20	SSELCE	ELCE Event Source Count Start Enable	0: Disables count start by the ELCE event input 1: Enables count start by the ELCE event input	R/W *2
b21	SSELCF	ELCF Event Source Count Start Enable	0: Disables count start by the ELCF event input 1: Enables count start by the ELCF event input	R/W *2
b22	SSELCG	ELCG Event Source Count Start Enable	0: Disables count start by the ELCG event input 1: Enables count start by the ELCG event input	R/W *2
b23	SSELCH	ELCH Event Source Count Start Enable	0: Disables count start by the ELCH event input 1: Enables count start by the ELCH event input	R/W *2
b30 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31	CSTRT	Software Source Count Start Enable	0: Disables count start by the GTSTR register 1: Enables count start by the GTSTR register	R/W *2

n = 0 to 7

Note 1. These bits are invalid in complementary PWM mode for GPTW0 to GPTW2 and GPTW4 to GPTW6.

Note 2. When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.

The GTSSR register sets a count start source for the GTCNT counter.

Inputs from the GTETRGA, GTETRGB, GTETRGC, and GTETRGD pins are input to the GPTW via the POEG. Set the polarity of these signals with the POEG.

24.2.6 General PWM Timer Stop Source Select Register (GTPSR)

Address(es): GPTW0.GTPSR 000C 2014h, GPTW1.GTPSR 000C 2114h, GPTW2.GTPSR 000C 2214h, GPTW3.GTPSR 000C 2314h, GPTW4.GTPSR 000C 2414h, GPTW5.GTPSR 000C 2514h, GPTW6.GTPSR 000C 2614h, GPTW7.GTPSR 000C 2714h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CSTOP	—	—	—	—	—	—	—	PSELCH	PSELCG	PSELCF	PSELCE	PSELCD	PSELCC	PSELCB	PSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PSCBFAH	PSCBFAL	PSCBRAH	PSCBRAL	PSCAFBH	PSCAFBL	PSCARBH	PSCARBL	PSGTRGDF	PSGTRGDR	PSGTRGCF	PSGTRGCR	PSGTRGBF	PSGTRGBR	PSGTRGAF	PSGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PSGTRGAR	GTETRGA Signal Edge Select	b1 b0 0 0: The GTETRGA signal is not used as a trigger to stop counting. 0 1: The counter stops at a rising edge of the GTETRGA signal. 1 0: The counter stops at a falling edge of the GTETRGA signal. 1 1: The counter stops at both edges of the GTETRGA signal.	R/W
b1	PSGTRGAF			R/W
b2	PSGTRGBR	GTETRGB Signal Edge Select	b3 b2 0 0: The GTETRGB signal is not used as a trigger to stop counting. 0 1: The counter stops at a rising edge of the GTETRGB signal. 1 0: The counter stops at a falling edge of the GTETRGB signal. 1 1: The counter stops at both edges of the GTETRGB signal.	R/W
b3	PSGTRGBF			R/W
b4	PSGTRGCR	GTETRGC Signal Edge Select	b5 b4 0 0: The GTETRGC signal is not used as a trigger to stop counting. 0 1: The counter stops at a rising edge of the GTETRGC signal. 1 0: The counter stops at a falling edge of the GTETRGC signal. 1 1: The counter stops at both edges of the GTETRGC signal.	R/W
b5	PSGTRGCF			R/W
b6	PSGTRGDR	GTETRGD Signal Edge Select	b7 b6 0 0: The GTETRGD signal is not used as a trigger to stop counting. 0 1: The counter stops at a rising edge of the GTETRGD signal. 1 0: The counter stops at a falling edge of the GTETRGD signal. 1 1: The counter stops at both edges of the GTETRGD signal.	R/W
b7	PSGTRGDF			R/W
b8	PSCARBL	GTIOCnA Signal Rising Edge Applying Condition Select*1	b9 b8 0 0: Rising edge of the GTIOCnA signal is not used as a trigger to stop counting. 0 1: The counter stops at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven low. 1 0: The counter stops at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven high. 1 1: The counter stops at a rising edge of the GTIOCnA signal.	R/W
b9	PSCARBH			R/W

Bit	Symbol	Bit Name	Description	R/W
b10	PSCAFBL	GTIOCnA Signal Falling Edge Applying Condition Select*1	b11 b10 0 0: Falling edge of the GTIOCnA signal is not used as a trigger to stop counting. 0 1: The counter stops at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven low. 1 0: The counter stops at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven high. 1 1: The counter stops at a falling edge of the GTIOCnA signal.	R/W
b11	PSCAFBH			R/W
b12	PSCBRAL	GTIOCnB Signal Rising Edge Applying Condition Select*1	b13 b12 0 0: Rising edge of the GTIOCnB signal is not used as a trigger to stop counting. 0 1: The counter stops at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven low. 1 0: The counter stops at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven high. 1 1: The counter stops at a rising edge of the GTIOCnB signal.	R/W
b13	PSCBRAH			R/W
b14	PSCBFAL	GTIOCnB Signal Falling Edge Applying Condition Select*1	b15 b14 0 0: Falling edge of the GTIOCnB signal is not used as a trigger to stop counting. 0 1: The counter stops at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven low. 1 0: The counter stops at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven high. 1 1: The counter stops at a falling edge of the GTIOCnB signal.	R/W
b15	PSCBFAH			R/W
b16	PSELCA	ELCA Event Source Count Stop Enable	0: Disables count stop by the ELCA event input 1: Enables count stop by the ELCA event input	R/W *2
b17	PSELCB	ELCB Event Source Count Stop Enable	0: Disables count stop by the ELCB event input 1: Enables count stop by the ELCB event input	R/W *2
b18	PSELCC	ELCC Event Source Count Stop Enable	0: Disables count stop by the ELCC event input 1: Enables count stop by the ELCC event input	R/W *2
b19	PSELCD	ELCD Event Source Count Stop Enable	0: Disables count stop by the ELCD event input 1: Enables count stop by the ELCD event input	R/W *2
b20	PSELCE	ELCE Event Source Count Stop Enable	0: Disables count stop by the ELCE event input 1: Enables count stop by the ELCE event input	R/W *2
b21	PSELCF	ELCF Event Source Count Stop Enable	0: Disables count stop by the ELCF event input 1: Enables count stop by the ELCF event input	R/W *2
b22	PSELCG	ELCG Event Source Count Stop Enable	0: Disables count stop by the ELCG event input 1: Enables count stop by the ELCG event input	R/W *2
b23	PSELCH	ELCH Event Source Count Stop Enable	0: Disables count stop by the ELCH event input 1: Enables count stop by the ELCH event input	R/W *2
b30 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31	CSTOP	Software Source Count Stop Enable	0: Disables count stop by the GTSTP register 1: Enables count stop by the GTSTP register	R/W *2

n = 0 to 7

Note 1. These bits are invalid in complementary PWM mode for GPTW0 to GPTW2 and GPTW4 to GPTW6.

Note 2. When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.

The GTPSR register sets a count stop source for the GTCNT counter.

Inputs from the GTETRGA, GTETRGB, GTETRGC, and GTETRGD pins are input to the GPTW via the POEG. Set the polarity of these signals with the POEG.

24.2.7 General PWM Timer Clear Source Select Register (GTCSR)

Address(es): GPTW0.GTCSR 000C 2018h, GPTW1.GTCSR 000C 2118h, GPTW2.GTCSR 000C 2218h, GPTW3.GTCSR 000C 2318h, GPTW4.GTCSR 000C 2418h, GPTW5.GTCSR 000C 2518h, GPTW6.GTCSR 000C 2618h, GPTW7.GTCSR 000C 2718h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CCLR	—	—	—	CP1CCE	CSCMSC[2:0]			CSELCH	CSELCG	CSELCF	CSELCE	CSELCD	CSELCC	CSELCB	CSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CSCBFAH	CSCBFAL	CSCBRAH	CSCBRAL	CSCAFBH	CSCAFBL	CSCARBH	CSCARBL	CSGTRGDF	CSGTRGDR	CSGTRGCF	CSGTRGCR	CSGTRGBF	CSGTRGBR	CSGTRGAF	CSGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CSGTRGAR	GTETRGA Signal Edge Select	b1 b0 0 0: The GTETRGA signal is not used as a trigger to clear the counter. 0 1: The counter is cleared at a rising edge of the GTETRGA signal. 1 0: The counter is cleared at a falling edge of the GTETRGA signal. 1 1: The counter is cleared at both edges of the GTETRGA signal.	R/W *1
b1	CSGTRGAF			R/W *1
b2	CSGTRGBR	GTETRGB Signal Edge Select	b3 b2 0 0: The GTETRGB signal is not used as a trigger to clear the counter. 0 1: The counter is cleared at a rising edge of the GTETRGB signal. 1 0: The counter is cleared at a falling edge of the GTETRGB signal. 1 1: The counter is cleared at both edges of the GTETRGB signal.	R/W *1
b3	CSGTRGBF			R/W *1
b4	CSGTRGCR	GTETRGC Signal Edge Select	b5 b4 0 0: The GTETRGC signal is not used as a trigger to clear the counter. 0 1: The counter is cleared at a rising edge of the GTETRGC signal. 1 0: The counter is cleared at a falling edge of the GTETRGC signal. 1 1: The counter is cleared at both edges of the GTETRGC signal.	R/W *1
b5	CSGTRGCF			R/W *1
b6	CSGTRGDR	GTETRGD Signal Edge Select	b7 b6 0 0: The GTETRGD signal is not used as a trigger to clear the counter. 0 1: The counter is cleared at a rising edge of the GTETRGD signal. 1 0: The counter is cleared at a falling edge of the GTETRGD signal. 1 1: The counter is cleared at both edges of the GTETRGD signal.	R/W *1
b7	CSGTRGDF			R/W *1
b8	CSCARBL	GTIOCnA Signal Rising Edge Applying Condition Select*2	b9 b8 0 0: Rising edge of the GTIOCnA signal is not used as a trigger to clear the counter. 0 1: The counter is cleared at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven low. 1 0: The counter is cleared at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven high. 1 1: The counter is cleared at a rising edge of the GTIOCnA signal.	R/W
b9	CSCARBH			R/W

Bit	Symbol	Bit Name	Description	R/W
b10	CSCAFBL	GTIOCnA Signal Falling Edge Applying Condition Select*2	b11 b10 0 0: Falling edge of the GTIOCnA signal is not used as a trigger to clear the counter. 0 1: The counter is cleared at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven low. 1 0: The counter is cleared at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven high. 1 1: The counter is cleared at a falling edge of the GTIOCnA signal.	R/W
b11	CSCAFBH			R/W
b12	CSCBRAL	GTIOCnB Signal Rising Edge Applying Condition Select*2	b13 b12 0 0: Rising edge of the GTIOCnB signal is not used as a trigger to clear the counter. 0 1: The counter is cleared at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven low. 1 0: The counter is cleared at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven high. 1 1: The counter is cleared at a rising edge of the GTIOCnB signal.	R/W
b13	CSCBRAH			R/W
b14	CSCBFAL	GTIOCnB Signal Falling Edge Applying Condition Select*2	b15 b14 0 0: Falling edge of the GTIOCnB signal is not used as a trigger to clear the counter. 0 1: The counter is cleared at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven low. 1 0: The counter is cleared at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven high. 1 1: The counter is cleared at a falling edge of the GTIOCnB signal.	R/W
b15	CSCBFAH			R/W
b16	CSELCA	ELCA Event Source Counter Clear Enable	0: Disables counter clear by the ELCA event input 1: Enables counter clear by the ELCA event input	R/W *1
b17	CSELCB	ELCB Event Source Counter Clear Enable	0: Disables counter clear by the ELCB event input 1: Enables counter clear by the ELCB event input	R/W *1
b18	CSELCC	ELCC Event Source Counter Clear Enable	0: Disables counter clear by the ELCC event input 1: Enables counter clear by the ELCC event input	R/W *1
b19	CSELCD	ELCD Event Source Counter Clear Enable	0: Disables counter clear by the ELCD event input 1: Enables counter clear by the ELCD event input	R/W *1
b20	CSELCE	ELCE Event Source Counter Clear Enable	0: Disables counter clear by the ELCE event input 1: Enables counter clear by the ELCE event input	R/W *1
b21	CSELCF	ELCF Event Source Counter Clear Enable	0: Disables counter clear by the ELCF event input 1: Enables counter clear by the ELCF event input	R/W *1
b22	CSELCG	ELCG Event Source Counter Clear Enable	0: Disables counter clear by the ELCG event input 1: Enables counter clear by the ELCG event input	R/W *1
b23	CSELCH	ELCH Event Source Counter Clear Enable	0: Disables counter clear by the ELCH event input 1: Enables counter clear by the ELCH event input	R/W *1

Bit	Symbol	Bit Name	Description	R/W
b26 to b24	CSCMSC[2:0]	Compare Match/Input Capture/ Synchronous Counter Clearing Source Counter Clear Enable	b26 b24 0 0 0: Counter clear disabled by Compare match/Input capture/Synchronous counter clearing group. 0 0 1: Counter clear enabled at the GTCCRA register compare match/Input capture. 0 1 0: Counter clear enabled at the GTCCRB register compare match/Input capture. 0 1 1: Counter clear enabled at the GTCCRC register compare match. 1 0 0: Counter clear enabled at the GTCCRD register compare match. 1 0 1: Counter clear enabled at the GTCCRE register compare match. 1 1 0: Counter clear enabled at the GTCCRF register compare match. 1 1 1: Counter clear enabled at the synchronous counter clearing group.	R/W
b27	CP1CCE	Complementary PWM Mode 1 Crest Source Counter Clear Enable	0: Counter clear disabled at the crest of complementary PWM mode 1 1: Counter clear enabled at the crest of complementary PWM mode 1	R/W *1
b30 to b28	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31	CCLR	Software Source Counter Clear Enable	0: Disables counter clear by the GTCLR register 1: Enables counter clear by the GTCLR register	R/W *1

n = 0 to 7

Note 1. When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.

Note 2. These bits are invalid in complementary PWM mode for GPTW0 to GPTW2 and GPTW4 to GPTW6.

The GTCSR register sets a counter clear source for the GTCNT counter.

Counter clearing can be executed whether the counter is running (the GTCR.CST bit = 1) or stopped (the GTCR.CST bit = 0).

Inputs from the GTETRGA, GTETRGB, GTETRGC, and GTETRGD pins are input to the GPTW via the POEG. Set the polarity of these signals with the POEG.

CSCMSC[2:0] Bits (Compare Match/Input Capture/Synchronous Counter Clearing Source Counter Clear Enable)

Select whether to enable or disable for the counter clear of the GTCNT counter by compare match/input capture/ synchronous counter clearing group. When counter clearing in response to a match in comparison or input capture is enabled, either can be used as a source for synchronous clearing by inter channel cooperation as described in section 24.3.8.3, Synchronous Clear Operation by Inter Channel Cooperation. When counter clearing by input capture is enabled while the setting of the CSCMSC[2:0] bits is 001b or 010b, the same sources for input capture as one set in the GTICmSR (m = A, B) register must be set as the counter clearing sources in the GTCSR register. Note that input capture on other channels cannot be used as the source to drive counter clearing, that is, the GTICASR.ASOC or GTICBSR.BSOC bit cannot be set to 1.

Since the compare match by the register that is performing the buffer operation (including the wave mode specific case) does not occur, the counter clear enable setting that makes the target register of the buffer operation the compare match factor is invalid.

In complementary PWM mode, the counter clear enable setting for compare match of the GTCCRB register, GTCCRE register, and GTCCRF register is invalid even when the buffer operation is not performed.

CP1CCE Bit (Complementary PWM Mode 1 Crest Source Counter Clear Enable)

Select enable or disable for the counter clear at the crest of complementary PWM mode 1.

To enable this bit, do not set 1 to the PSYE bit of the GTIOR register.

It is valid only for the master channel in complementary PWM mode. The master channel setting also clears the GTCNT counter of the slave channel in complementary PWM mode.

24.2.8 General PWM Timer Count-Up Source Select Register (GTUPSR)

Address(es): GPTW0.GTUPSR 000C 201Ch, GPTW1.GTUPSR 000C 211Ch, GPTW2.GTUPSR 000C 221Ch, GPTW3.GTUPSR 000C 231Ch, GPTW4.GTUPSR 000C 241Ch, GPTW5.GTUPSR 000C 251Ch, GPTW6.GTUPSR 000C 261Ch, GPTW7.GTUPSR 000C 271Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	USILVL[3:0]				USELCH	USELCG	USELCF	USELCE	USELCD	USELCC	USELCB	USELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	USCBFAH	USCBFAL	USCBRAH	USCBRAL	USCAFBH	USCAFBL	USCARBH	USCARBL	USGTRGDF	USGTRGDR	USGTRGCF	USGTRGCR	USGTRGBF	USGTRGBR	USGTRGAF	USGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	USGTRGAR	GTETRGA Signal Edge Select	b1 b0 0 0: The GTETRGA signal is not used as a trigger to increment the counter. 0 1: The counter is incremented at a rising edge of the GTETRGA signal. 1 0: The counter is incremented at a falling edge of the GTETRGA signal. 1 1: The counter is incremented at both edges of the GTETRGA signal.	R/W
b1	USGTRGAF			R/W
b2	USGTRGBR	GTETRGB Signal Edge Select	b3 b2 0 0: The GTETRGB signal is not used as a trigger to increment the counter. 0 1: The counter is incremented at a rising edge of the GTETRGB signal. 1 0: The counter is incremented at a falling edge of the GTETRGB signal. 1 1: The counter is incremented at both edges of the GTETRGB signal.	R/W
b3	USGTRGBF			R/W
b4	USGTRGCR	GTETRGC Signal Edge Select	b5 b4 0 0: The GTETRGC signal is not used as a trigger to increment the counter. 0 1: The counter is incremented at a rising edge of the GTETRGC signal. 1 0: The counter is incremented at a falling edge of the GTETRGC signal. 1 1: The counter is incremented at both edges of the GTETRGC signal.	R/W
b5	USGTRGCF			R/W
b6	USGTRGDR	GTETRGD Signal Edge Select	b7 b6 0 0: The GTETRGD signal is not used as a trigger to increment the counter. 0 1: The counter is incremented at a rising edge of the GTETRGD signal. 1 0: The counter is incremented at a falling edge of the GTETRGD signal. 1 1: The counter is incremented at both edges of the GTETRGD signal.	R/W
b7	USGTRGDF			R/W
b8	USCARBL	GTIOCnA Signal Rising Edge Applying Condition Select	b9 b8 0 0: Rising edge of the GTIOCnA signal is not used as a trigger to increment the counter. 0 1: The counter is incremented at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven low. 1 0: The counter is incremented at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven high. 1 1: The counter is incremented at a rising edge of the GTIOCnA signal.	R/W
b9	USCARBH			R/W

Bit	Symbol	Bit Name	Description	R/W
b10	USCAFBL	GTIOCnA Signal Falling Edge Applying Condition Select	b11 b10 0 0: Falling edge of the GTIOCnA signal is not used as a trigger to increment the counter. 0 1: The counter is incremented at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven low. 1 0: The counter is incremented at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven high. 1 1: The counter is incremented at a falling edge of the GTIOCnA signal.	R/W
b11	USCAFBH			R/W
b12	USCBRAL	GTIOCnB Signal Rising Edge Applying Condition Select	b13 b12 0 0: Rising edge of the GTIOCnB signal is not used as a trigger to increment the counter. 0 1: The counter is incremented at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven low. 1 0: The counter is incremented at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven high. 1 1: The counter is incremented at a rising edge of the GTIOCnB signal.	R/W
b13	USCBRAH			R/W
b14	USCBFAL	GTIOCnB Signal Falling Edge Applying Condition Select	b15 b14 0 0: Falling edge of the GTIOCnB signal is not used as a trigger to increment the counter. 0 1: The counter is incremented at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven low. 1 0: The counter is incremented at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven high. 1 1: The counter is incremented at a falling edge of the GTIOCnB signal.	R/W
b15	USCBFAH			R/W
b16	USELCA	ELCA Event Source Count-Up Enable	0: Disables count-up by the ELCA event input 1: Enables count-up by the ELCA event input	R/W
b17	USELCB	ELCB Event Source Count-Up Enable	0: Disables count-up by the ELCB event input 1: Enables count-up by the ELCB event input	R/W
b18	USELCC	ELCC Event Source Count-Up Enable	0: Disables count-up by the ELCC event input 1: Enables count-up by the ELCC event input	R/W
b19	USELCD	ELCD Event Source Count-Up Enable	0: Disables count-up by the ELCD event input 1: Enables count-up by the ELCD event input	R/W
b20	USELCE	ELCE Event Source Count-Up Enable	0: Disables count-up by the ELCE event input 1: Enables count-up by the ELCE event input	R/W
b21	USELCF	ELCF Event Source Count-Up Enable	0: Disables count-up by the ELCF event input 1: Enables count-up by the ELCF event input	R/W
b22	USELCG	ELCG Event Source Count-Up Enable	0: Disables count-up by the ELCG event input 1: Enables count-up by the ELCG event input	R/W
b23	USELCH	ELCH Event Source Count-Up Enable	0: Disables count-up by the ELCH event input 1: Enables count-up by the ELCH event input	R/W

Bit	Symbol	Bit Name	Description	R/W
b27to b24	USILVL[3:0]	External Input Level Source Count-Up Enable	b27 b24 0 0 0 0: Disable count-up by external input. 0 0 0 1: Setting prohibited. 0 0 1 0: Enable count-up by GTIOCnA pin input level Low. 0 0 1 1: Enable count-up by GTIOCnA pin input level High. 0 1 0 0: Enable count-up by GTIOCnB pin input level Low. 0 1 0 1: Enable count-up by GTIOCnB pin input level High. 0 1 1 0: Setting prohibited. 0 1 1 1: Setting prohibited. 1 0 0 0: Enable count-up by GTETRGA pin input level Low. 1 0 0 1: Enable count-up by GTETRGA pin input level High. 1 0 1 0: Enable count-up by GTETRGB pin input level Low. 1 0 1 1: Enable count-up by GTETRGB pin input level High. 1 1 0 0: Enable count-up by GTETRGC pin input level Low. 1 1 0 1: Enable count-up by GTETRGC pin input level High. 1 1 1 0: Enable count-up by GTETRGD pin input level Low. 1 1 1 1: Enable count-up by GTETRGD pin input level High.	R/W
b31 to b28	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

n = 0 to 7

The GTUPSR register sets a count-up source for the GTCNT counter.
When at least one bit among bits in the GTUPSR register is set as 1, counting of the GTCNT counter by the count clock set by the GTCR.TPCS[3:0] bits becomes invalid, and the count-up by a source set as 1 by the GTUPSR register is executed.
Number of increment in counting is one even when multiple sources are generated simultaneously.
Inputs from the GTETRGA, GTETRGB, GTETRGC, and GTETRGD pins are input to the GPTW via the POEG. Set the polarity of these signals with the POEG.

24.2.9 General PWM Timer Count-Down Source Select Register (GTDNSR)

Address(es): GPTW0.GTDNSR 000C 2020h, GPTW1.GTDNSR 000C 2120h, GPTW2.GTDNSR 000C 2220h, GPTW3.GTDNSR 000C 2320h, GPTW4.GTDNSR 000C 2420h, GPTW5.GTDNSR 000C 2520h, GPTW6.GTDNSR 000C 2620h, GPTW7.GTDNSR 000C 2720h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	DSILVL[3:0]				DSELCH	DSELCG	DSELCF	DSELCE	DSELCD	DSELCC	DSELCB	DSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DSCBFAH	DSCBFAL	DSCBRAH	DSCBRAL	DSCAFBH	DSCAFBL	DSCARBH	DSCARBL	DSGTRGDF	DSGTRGDR	DSGTRGCF	DSGTRGCR	DSGTRGBF	DSGTRGBR	DSGTRGAF	DSGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DSGTRGAR	GTETRGA Signal Edge Select	b1 b0 0 0: The GTETRGA signal is not used as a trigger to decrement the counter. 0 1: The counter is decremented at a rising edge of the GTETRGA signal. 1 0: The counter is decremented at a falling edge of the GTETRGA signal. 1 1: The counter is decremented at both edges of the GTETRGA signal.	R/W
b1	DSGTRGAF			R/W
b2	DSGTRGBR	GTETRGB Signal Edge Select	b3 b2 0 0: The GTETRGB signal is not used as a trigger to decrement the counter. 0 1: The counter is decremented at a rising edge of the GTETRGB signal. 1 0: The counter is decremented at a falling edge of the GTETRGB signal. 1 1: The counter is decremented at both edges of the GTETRGB signal.	R/W
b3	DSGTRGBF			R/W
b4	DSGTRGCR	GTETRGC Signal Edge Select	b5 b4 0 0: The GTETRGC signal is not used as a trigger to decrement the counter. 0 1: The counter is decremented at a rising edge of the GTETRGC signal. 1 0: The counter is decremented at a falling edge of the GTETRGC signal. 1 1: The counter is decremented at both edges of the GTETRGC signal.	R/W
b5	DSGTRGCF			R/W
b6	DSGTRGDR	GTETRGD Signal Edge Select	b7 b6 0 0: The GTETRGD signal is not used as a trigger to decrement the counter. 0 1: The counter is decremented at a rising edge of the GTETRGD signal. 1 0: The counter is decremented at a falling edge of the GTETRGD signal. 1 1: The counter is decremented at both edges of the GTETRGD signal.	R/W
b7	DSGTRGDF			R/W
b8	DSCARBL	GTIOCnA Signal Rising Edge Applying Condition Select	b9 b8 0 0: Rising edge of the GTIOCnA signal is not used as a trigger to decrement the counter. 0 1: The counter is decremented at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven low. 1 0: The counter is decremented at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven high. 1 1: The counter is decremented at a rising edge of the GTIOCnA signal.	R/W
b9	DSCARBH			R/W

Bit	Symbol	Bit Name	Description	R/W
b10	DSCAFBL	GTIOCnA Signal Falling Edge Applying Condition Select	b11 b10 0 0: Falling edge of the GTIOCnA signal is not used as a trigger to decrement the counter. 0 1: The counter is decremented at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven low. 1 0: The counter is decremented at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven high. 1 1: The counter is decremented at a falling edge of the GTIOCnA signal.	R/W
b11	DSCAFBH			R/W
b12	DSCBRAL	GTIOCnB Signal Rising Edge Applying Condition Select	b13 b12 0 0: Rising edge of the GTIOCnB signal is not used as a trigger to decrement the counter. 0 1: The counter is decremented at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven low. 1 0: The counter is decremented at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven high. 1 1: The counter is decremented at a rising edge of the GTIOCnB signal.	R/W
b13	DSCBRAH			R/W
b14	DSCBFAL	GTIOCnB Signal Falling Edge Applying Condition Select	b15 b14 0 0: Falling edge of the GTIOCnB signal is not used as a trigger to decrement the counter. 0 1: The counter is decremented at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven low. 1 0: The counter is decremented at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven high. 1 1: The counter is decremented at a falling edge of the GTIOCnB signal.	R/W
b15	DSCBFAH			R/W
b16	DSELCA	ELCA Event Source Count-Down Enable	0: Disables count-down by the ELCA event input 1: Enables count-down by the ELCA event input	R/W
b17	DSELCB	ELCB Event Source Count-Down Enable	0: Disables count-down by the ELCB event input 1: Enables count-down by the ELCB event input	R/W
b18	DSELCC	ELCC Event Source Count-Down Enable	0: Disables count-down by the ELCC event input 1: Enables count-down by the ELCC event input	R/W
b19	DSELCD	ELCD Event Source Count-Down Enable	0: Disables count-down by the ELCD event input 1: Enables count-down by the ELCD event input	R/W
b20	DSELCE	ELCE Event Source Count-Down Enable	0: Disables count-down by the ELCE event input 1: Enables count-down by the ELCE event input	R/W
b21	DSELCF	ELCF Event Source Count-Down Enable	0: Disables count-down by the ELCF event input 1: Enables count-down by the ELCF event input	R/W
b22	DSELCG	ELCG Event Source Count-Down Enable	0: Disables count-down by the ELCG event input 1: Enables count-down by the ELCG event input	R/W
b23	DSELCH	ELCH Event Source Count-Down Enable	0: Disables count-down by the ELCH event input 1: Enables count-down by the ELCH event input	R/W

Bit	Symbol	Bit Name	Description	R/W
b27 to b24	DSILVL[3:0]	External Input Level Source Count-Down Enable	b27 b24 0 0 0 0: Disable count-down by external input. 0 0 0 1: Setting prohibited. 0 0 1 0: Enable count-down by GTIOCnA pin input level Low. 0 0 1 1: Enable count-down by GTIOCnA pin input level High. 0 1 0 0: Enable count-down by GTIOCnB pin input level Low. 0 1 0 1: Enable count-down by GTIOCnB pin input level High. 0 1 1 0: Setting prohibited. 0 1 1 1: Setting prohibited. 1 0 0 0: Enable count-down by GTETRGA pin input level Low. 1 0 0 1: Enable count-down by GTETRGA pin input level High. 1 0 1 0: Enable count-down by GTETRGB pin input level Low. 1 0 1 1: Enable count-down by GTETRGB pin input level High. 1 1 0 0: Enable count-down by GTETRGC pin input level Low. 1 1 0 1: Enable count-down by GTETRGC pin input level High. 1 1 1 0: Enable count-down by GTETRGD pin input level Low. 1 1 1 1: Enable count-down by GTETRGD pin input level High.	R/W
b31 to b28	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

n = 0 to 7

The GTDNSR register sets a count-down source for the GTCNT counter.
When at least one bit among bits in the GTDNSR register is set as 1, counting of the GTCNT counter by the count clock set by the GTCR.TPCS[3:0] bits becomes invalid, and the count-down by a source set as 1 by the GTDNSR register is executed.
Number of decrement in counting is one even when multiple sources are generated simultaneously.
Inputs from the GTETRGA, GTETRGB, GTETRGC, and GTETRGD pins are input to the GPTW via the POEG. Set the polarity of these signals with the POEG.

24.2.10 General PWM Timer Input Capture Source Select Register A (GTICASR)

Address(es): GPTW0.GTICASR 000C 2024h, GPTW1.GTICASR 000C 2124h, GPTW2.GTICASR 000C 2224h, GPTW3.GTICASR 000C 2324h, GPTW4.GTICASR 000C 2424h, GPTW5.GTICASR 000C 2524h, GPTW6.GTICASR 000C 2624h, GPTW7.GTICASR 000C 2724h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	ASOC	ASELCH	ASELCG	ASELCF	ASELCE	ASELCD	ASELCC	ASELCB	ASELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ASCBFAH	ASCBFAL	ASCBRAH	ASCBRAL	ASCAFBH	ASCAFBL	ASCARBH	ASCARBL	ASGTRGDF	ASGTRGDR	ASGTRGCF	ASGTRGCR	ASGTRGBF	ASGTRGBR	ASGTRGAF	ASGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ASGTRGAR	GTETRGA Signal Edge Select	b1 b0 0 0: The GTETRGA signal is not used as a trigger to capture the counter value in the GTCCRA register. 0 1: The counter value is captured in the GTCCRA register at a rising edge of the GTETRGA signal. 1 0: The counter value is captured in the GTCCRA register at a falling edge of the GTETRGA signal. 1 1: The counter value is captured in the GTCCRA register at both edges of the GTETRGA signal.	R/W
b1	ASGTRGAF			R/W
b2	ASGTRGBR	GTETRGB Signal Edge Select	b3 b2 0 0: The GTETRGB signal is not used as a trigger to capture the counter value in the GTCCRA register. 0 1: The counter value is captured in the GTCCRA register at a rising edge of the GTETRGB signal. 1 0: The counter value is captured in the GTCCRA register at a falling edge of the GTETRGB signal. 1 1: The counter value is captured in the GTCCRA register at both edges of the GTETRGB signal.	R/W
b3	ASGTRGBF			R/W
b4	ASGTRGCR	GTETRGC Signal Edge Select	b5 b4 0 0: The GTETRGC signal is not used as a trigger to capture the counter value in the GTCCRA register. 0 1: The counter value is captured in the GTCCRA register at a rising edge of the GTETRGC signal. 1 0: The counter value is captured in the GTCCRA register at a falling edge of the GTETRGC signal. 1 1: The counter value is captured in the GTCCRA register at both edges of the GTETRGC signal.	R/W
b5	ASGTRGCF			R/W
b6	ASGTRGDR	GTETRGD Signal Edge Select	b7 b6 0 0: The GTETRGD signal is not used as a trigger to capture the counter value in the GTCCRA register. 0 1: The counter value is captured in the GTCCRA register at a rising edge of the GTETRGD signal. 1 0: The counter value is captured in the GTCCRA register at a falling edge of the GTETRGD signal. 1 1: The counter value is captured in the GTCCRA register at both edges of the GTETRGD signal.	R/W
b7	ASGTRGDF			R/W
b8	ASCARBL	GTIOCnA Signal Rising Edge Applying Condition Select	b9 b8 0 0: Rising edge of the GTIOCnA signal is not used as a trigger to capture the counter value in the GTCCRA register. 0 1: The counter value is captured in the GTCCRA register at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven low. 1 0: The counter value is captured in the GTCCRA register at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven high. 1 1: The counter value is captured in the GTCCRA register at a rising edge of the GTIOCnA signal.	R/W
b9	ASCARBH			R/W

Bit	Symbol	Bit Name	Description	R/W
b10	ASCAFBL	GTIOCnA Signal Falling Edge Applying Condition Select	b11 b10 0 0: Falling edge of the GTIOCnA signal is not used as a trigger to capture the counter value in the GTCCRA register. 0 1: The counter value is captured in the GTCCRA register at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven low. 1 0: The counter value is captured in the GTCCRA register at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven high. 1 1: The counter value is captured in the GTCCRA register at a falling edge of the GTIOCnA signal.	R/W
b11	ASCAFBH			R/W
b12	ASCBRAL	GTIOCnB Signal Rising Edge Applying Condition Select	b13 b12 0 0: Rising edge of the GTIOCnB signal is not used as a trigger to capture the counter value in the GTCCRA register. 0 1: The counter value is captured in the GTCCRA register at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven low. 1 0: The counter value is captured in the GTCCRA register at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven high. 1 1: The counter value is captured in the GTCCRA register at a rising edge of the GTIOCnB signal.	R/W
b13	ASCBRAH			R/W
b14	ASCBFAL	GTIOCnB Signal Falling Edge Applying Condition Select	b15 b14 0 0: Falling edge of the GTIOCnB signal is not used as a trigger to capture the counter value in the GTCCRA register. 0 1: The counter value is captured in the GTCCRA register at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven low. 1 0: The counter value is captured in the GTCCRA register at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven high. 1 1: The counter value is captured in the GTCCRA register at a falling edge of the GTIOCnB signal.	R/W
b15	ASCBFAH			R/W
b16	ASELCA	ELCA Event Source GTCCRA Input Capture Enable	0: Disables GTCCRA input capture by the ELCA event input 1: Enables GTCCRA input capture by the ELCA event input	R/W
b17	ASELCB	ELCB Event Source GTCCRA Input Capture Enable	0: Disables GTCCRA input capture by the ELCB event input 1: Enables GTCCRA input capture by the ELCB event input	R/W
b18	ASELCC	ELCC Event Source GTCCRA Input Capture Enable	0: Disables GTCCRA input capture by the ELCC event input 1: Enables GTCCRA input capture by the ELCC event input	R/W
b19	ASELCD	ELCD Event Source GTCCRA Input Capture Enable	0: Disables GTCCRA input capture by the ELCD event input 1: Enables GTCCRA input capture by the ELCD event input	R/W
b20	ASELCE	ELCE Event Source GTCCRA Input Capture Enable	0: Disables GTCCRA input capture by the ELCE event input 1: Enables GTCCRA input capture by the ELCE event input	R/W
b21	ASELCF	ELCF Event Source GTCCRA Input Capture Enable	0: Disables GTCCRA input capture by the ELCF event input 1: Enables GTCCRA input capture by the ELCF event input	R/W
b22	ASELCG	ELCG Event Source GTCCRA Input Capture Enable	0: Disables GTCCRA input capture by the ELCG event input 1: Enables GTCCRA input capture by the ELCG event input	R/W

Bit	Symbol	Bit Name	Description	R/W
b23	ASELCH	ELCH Event Source GTCCRA Input Capture Enable	0: Disables GTCCRA input capture by the ELCH event input 1: Enables GTCCRA input capture by the ELCH event input	R/W
b24	ASOC	Other Channel Source GTCCRA Input Capture Enable	0: Disables GTCCRA input capture by the other channel factor 1: Enables GTCCRA input capture by the other channel factor	R/W
b31 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

n = 0 to 7

The GTICASR register sets an input capture source for the GTCCRA register.
When at least one bit among bits in the GTICASR register is set to 1, input capture operation making the GTCCRA register as an input capture register is performed.
Inputs from the GTETRGA, GTETRGB, GTETRGC, and GTETRGD pins are input to the GPTW via the POEG. Set the polarity of these signals with the POEG.

ASOC Bit (Other Channel Source GTCCRA Input Capture Enable)

Select enable or disable for an input capture to the GTCCRA register by other channel factor.
Input capture of other channel factors is not subject to input capture factors to other channels set by the GTICCR.ICAFA, ICBFA bits.

24.2.11 General PWM Timer Input Capture Source Select Register B (GTICBSR)

Address(es): GPTW0.GTICBSR 000C 2028h, GPTW1.GTICBSR 000C 2128h, GPTW2.GTICBSR 000C 2228h, GPTW3.GTICBSR 000C 2328h, GPTW4.GTICBSR 000C 2428h, GPTW5.GTICBSR 000C 2528h, GPTW6.GTICBSR 000C 2628h, GPTW7.GTICBSR 000C 2728h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	BSOC	BSELCH	BSELCG	BSELCF	BSELCE	BSELCD	BSELCC	BSELCB	BSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BSCBFAH	BSCBFAL	BSCBRAH	BSCBRAL	BSCAFBH	BSCAFBL	BSCARBH	BSCARBL	BSGTRGDF	BSGTRGDR	BSGTRGCF	BSGTRGCR	BSGTRGBF	BSGTRGBR	BSGTRGAF	BSGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BSGTRGAR	GTETRGA Signal Edge Select	b1 b0 0 0: The GTETRGA signal is not used as a trigger to capture the counter value in the GTCCRB register. 0 1: The counter value is captured in the GTCCRB register at a rising edge of the GTETRGA signal. 1 0: The counter value is captured in the GTCCRB register at a falling edge of the GTETRGA signal. 1 1: The counter value is captured in the GTCCRB register at both edges of the GTETRGA signal.	R/W *1
b1	BSGTRGAF			R/W *1
b2	BSGTRGBR	GTETRGB Signal Edge Select	b3 b2 0 0: The GTETRGB signal is not used as a trigger to capture the counter value in the GTCCRB register. 0 1: The counter value is captured in the GTCCRB register at a rising edge of the GTETRGB signal. 1 0: The counter value is captured in the GTCCRB register at a falling edge of the GTETRGB signal. 1 1: The counter value is captured in the GTCCRB register at both edges of the GTETRGB signal.	R/W *1
b3	BSGTRGBF			R/W *1
b4	BSGTRGCR	GTETRGC Signal Edge Select	b5 b4 0 0: The GTETRGC signal is not used as a trigger to capture the counter value in the GTCCRB register. 0 1: The counter value is captured in the GTCCRB register at a rising edge of the GTETRGC signal. 1 0: The counter value is captured in the GTCCRB register at a falling edge of the GTETRGC signal. 1 1: The counter value is captured in the GTCCRB register at both edges of the GTETRGC signal.	R/W *1
b5	BSGTRGCF			R/W *1
b6	BSGTRGDR	GTETRGD Signal Edge Select	b7 b6 0 0: The GTETRGD signal is not used as a trigger to capture the counter value in the GTCCRB register. 0 1: The counter value is captured in the GTCCRB register at a rising edge of the GTETRGD signal. 1 0: The counter value is captured in the GTCCRB register at a falling edge of the GTETRGD signal. 1 1: The counter value is captured in the GTCCRB register at both edges of the GTETRGD signal.	R/W *1
b7	BSGTRGDF			R/W *1
b8	BSCARBL	GTIOCnA Signal Rising Edge Applying Condition Select	b9 b8 0 0: Rising edge of the GTIOCnA signal is not used as a trigger to capture the counter value in the GTCCRB register. 0 1: The counter value is captured in the GTCCRB register at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven low. 1 0: The counter value is captured in the GTCCRB register at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven high. 1 1: The counter value is captured in the GTCCRB register at a rising edge of the GTIOCnA signal.	R/W
b9	BSCARBH			R/W

Bit	Symbol	Bit Name	Description	R/W
b10	BSCAFBL	GTIOCnA Signal Falling Edge Applying Condition Select	b11 b10 0 0: Falling edge of the GTIOCnA signal is not used as a trigger to capture the counter value in the GTCCRB register. 0 1: The counter value is captured in the GTCCRB register at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven low. 1 0: The counter value is captured in the GTCCRB register at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven high. 1 1: The counter value is captured in the GTCCRB register at a falling edge of the GTIOCnA signal.	R/W
b11	BSCAFBH			R/W
b12	BSCBRAL	GTIOCnB Signal Rising Edge Applying Condition Select	b13 b12 0 0: Rising edge of the GTIOCnB signal is not used as a trigger to capture the counter value in the GTCCRB register. 0 1: The counter value is captured in the GTCCRB register at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven low. 1 0: The counter value is captured in the GTCCRB register at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven high. 1 1: The counter value is captured in the GTCCRB register at a rising edge of the GTIOCnB signal.	R/W
b13	BSCBRAH			R/W
b14	BSCBFAL	GTIOCnB Signal Falling Edge Applying Condition Select	b15 b14 0 0: Falling edge of the GTIOCnB signal is not used as a trigger to capture the counter value in the GTCCRB register. 0 1: The counter value is captured in the GTCCRB register at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven low. 1 0: The counter value is captured in the GTCCRB register at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven high. 1 1: The counter value is captured in the GTCCRB register at a falling edge of the GTIOCnB signal.	R/W
b15	BSCBFAH			R/W
b16	BSELCA	ELCA Event Source GTCCRB Input Capture Enable	0: Disables GTCCRB input capture by the ELCA event input 1: Enables GTCCRB input capture by the ELCA event input	R/W
b17	BSELCB	ELCB Event Source GTCCRB Input Capture Enable	0: Disables GTCCRB input capture by the ELCB event input 1: Enables GTCCRB input capture by the ELCB event input	R/W
b18	BSELCC	ELCC Event Source GTCCRB Input Capture Enable	0: Disables GTCCRB input capture by the ELCC event input 1: Enables GTCCRB input capture by the ELCC event input	R/W
b19	BSELCD	ELCD Event Source GTCCRB Input Capture Enable	0: Disables GTCCRB input capture by the ELCD event input 1: Enables GTCCRB input capture by the ELCD event input	R/W
b20	BSELCE	ELCE Event Source GTCCRB Input Capture Enable	0: Disables GTCCRB input capture by the ELCE event input 1: Enables GTCCRB input capture by the ELCE event input	R/W
b21	BSELCF	ELCF Event Source GTCCRB Input Capture Enable	0: Disables GTCCRB input capture by the ELCF event input 1: Enables GTCCRB input capture by the ELCF event input	R/W
b22	BSELCG	ELCG Event Source GTCCRB Input Capture Enable	0: Disables GTCCRB input capture by the ELCG event input 1: Enables GTCCRB input capture by the ELCG event input	R/W

Bit	Symbol	Bit Name	Description	R/W
b23	BSELCH	ELCH Event Source GTCCRB Input Capture Enable	0: Disables GTCCRB input capture by the ELCH event input 1: Enables GTCCRB input capture by the ELCH event input	R/W
b24	BSOC	Other Channel Source GTCCRB Input Capture Enable	0: Disables GTCCRB input capture by the other channel factor 1: Enables GTCCRB input capture by the other channel factor	R/W
b31 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

n = 0 to 7

Note 1. When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.

The GTICBSR register sets an input capture source for the GTCCRB register.
When at least one bit among bits in the GTICBSR register is set to 1, input capture operation making the GTCCRB register as an input capture register is performed.
Inputs from the GTETRGA, GTETRGB, GTETRGC, and GTETRGD pins are input to the GPTW via the POEG. Set the polarity of these signals with the POEG.

BSOC Bit (Other Channel Source GTCCRB Input Capture Enable)

Select enable or disable for an input capture to the GTCCRB register by other channel factor.
Input capture of other channel factors is not subject to input capture factors to other channels set by the GTICCR.ICAFB, ICBFB bits.

24.2.12 General PWM Timer Control Register (GTCR)

- GPTW0.GTCR, GPTW1.GTCR, GPTW2.GTCR, GPTW4.GTCR, GPTW5.GTCR, GPTW6.GTCR

Address(es): GPTW0.GTCR 000C 202Ch, GPTW1.GTCR 000C 212Ch, GPTW2.GTCR 000C 222Ch, GPTW4.GTCR 000C 242Ch, GPTW5.GTCR 000C 252Ch, GPTW6.GTCR 000C 262Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	CKEG[1:0]		TPCS[3:0]				—	—	—	MD[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SSCEN	—	—	—	SSCGRP[1:0]		SCGTIOC	ICDS	—	—	—	—	—	—	—	CST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CST	Count Start	0: Count operation is stopped 1: Count operation is started	R/W *1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	ICDS	Input Capture Operation Select at Count Stop	0: Input capture is operated at count stop 1: Input capture is not operated at count stop	R/W
b9	SCGTIOC	GTIOC input Source Synchronous Clear Enable	0: Disables to use the counter clear by GTIOC input as the clear factor for other channels 1: Enables to use the counter clear by GTIOC input as the clear factor for other channels	R/W
b11, b10	SSCGRP[1:0]	Synchronous Set/Clear Group Select	b11 b10 0 0: Select synchronous set/clear group A 0 1: Select synchronous set/clear group B 1 0: Select synchronous set/clear group C 1 1: Select synchronous set/clear group D	R/W *1
b14 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	SSCEN	Synchronous Set/Clear Enable	0: Disable Synchronous set/clear of the GTCNT counter 1: Enable Synchronous set/clear of the GTCNT counter	R/W *1
b19 to b16	MD[3:0]	Mode Select	b19 b16 0 0 0 0: Sawtooth-wave PWM mode 1 (single buffer or double buffer possible) 0 0 0 1: Sawtooth-wave one-shot pulse mode (fixed buffer operation) 0 0 1 0: Sawtooth-wave PWM mode 2 (single buffer or double buffer possible) 0 0 1 1: Setting prohibited 0 1 0 0: Triangle-wave PWM mode 1 (32-bit transfer at trough) (single buffer or double buffer possible) 0 1 0 1: Triangle-wave PWM mode 2 (32-bit transfer at crest and trough) (single buffer or double buffer possible) 0 1 1 0: Triangle-wave PWM mode 3 (64-bit transfer at trough) (fixed buffer operation) 0 1 1 1: Setting prohibited 1 0 0 0: Setting prohibited 1 0 0 1: Setting prohibited 1 0 1 0: Setting prohibited 1 0 1 1: Setting prohibited 1 1 0 0: Complementary PWM mode 1 (transfer at crest) 1 1 0 1: Complementary PWM mode 2 (transfer at trough) 1 1 1 0: Complementary PWM mode 3 (transfer at crest and trough) 1 1 1 1: Complementary PWM mode 4 (immediate transfer)	R/W *1
b22 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit Name	Description	R/W
b26 to b23	TPCS[3:0]	Timer Prescaler Select	b26 b23 0 0 0 0: PCLKC 0 0 0 1: PCLKC/2 0 0 1 0: PCLKC/4 0 0 1 1: PCLKC/8 0 1 0 0: PCLKC/16 0 1 0 1: PCLKC/32 0 1 1 0: PCLKC/64 0 1 1 1: PCLKC/128 1 0 0 0: PCLKC/256 1 0 0 1: PCLKC/512 1 0 1 0: PCLKC/1024 1 0 1 1: Setting prohibited 1 1 0 0: GTETRGA (via the POEG) 1 1 0 1: GTETRGB (via the POEG) 1 1 1 0: GTETRGC (via the POEG) 1 1 1 1: GTETRGD (via the POEG)	R/W *1
b28, b27	CKEG[1:0]	Clock Edge Select	b28 b27 0 0: Select rising edge of GTETRG for clock count 0 1: Select falling edge of GTETRG for clock count 1 0: Select both edge of GTETRG for clock count 1 1: Select both edge of GTETRG for clock count	R/W *1
b31 to b29	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.

The GTCR register controls the GTCNT counter.

CST Bit (Count Start)

This bit controls start or stop for the GTCNT counter.

[Setting conditions]

- When 1 is written to a bit related to a channel number for the GTSTR register while the GTSSR.CSTRT bit is set to 1
- When an ELC event input, an external trigger, or a condition for the GTIOCnA pin input and GTIOCnB pin input (n = 0 to 7), which is enabled as a count start source by the GTSSR register, is generated
- 1 is written by software

[Clearing conditions]

- When 1 is written to a bit related to a channel number for the GTSTP register while the GTPSR.CSTOP bit is set to 1
- When an ELC event input, an external trigger, or a condition for the GTIOCnA pin input and GTIOCnB pin input, which is enabled as a count stop source by the GTPSR register, is generated
- A cycle-counting operation is completed while the GTPC.ASTP bit is 1
- 0 is written by software

ICDS Bit (Input Capture Operation Select at Count Stop)

This bit selects Input capture operation at count stop when the input capture function is selected.

SCGTIOC Bit (GTIOC input Source Synchronous Clear Enable)

This bit selects enables or disables the use of counter clearing at the GTIOCnA/GTIOCnB input pin selected by the GTCSR register as a counter clear factor for other channels.

SSCGRP[1:0] Bits (Synchronous Set/Clear Group Select)

This bit selects the synchronization set/clear channel group.

SSCEN Bit (Synchronous Set/Clear Enable)

This bit selects disable or enable of Synchronous set/clear.
In complementary PWM mode, slave channels are also controlled by setting the SSCEN bits of the master channel.

MD[3:0] Bits (Mode Select)

These bits select the GPTW operating mode.
In complementary PWM mode, the master channel setting of the MD[3:0] bits is also used to control the slave channels.
Only the MD[3:2] bits are valid at input capture.
Counting in sawtooth-wave mode is performed with 00b for the MD[3:2] bits, counting in triangle-wave mode is performed with 01b for the MD[3:2] bits.
Set the MD[3:0] bits while the GTCNT counter operation is stopped.
During the event count operation (when at least one bit among the bits for the GTUPSR and GTDNSR registers is set to 1), setting of the MD[3:0] bits are ignored, where counting in sawtooth-wave, triangle modes or complementary PWM mode is not performed. Instead, up-counting or down-counting by a source set by the GTUPSR and GTDNSR registers is performed.

TPCS[3:0] Bits (Timer Prescaler Select)

These bits select a clock for the GTCNT counter. The clock source can be selected independently for each channel.
Set the TPCS[3:0] bits while the GTCNT counter operation is stopped.

CKEG[1:0] Bits (Clock Edge Select)

When GTETRG input is selected by TPCS[3:0] bits, select the edge of GTETRG input used as the clock of GTCNT counter.
Set the CKEG[1:0] bits only when the GTCNT counter operation is stopped.

- GPTW3.GTCR, GPTW7.GTCR

Address(es):
GPTW3.GTCR 000C 232Ch, GPTW7.GTCR 000C 272Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	CKEG[1:0]		TPCS[3:0]				—	—	—	—	MD[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SSCEN	—	—	—	SSCGRP[1:0]		SCGTIOC	ICDS	—	—	—	—	—	—	—	CST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CST	Count Start	0: Count operation is stopped 1: Count operation is started	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	ICDS	Input Capture Operation Select at Count Stop	0: Input capture is operated at count stop 1: Input capture is not operated at count stop	R/W
b9	SCGTIOC	GTIOC input Source Synchronous Clear Enable	0: Disables to use the counter clear by GTIOC input as the clear factor for other channels 1: Enables to use the counter clear by GTIOC input as the clear factor for other channels	R/W
b11, b10	SSCGRP[1:0]	Synchronous Set/Clear Group Select	b11 b10 0 0: Select synchronous set/clear group A 0 1: Select synchronous set/clear group B 1 0: Select synchronous set/clear group C 1 1: Select synchronous set/clear group D	R/W
b14 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	SSCEN	Synchronous Set/Clear Enable	0: Disable Synchronous set/clear of the GTCNT counter 1: Enable Synchronous set/clear of the GTCNT counter	R/W
b18 to b16	MD[2:0]	Mode Select	b18 b16 0 0 0: Sawtooth-wave PWM mode (single buffer or double buffer possible) 0 0 1: Sawtooth-wave one-shot pulse mode (fixed buffer operation) 0 1 0: Setting prohibited 0 1 1: Setting prohibited 1 0 0: Triangle-wave PWM mode 1 (32-bit transfer at trough) (single buffer or double buffer possible) 1 0 1: Triangle-wave PWM mode 2 (32-bit transfer at crest and trough) (single buffer or double buffer possible) 1 1 0: Triangle-wave PWM mode 3 (64-bit transfer at trough) (fixed buffer operation) 1 1 1: Setting prohibited	R/W
b22 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit Name	Description	R/W
b26 to b23	TPCS[3:0]	Timer Prescaler Select	b26 b23 0 0 0 0: PCLKC 0 0 0 1: PCLKC/2 0 0 1 0: PCLKC/4 0 0 1 1: PCLKC/8 0 1 0 0: PCLKC/16 0 1 0 1: PCLKC/32 0 1 1 0: PCLKC/64 0 1 1 1: PCLKC/128 1 0 0 0: PCLKC/256 1 0 0 1: PCLKC/512 1 0 1 0: PCLKC/1024 1 0 1 1: Setting prohibited 1 1 0 0: GTETRGA (via the POEG) 1 1 0 1: GTETRGB (via the POEG) 1 1 1 0: GTETRGC (via the POEG) 1 1 1 1: GTETRGD (via the POEG)	R/W
b28, b27	CKEG[1:0]	Clock Edge Select	b28 b27 0 0: Select rising edge of GTETRG for clock count 0 1: Select falling edge of GTETRG for clock count 1 0: Select both edge of GTETRG for clock count 1 1: Select both edge of GTETRG for clock count	R/W
b31 to b29	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTCR register controls the GTCNT counter.

CST Bit (Count Start)

This bit controls start or stop for the GTCNT counter.

[Setting conditions]

- When 1 is written to a bit related to a channel number for the GTSTR register while the GTSSR.CSTRT bit is set to 1
- When an ELC event input, an external trigger, or a condition for the GTIOCnA pin input and GTIOCnB pin input (n = 0 to 7), which is enabled as a count start source by the GTSSR register, is generated
- 1 is written by software

[Clearing conditions]

- When 1 is written to a bit related to a channel number for the GTSTP register while the GTPSR.CSTOP bit is set to 1
- When an ELC event input, an external trigger, or a condition for the GTIOCnA pin input and GTIOCnB pin input, which is enabled as a count stop source by the GTPSR register, is generated
- A cycle-counting operation is completed while the GTPC.ASTP bit is 1
- 0 is written by software

ICDS Bit (Input Capture Operation Select at Count Stop)

This bit selects Input capture operation at count stop when the input capture function is selected.

SCGTIOC Bit (GTIOC input Source Synchronous Clear Enable)

This bit selects enables or disables the use of counter clearing at the GTIOCnA/GTIOCnB input pin selected by the GTCSR register as a counter clear factor for other channels.

SSCGRP[1:0] Bits (Synchronous Set/Clear Group Select)

This bit selects the synchronization set/clear channel group.

SSCEN Bit (Synchronous Set/Clear Enable)

This bit selects disable or enable of Synchronous set/clear.
In complementary PWM mode, slave channels are also controlled by setting the SSCEN bits of the master channel.

MD[2:0] Bits (Mode Select)

These bits select the GPTW operating mode.
Only the MD[2] bit is valid at input capture.
Counting in sawtooth-wave mode is performed with 0 for the MD[2] bit, counting in triangle-wave mode is performed with 1 for the MD[2] bit.
Set the MD[2:0] bits while the GTCNT counter operation is stopped.
During the event count operation (when at least one bit among the bits for the GTUPSR and GTDNSR registers is set to 1), setting of the MD[2:0] bits are ignored, where counting in sawtooth-wave or triangle modes is not performed. Instead, up-counting or down-counting by a source set by the GTUPSR and GTDNSR registers is performed.

TPCS[3:0] Bits (Timer Prescaler Select)

These bits select a clock for the GTCNT counter. The clock source can be selected independently for each channel.
Set the TPCS[3:0] bits while the GTCNT counter operation is stopped.
When the GTETRGA, GTETRGB, GTETRGC, or GTETRGD is selected, output for the POEG at rising becomes a clock source. Set the polarity of these signals with the POEG.

CKEG[1:0] Bits (Clock Edge Select)

When GTETRG input is selected by TPCS[3:0] bits, select the edge of GTETRG input used as the clock of GTCNT counter.
Set the CKEG[1:0] bits only when the GTCNT counter operation is stopped.

24.2.13 General PWM Timer Count Direction and Duty Setting Register (GTUDDTYC)

Address(es): GPTW0.GTUDDTYC 000C 2030h, GPTW1.GTUDDTYC 000C 2130h, GPTW2.GTUDDTYC 000C 2230h, GPTW3.GTUDDTYC 000C 2330h, GPTW4.GTUDDTYC 000C 2430h, GPTW5.GTUDDTYC 000C 2530h, GPTW6.GTUDDTYC 000C 2630h, GPTW7.GTUDDTYC 000C 2730h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	OBDTYR	OBDTYF	OBDTY[1:0]		—	—	—	—	OADTYR	OADTYF	OADTY[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	UDF	UD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	UD	Count Direction Setting	0: GTCNT counts down. 1: GTCNT counts up.	R/W
b1	UDF	Forcible Count Direction Setting	0: Count direction is not forcibly set. 1: Count direction is forcibly set.	R/W
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b17, b16	OADTY[1:0]	GTIOCnA Pin Output Duty Setting	b17b16 0 x: Compare matches determine the duty cycle of the output on the GTIOCnA pin. 1 0: The duty cycle of the output on the GTIOCnA pin is 0%. 1 1: The duty cycle of the output on the GTIOCnA pin is 100%.	R/W
b18	OADTYF	GTIOCnA Pin Output Duty Forced Setting	0: Duty of the GTIOCnA pin output is not forcibly set. 1: Duty of the GTIOCnA pin output is forcibly set.	R/W
b19	OADTYR	Output after Release of GTIOCnA Pin Output 0%/ 100% Duty Cycle Settings	0: The function selected by the GTIOA[3:2] bits is applied to the output value when the duty cycle is set after release from the 0 or 100% duty-cycle setting. 1: The function selected by the GTIOA[3:2] bits is applied to the compare match output value which is masked after release from the 0 or 100% duty-cycle setting.	R/W
b23 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b25, b24	OBDTY[1:0]	GTIOCnB Pin Output Duty Setting	b25b24 0 x: Compare matches determine the duty cycle of the output on the GTIOCnB pin 1 0: The duty cycle of the output on the GTIOCnB pin is 0%. 1 1: The duty cycle of the output on the GTIOCnB pin is 100%.	R/W
b26	OBDTYF	GTIOCnB Pin Output Duty Forced Setting	0: Duty of the GTIOCnB pin output is not forcibly set. 1: Duty of the GTIOCnB pin output is forcibly set.	R/W
b27	OBDTYR	Output after Release of GTIOCnB Pin Output 0%/ 100% Duty Cycle Settings	0: The function selected by the GTIOB[3:2] bits is applied to the output value when the duty cycle is set after release from the 0 or 100% duty-cycle setting. 1: The function selected by the GTIOB[3:2] bits is applied to the compare match output value which is masked after release from the 0 or 100% duty-cycle setting.	R/W
b31 to b28	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

n = 0 to 7

The GTUDDTYC register sets the count direction (up-counting/down-counting) for the GTCNT counter and configures duty setting for output from the GTIOCnA and GTIOCnB pins.
The setting is invalid during the event count operation, sawtooth-wave PWM mode 2, Complementary PWM mode.

(1) Setting of Count Direction

- In sawtooth-wave mode

When the UD bit value is set to 0 during up-counting, the count direction is changed at an overflow (count clock when the GTCNT counter value is equal to the GTPR register value). When the UD bit value is set to 1 during down-counting, the count direction is changed at an underflow (count clock when the GTCNT counter value is equal to 0000 0000h).
If the UD bit value is changed from 1 to 0 with the UDF bit being 0 and while counting is stopped, the counter starts up-counting and the count direction is changed at an overflow (count clock when the GTCNT counter value is equal to the GTPR register value).
If the UD bit value is changed from 0 to 1 with the UDF bit being 0 and while counting is stopped, the counter starts down-counting and the count direction is changed at an underflow (count clock when the GTCNT counter value is equal to 0000 0000h).
When the UDF bit is set to 1 while counting is stopped, the UD bit value at that time is reflected in the count direction when counting starts.

- In triangle-wave mode

Even if the UD bit value is changed during counting, the change will not be reflected in the count direction.
If the UD bit value is modified while the UDF bit is 0 and counting is stopped, the change will not be reflected in the count direction when counting starts. If the UDF bit is set to 1 while counting is stopped, the UD bit value at that time is reflected in the count direction when counting starts.

UD Bit (Count Direction Setting)

This bit sets the count direction (up-counting or down-counting) for the GTCNT counter.

UDF Bit (Forcible Count Direction Setting)

This bit forcibly sets the count direction when the GTCNT counter starts operation as the UD bit value.
Only 0 should be written to this bit during count operation.
When 1 has been written to this bit while counting is stopped, this bit should be returned to 0 before counting starts.

(2) Setting of Output Duty

- In sawtooth-wave mode

When the OADTY[1:0]/OBDTY[1:0] bits are changed during up-count operation, the duty setting changed at an overflow is reflected.
When the OADTY[1:0]/OBDTY[1:0] bits are changed during down-count operation, the duty setting changed at an underflow is reflected.
When the OADTYF/OBDTYF bit is 0 while count operation is stopped and the OADTY[1:0]/OBDTY[1:0] bits are changed, the change in the duty-cycle setting is not reflected in the first count operation, but the change is reflected on an overflow in the case of counting up and on an underflow in the case of counting down.
When the OADTYF/OBDTYF bit is 1 while count operation is stopped and the OADTY[1:0]/OBDTY[1:0] bits are changed, the change in the duty-cycle setting is immediately reflected during the first count operation.

- In triangle-wave mode

When the OADTY[1:0]/OBDTY[1:0] bits are changed during count operation, the duty-cycle setting changed at an underflow is reflected.
When the OADTYF/OBDTYF bit is 0 during count operation and the OADTY[1:0]/OBDTY[1:0] bits are changed, the duty-cycle setting changed during the first count operation is not reflected, the duty-cycle setting changed at an underflow is reflected.
When the OADTYF/OBDTYF bit is 1 while count operation is stopped and the OADTY[1:0]/OBDTY[1:0] bits are changed, the change in the duty-cycle setting is immediately reflected during the first count operation.

Whether it’s a sawtooth wave or a triangle wave, When the OADTYF/OBDTYF bit is set to 1 during the count stop and

the duty of the first cycle after the count start is set to the OADTY[1:0]/OBDTY[1:0] bits, Even if the OADTYF/OBDTYF bit is set to 0 and the OADTY[1:0]/OBDTY[1:0] bits are set again, When the count starts, the duty of the first cycle and the duty of the next cycle will be the duty setting of the OADTY[1:0]/OBDTY[1:0] bits set during the count stop.

OmDTY[1:0] Bits (GTIOCnm Pin Output Duty Setting) (n = 0 to 7; m = A, B)

These bits set the output of the duty cycle (0%, 100% or compare match control) on the GTIOCnm pin.

OmDTYF Bit (GTIOCnm Pin Output Duty Forced Setting) (n = 0 to 7; m = A, B)

This bit forcibly specifies the duty cycle at the start of the GTCNT counter in the OmDTY[1:0] bits.
Always write 0 to the bit during count operation.
When writing 1 to the bit while count operation is stopped, set the bit back to 0 by the end of the first cycle after the count operation started and specify the next cycle.

OmDTYR Bit (Output after Release of GTIOCnm Pin Output 0%/100% Duty Cycle Settings) (n = 0 to 7; m = A, B)

When the setting of a GTIOCnm pin for a 0% or 100% duty cycle is forcibly changed due to a compare match, the output level on the pin is maintained after the period has elapsed if the value of the GTIOR.GTIOm[3:2] bits is 00b, but maintenance and a value to control toggling are selected after the period has elapsed if the output has become toggled and the setting is 11b.
The GPTW internally continues to perform compare match operation during duty-cycle 0% or 100% operation. When the OmDTYR bit is 1, the value after the period has elapsed due this compare match operation is target for the GTIOm[3:2] bits.

24.2.14 General PWM Timer I/O Control Register (GTIOR)

Address(es): GPTW0.GTIOR 000C 2034h, GPTW1.GTIOR 000C 2134h, GPTW2.GTIOR 000C 2234h, GPTW3.GTIOR 000C 2334h, GPTW4.GTIOR 000C 2434h, GPTW5.GTIOR 000C 2534h, GPTW6.GTIOR 000C 2634h, GPTW7.GTIOR 000C 2734h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	NFCSB[1:0]		NFBEN	—	OBEOCD	OBDF[1:0]		OBE	OBHLD	OBDFLT	—	GTIOB[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	NFCSA[1:0]		NFAEN	PSYE	OAEOCD	OADF[1:0]		OAE	OAHLD	OADFLT	CPSCIR	GTIOA[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	GTIOA[4:0]	GTIOCnA Pin Function Select	See Table 24.4 and Table 24.5.	R/W
b5	CPSCIR	Complementary PWM Mode Initial Output at Synchronous Clear Disable	0: Output the initial value set by the GTIOR.GTIOA[4:0] and GTIOB[4:0] bits when synchronous clear occurs in Trough section of complementary PWM mode 1: Disable output the initial value	R/W
b6	OADFLT	GTIOCnA Pin Output Value Setting at the Count Stop	0: The GTIOCnA pin outputs low when counting is stopped. 1: The GTIOCnA pin outputs high when counting is stopped.	R/W
b7	OAHLD	GTIOCnA Pin Output Retention at the Start/Stop Count	0: The GTIOCnA pin output level at start/stop of counting depends on the register setting. 1: The GTIOCnA pin output level is retained at start/stop of counting.	R/W
b8	OAE	GTIOCnA Pin Output Enable	0: Pin output is disabled 1: Pin output is enabled	R/W
b10, b9	OADF[1:0]	GTIOCnA Pin Negate Value Setting	b10 b9 0 0: None of the following sources is specified 0 1: GTIOCnA pin is placed in the Hi-Z state in response to control for output negation. 1 0: GTIOCnA pin is set to 0 in response to control for output negation. 1 1: GTIOCnA pin is set to 1 in response to control for output negation.	R/W
b11	OAEOCD	GTCCRA Register Compare Match Cycle End Output Invalidate	0: Validate GTIOA[3:0] setting 1: Invalidate GTIOA[3:0] setting (GTIOCnA pin output is retained)	R/W
b12	PSYE	PWM Synchronous Output Enable	0: Disable GTCPPOn pin output 1: Enable GTCPPOn pin output	R/W
b13	NFAEN	GTIOCnA Pin Input Noise Filter Enable	0: The noise filter for GTIOCnA pin input is disabled 1: The noise filter for GTIOCnA pin input is enabled	R/W
b15, b14	NFCSA[1:0]	GTIOCnA Pin Input Noise Filter Sampling Clock Select	b15b14 0 0: PCLKC 0 1: PCLKC/4 1 0: PCLKC/16 1 1: PCLKC/64	R/W
b20 to b16	GTIOB[4:0]	GTIOCnB Pin Function Select	See Table 24.4.	R/W
b21	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b22	OBDFLT	GTIOCnB Pin Output Value Setting at the Count Stop	0: The GTIOCnB pin outputs low when counting is stopped. 1: The GTIOCnB pin outputs high when counting is stopped.	R/W
b23	OBHLD	GTIOCnB Pin Output Retention at the Start/Stop Count	0: The GTIOCnB pin output level at start/stop of counting depends on the register setting. 1: The GTIOCnB pin output level is retained at start/stop of counting.	R/W
b24	OBE	GTIOCnB Pin Output Enable	0: Pin output is disabled 1: Pin output is enabled	R/W

Bit	Symbol	Bit Name	Description	R/W
b26, b25	OBDF[1:0]	GTIOCnB Pin Negate Value Setting	b26b25 0 0: None of the following sources is specified 0 1: GTIOCnB pin is placed in the Hi-Z state in response to control for output negation. 1 0: GTIOCnB pin is set to 0 in response to control for output negation. 1 1: GTIOCnB pin is set to 1 in response to control for output negation.	R/W
b27	OBEOCD	GTCCRB Register Compare Match Cycle End Output Invalidate	• When sawtooth-wave PWM mode 1 0: Validate GTIOB[3:2] setting 1: Invalidate GTIOB[3:2] setting (GTIOCnB pin output is retained) • When sawtooth-wave PWM mode 2 0: Validate GTIOA[3:2] setting 1: Invalidate GTIOA[3:2] setting (GTIOCnA pin output is retained)	R/W
b28	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b29	NFBEN	GTIOCnB Pin Input Noise Filter Enable	0: The noise filter for GTIOCnB pin input is disabled 1: The noise filter for GTIOCnB pin input is enabled	R/W
b31, b30	NFCSB[1:0]	GTIOCnB Pin Input Noise Filter Sampling Clock Select	b31b30 0 0: PCLKC 0 1: PCLKC/4 1 0: PCLKC/16 1 1: PCLKC/64	R/W

n = 0 to 7

GTIOR sets the functions of the GTIOCnA, GTIOCnB, and GTCPPOn pins.

GTIOA[4:0] Bits (GTIOCnA Pin Function Select) (n = 0 to 7)

These bits select the GTIOCnA pin function. For details, see Table 24.4.

CPSCIR Bit (Complementary PWM Mode Initial Output at Synchronous Clear Disable)

Select the output waveform when synchronous clear occurs in complementary PWM mode.

The initial output is disabled by this function only when synchronous clear occurs in the trough in complementary PWM mode. If a synchronous clear occurs at any other time, the initial value set by the GTIOA[4]/GTIOB[4] bits is output regardless of the CPSCIR bit setting. In addition, the initial value set by the GTIOA[4]/GTIOB[4] bits is output even when the synchronous clear occurs in the trough immediately after the count starts.

OADFLT Bit (GTIOCnA Pin Output Value Setting at the Count Stop) (n = 0 to 7)

This bit sets whether the GTIOCnA pin outputs high or low when counting is stopped.

OAHLD Bit (GTIOCnA Pin Output Retention at the Start/Stop Count) (n = 0 to 7)

This bit specifies whether the GTIOCnA pin output level is retained or the level depends on the register setting when counting is started or stopped.

[Setting conditions]

- The output is retained when counting starts or stops.

[Clearing conditions]

- The value specified by the GTIOA[4] bit is output when counting starts.
- The value specified by the OADFLT bit is output when counting stops.
- If the OADFLT bit is modified while counting is stopped, it is immediately reflected in the output.

OAE Bit (GTIOCnA Pin Output Enable) (n = 0 to 7)

This bit selects whether the GTIOCnA pin output is performed or not.

When the GTCCRA register is used as an input capture register (at least one bit among the bits for the GTICASR register is set to 1), the GTIOCnA pin output is not performed regardless of the setting of the OAE bit.

OADF[1:0] Bits (GTIOCnA Pin Negate Value Setting) (n = 0 to 7)

Select the value for output from the GTIOCnA pin in response to a request to disable output from the POEG.

OAEOCD Bit (GTCCRA Register Compare Match Cycle End Output Invalidate)

When the end of the cycle and the timing of the compare match of the GTCCRA register match in sawtooth-wave PWM mode 1 and 2, select whether to disable or enable the GTIOA[3:2] bit setting.
When this bit is 1 (disabled), the GTIOCnA pin holds the output when the cycle end and the GTCCRA compare match timing match.

PSYE Bit (PWM Synchronous Output Enable)

This bit sets enable or disable of output signal from GTCPPOn pin synchronized with the PWM cycle that toggles at the of complementary PWM mode, crest/trough/GTCNT counter clear at Triangle-wave mode, or the end of the cycle of the Sawtooth-wave mode.

NFAEN Bit (GTIOCnA Pin Input Noise Filter Enable) (n = 0 to 7)

This bit sets enable or disable of the noise filter function for the GTIOCnA pin input. Switch these bits while the corresponding pin function for the timer I/O control register is set as the output compare function, because unintended internal edge may be generated when these bits are switched.

NFCSA[1:0] Bits (GTIOCnA Pin Input Noise Filter Sampling Clock Select) (n = 0 to 7)

These bits set a sampling clock for the noise filter to the GTIOCnA pin input. Set to the input capture function after waiting two cycles for the sampling clock cycle following the setting of these bits.

GTIOB[4:0] Bits (GTIOCnB Pin Function Select) (n = 0 to 7)

These bits select the GTIOCnB pin function. For details, see Table 24.4.
In sawtooth-wave PWM mode 2, only the GTIOB[1:0] bits are valid, and the GTIOCnA pin output is selected instead of the GTIOCnB pin by the GTCCRB register compare match.

OBDFLT Bit (GTIOCnB Pin Output Value Setting at the Count Stop) (n = 0 to 7)

This bit sets whether the GTIOCnB pin outputs high or low when counting is stopped.

OBHLD Bit (GTIOCnB Pin Output Retention at the Start/Stop Count) (n = 0 to 7)

This bit specifies whether the GTIOCnB pin output level is retained or the level depends on the register setting when counting is started or stopped.
[Setting conditions]

- The output is retained when counting starts or stops.

[Clearing conditions]

- The value specified by the GTIOB[4] bit is output when counting starts.
- The value specified by the OBDFLT bit is output when counting stops.
- If the OBDFLT bit is modified while counting is stopped, it is immediately reflected in the output.

OBE Bit (GTIOCnB Pin Output Enable) (n = 0 to 7)

This bit selects whether the GTIOCnB pin output is performed or not.
When the GTCCRB register is used as an input capture register (at least one bit among the bits for the GTICBSR register is set to 1), the GTIOCnB pin output is not performed regardless of the setting of the OBE bit.

OBDF[1:0] Bits (GTIOCnB Pin Negate Value Setting) (n = 0 to 7)

Select the value for output from the GTIOCnB pin in response to a request to disable output from the POEG.

OBEOCD Bit (GTCCRB Register Compare Match Cycle End Output Invalidate)

When the end of the cycle and the timing of the compare match of the GTCCRB register match in sawtooth-wave PWM mode 1 and 2, select whether to disable or enable the GTIOB[3:2] bit setting.
When this bit is 1 (disabled), the GTIOCnB pin holds the output when the cycle end and the GTCCRB compare match timing match.

NFBEN Bit (GTIOCnB Pin Input Noise Filter Enable) (n = 0 to 7)

This bit sets enable or disable of the noise filter function for the GTIOCnB pin input. Switch these bits while the corresponding pin function for the timer I/O control register is set as the output compare function, because unintended internal edge may be generated when these bits are switched.

NFCSB[1:0] Bits (GTIOCnB Pin Input Noise Filter Sampling Clock Select) (n = 0 to 7)

These bits set a sampling clock for the noise filter to the GTIOCnB pin input. Set to the input capture function after waiting two cycles for the sampling clock cycle following the setting of these bits.

Table 24.4 Settings of GTIOA[4:0] Bits (GTIOB[4:0] Bits) (Sawtooth-wave mode, Triangle-wave mode) (1/2)

GTIOA[4:0] (GTIOB[4:0]) Bits					Function		
b4	b3	b2	b1	b0	b4	b3, b2	b1, b0
0	0	0	0	0	Initial output is Low.	Output retained at the end of a cycle	Output retained at GTCCRA/GTCCRB compare match
0	0	0	0	1			Low output at GTCCRA/GTCCRB compare match
0	0	0	1	0			High output at GTCCRA/GTCCRB compare match
0	0	0	1	1			Output toggled at GTCCRA/GTCCRB compare match
0	0	1	0	0		Low output at the end of a cycle	Output retained at GTCCRA/GTCCRB compare match
0	0	1	0	1			Low output at GTCCRA/GTCCRB compare match
0	0	1	1	0			High output at GTCCRA/GTCCRB compare match
0	0	1	1	1			Output toggled at GTCCRA/GTCCRB compare match
0	1	0	0	0		High output at the end of a cycle	Output retained at GTCCRA/GTCCRB compare match
0	1	0	0	1			Low output at GTCCRA/GTCCRB compare match
0	1	0	1	0			High output at GTCCRA/GTCCRB compare match
0	1	0	1	1			Output toggled at GTCCRA/GTCCRB compare match
0	1	1	0	0		Output toggled at the end of a cycle	Output retained at GTCCRA/GTCCRB compare match
0	1	1	0	1			Low output at GTCCRA/GTCCRB compare match
0	1	1	1	0			High output at GTCCRA/GTCCRB compare match
0	1	1	1	1			Output toggled at GTCCRA/GTCCRB compare match

Table 24.4 Settings of GTIOA[4:0] Bits (GTIOB[4:0] Bits) (Sawtooth-wave mode, Triangle-wave mode) (2/2)

GTIOA[4:0] (GTIOB[4:0]) Bits					Function		
b4	b3	b2	b1	b0	b4	b3, b2	b1, b0
1	0	0	0	0	Initial output is High.	Output retained at the end of a cycle	Output retained at GTCCRA/GTCCRB compare match
1	0	0	0	1			Low output at GTCCRA/GTCCRB compare match
1	0	0	1	0			High output at GTCCRA/GTCCRB compare match
1	0	0	1	1			Output toggled at GTCCRA/GTCCRB compare match
1	0	1	0	0		Low output at the end of a cycle	Output retained at GTCCRA/GTCCRB compare match
1	0	1	0	1			Low output at GTCCRA/GTCCRB compare match
1	0	1	1	0			High output at GTCCRA/GTCCRB compare match
1	0	1	1	1			Output toggled at GTCCRA/GTCCRB compare match
1	1	0	0	0		High output at the end of a cycle	Output retained at GTCCRA/GTCCRB compare match
1	1	0	0	1			Low output at GTCCRA/GTCCRB compare match
1	1	0	1	0			High output at GTCCRA/GTCCRB compare match
1	1	0	1	1			Output toggled at GTCCRA/GTCCRB compare match
1	1	1	0	0		Output toggled at the end of a cycle	Output retained at GTCCRA/GTCCRB compare match
1	1	1	0	1			Low output at GTCCRA/GTCCRB compare match
1	1	1	1	0			High output at GTCCRA/GTCCRB compare match
1	1	1	1	1			Output toggled at GTCCRA/GTCCRB compare match

Note: In sawtooth-wave mode, “end of a cycle” refers to an overflow (the value of the GTCNT counter changing from that of the GTPR register to 0000 0000h in up-counting), an underflow (the value of the GTCNT counter changing from 0000 0000h to that of the GTPR register in down-counting), or counter clearing. It refers to a trough in triangle-wave mode (the value of the GTCNT counter changing from 0000 0000h to 0000 0001h)).

Note: When the timing of an end of a cycle and the timing of a GTCCRA/GTCCRB register compare match are the same in a compare-match operation, if the OAEOCD and OBEOCD bits are set to 0 and the end of cycle output is enabled, the b3 and b2 settings are given priority in sawtooth-wave PWM mode, and the b1 and b0 settings are given priority in any other mode.

Note: During the event count operation (when at least one bit among the bits for the GTUPSR and GTDNSR registers is set to 1), the b3 and b2 settings become invalid.

Note: In sawtooth-wave PWM mode 2, the GTIOB[4:2] bits are invalid. Since only the GTIOCnA pin is the output pin, set the GTIOA[4] bit for the initial output. Set the GTIOA[3:2] bits for output at the end of the cycle.

Table 24.5 Settings of GTIOA[4:0] Bits (GTIOB[4:0] Bits) (Complementary PWM mode)

GTIOA[4:0] (GTIOB[4:0]) Bits					Function		
					Initial Output, Active level	Up-count Compare Match input	Down-count Compare Match output
b4	b3	b2	b1	b0	b4	b3, b2	b1, b0
0	0	1	1	0	Initial output is Low. Active level is High.	Low output	High output
0	1	0	0	1		High output	Low output
1	0	1	1	0	Initial output is High. Active level is Low.	Low output	High output
1	1	0	0	1		High output	Low output

Note: In complementary PWM mode, the only values that can be set in the GTIOA[4:0] bits are 01001b, and 10110b. Setting other values is prohibited.

Note: In complementary PWM mode, the only values that can be set in the GTIOB[4:0] bits are 00110b, and 11001b. Setting other values is prohibited.

Note: In complementary PWM mode, setting the GTIOB[4:0] bits does not use compare match of the GTCCRB register. The combination of counter and register that is the target of compare match depends on the operation period of complementary PWM mode. For details, see (7), Complementary PWM mode 1, 2, 3.

24.2.15 General PWM Timer Interrupt Output Setting Register (GTINTAD)

Address(es): GPTW0.GTINTAD 000C 2038h, GPTW1.GTINTAD 000C 2138h, GPTW2.GTINTAD 000C 2238h, GPTW3.GTINTAD 000C 2338h, GPTW4.GTINTAD 000C 2438h, GPTW5.GTINTAD 000C 2538h, GPTW6.GTINTAD 000C 2638h, GPTW7.GTINTAD 000C 2738h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	GTINTPC	GRPABL	GRPABH	GRPDTE	—	—	GRP[1:0]		—	—	—	—	ADTRBDEN	ADTRBUEN	ADTRADEN	ADTRAUEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SCFPU	SCFPO	SCFF	SCFE	SCFD	SCFC	SCFB	SCFA	GTINTPR[1:0]		GTINTF	GTINTE	GTINTD	GTINTC	GTINTB	GTINTA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	GTINTA	GTCCRA Register Compare Match/ Input Capture Interrupt Enable	0: Interrupt request is disabled. 1: Interrupt request is enabled.	R/W
b1	GTINTB	GTCCRB Register Compare Match/ Input Capture Interrupt Enable	0: Interrupt request is disabled. 1: Interrupt request is enabled.	R/W
b2	GTINTC	GTCCRC Register Compare Match Interrupt Enable	0: Interrupt request is disabled. 1: Interrupt request is enabled.	R/W
b3	GTINTD	GTCCRD Register Compare Match Interrupt Enable	0: Interrupt request is disabled. 1: Interrupt request is enabled.	R/W
b4	GTINTE	GTCCRE Register Compare Match Interrupt Enable	0: Interrupt request is disabled. 1: Interrupt request is enabled.	R/W
b5	GTINTF	GTCCRF Register Compare Match Interrupt Enable	0: Interrupt request is disabled. 1: Interrupt request is enabled.	R/W
b7, b6	GTINTPR[1:0]	GTPR Register Compare Match Interrupt Enable	b7 b6 0 0: Interrupt request is disabled. 0 1: In sawtooth-wave mode, interrupt requests are enabled at overflows. In triangle-wave mode, interrupt requests are enabled at crests. 1 0: In sawtooth-wave mode, interrupt requests are enabled at underflows. In triangle-wave mode, interrupt requests are enabled at troughs. 1 1: In sawtooth-wave mode, interrupt requests are enabled at both overflows and underflows. In triangle-wave mode, interrupt requests are enabled at both crests and troughs.	R/W
b8	SCFA	GTCCRA Register Compare Match/ Input Capture Source Synchronous Clear Enable	0: Disables the GTCCRA register comparison match/ input capture as a source for other channel clears. 1: Enables the GTCCRA register comparison match/ input capture as a source for other channel clears.	R/W
b9	SCFB	GTCCRB Register Compare Match Source Synchronous Clear Enable	0: Disables the GTCCRB register comparison match/ input capture as a source for other channel clears. 1: Enables the GTCCRB register comparison match/ input capture as a source for other channel clears.	R/W
b10	SCFC	GTCCRC Register Compare Match Source Synchronous Clear Enable	0: Disables the GTCCRC register comparison match/ input capture as a source for other channel clears. 1: Enables the GTCCRC register comparison match/ input capture as a source for other channel clears.	R/W
b11	SCFD	GTCCRD Register Compare Match Source Synchronous Clear Enable	0: Disables the GTCCRD register comparison match/ input capture as a source for other channel clears. 1: Enables the GTCCRD register comparison match/ input capture as a source for other channel clears.	R/W
b12	SCFE	GTCCRE Register Compare Match Source Synchronous Clear Enable	0: Disables the GTCCRE register comparison match/ input capture as a source for other channel clears. 1: Enables the GTCCRE register comparison match/ input capture as a source for other channel clears.	R/W

Bit	Symbol	Bit Name	Description	R/W
b13	SCFF	GTCCRF Register Compare Match Source Synchronous Clear Enable	0: Disables the GTCCRF register comparison match/ input capture as a source for other channel clears. 1: Enables the GTCCRF register comparison match/ input capture as a source for other channel clears.	R/W
b14	SCFPO	Overflow Source Synchronous Clear Enable	0: Disables Overflow as a source for other channel clears. 1: Enables Overflow as a source for other channel clears.	R/W
b15	SCFPU	Underflow Source Synchronous Clear Enable	0: Disables Underflow as a source for other channel clears. 1: Enables Underflow as a source for other channel clears.	R/W
b16	ADTRAUEN	GTADTRA Register Compare Match (Up-Counting) A/D Conversion Start Request Enable	0: A/D conversion start request is disabled. 1: A/D conversion start request is enabled.	R/W
b17	ADTRADEN	GTADTRA Register Compare Match (Down-Counting) A/D Conversion Start Request Enable	0: A/D conversion start request is disabled. 1: A/D conversion start request is enabled.	R/W
b18	ADTRBUEN	GTADTRB Register Compare Match (Up-Counting) A/D Conversion Start Request Enable	0: A/D conversion start request is disabled. 1: A/D conversion start request is enabled.	R/W
b19	ADTRBDEN	GTADTRB Register Compare Match (Down-Counting) A/D Conversion Start Request Enable	0: A/D conversion start request is disabled. 1: A/D conversion start request is enabled.	R/W
b23 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b25, b24	GRP[1:0]	Output Stop Group Select	b25b24 0 0: Group A is selected 0 1: Group B is selected 1 0: Group C is selected 1 1: Group D is selected	R/W
b27, b26	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b28	GRPDTE	Dead Time Error Output Stop Detection Enable	0: Dead time error output stop detection is disabled 1: Dead time error output stop detection is enabled	R/W
b29	GRPABH	Simultaneous High Output Stop Detection Enable	0: Simultaneous high output stop detection is disabled 1: Simultaneous high output stop detection is enabled	R/W
b30	GRPABL	Simultaneous Low Output Stop Detection Enable	0: Simultaneous low output stop detection is disabled 1: Simultaneous low output stop detection is enabled	R/W
b31	GTINTPC	Cycle Count End Interrupt Enable	0: Interrupt request is disabled. 1: Interrupt request is enabled.	R/W

The GTINTAD register sets enabling or disabling for interrupt request, A/D conversion start request, and output stop detection.

GTINTA Bit (GTCCRA Register Compare Match/Input Capture Interrupt Enable)

This bit enables or disables interrupt requests by the GTCCRA register compare match/input capture (GTCIA).

GTINTB Bit (GTCCRB Register Compare Match/Input Capture Interrupt Enable)

This bit enables or disables interrupt requests by the GTCCRB register compare match/input capture (GTCIB).

GTINTC Bit (GTCCRC Register Compare Match Interrupt Enable)

This bit enables or disables interrupt requests by the GTCCRC register compare match (GTCIC).

GTINTD Bit (GTCCRD Register Compare Match Interrupt Enable)

This bit enables or disables interrupt requests by the GTCCRD register compare match (GTCID).

GTINTE Bit (GTCCRE Register Compare Match Interrupt Enable)

This bit enables or disables interrupt requests by the GTCCRE register compare match (GTCIE).

GTINTF Bit (GTCCRF Register Compare Match Interrupt Enable)

This bit enables or disables interrupt requests by the GTCCRF register compare match (GTCIF).

GTINTPR[1:0] Bits (GTPR Register Compare Match Interrupt Enable)

These bits enable or disable interrupt requests by a GTPR register compare match (GTCNT counter overflow) and those by a GTCNT counter underflow (GTCIV/GTCIU).

SCFA Bit (GTCCRA Register Compare Match/Input Capture Source Synchronous Clear Enable)

This bit enables or disables the GTCCRA register compare match/input capture as a clearing factor for other channels.

SCFB Bit (GTCCRB Register Compare Match Source Synchronous Clear Enable)

This bit enables or disables the GTCCRB register compare match/input capture as a clearing factor for other channels.
This setting is invalid in complementary PWM mode.

SCFC Bit (GTCCRC Register Compare Match Source Synchronous Clear Enable)

This bit enables or disables the GTCCRC register compare match as a clearing factor for other channels.
This setting is invalid in complementary PWM mode.

SCFD Bit (GTCCRD Register Compare Match Source Synchronous Clear Enable)

This bit enables or disables the GTCCRD register compare match as a clearing factor for other channels.
This setting is invalid in complementary PWM mode.

SCFE Bit (GTCCRE Register Compare Match Source Synchronous Clear Enable)

This bit enables or disables the GTCCRE register compare match as a clearing factor for other channels.
This setting is invalid in complementary PWM mode.

SCFF Bit (GTCCRF Register Compare Match Source Synchronous Clear Enable)

This bit enables or disables the GTCCRF register compare match as a clearing factor for other channels.
This setting is invalid in complementary PWM mode.

SCFPO Bit (Overflow Source Synchronous Clear Enable)

This bit enables or prohibits overflow as a clearing factor for other channels.
In complementary PWM mode, this setting is valid only for the master channel.

SCFPU Bit (Underflow Source Synchronous Clear Enable)

This bit enables or prohibits underflow as a clearing factor for other channels.
In complementary PWM mode, this setting is valid only for the master channel.

ADTRAUEN Bit (GTADTRA Register Compare Match (Up-Counting) A/D Conversion Start Request Enable)

This bit enables or disables A/D conversion start requests generated by the GTADTRA register compare matches during GTCNT counter up-counting.
The setting is invalid during the event count operation, and the A/D conversion start request is not generated.

ADTRADEN Bit (GTADTRA Register Compare Match (Down-Counting) A/D Conversion Start Request Enable)

This bit enables or disables A/D conversion start requests generated by the GTADTRA register compare matches during GTCNT counter down-counting.
The setting is invalid during the event count operation, and the A/D conversion start request is not generated.

ADTRBUEN Bit (GTADTRB Register Compare Match (Up-Counting) A/D Conversion Start Request Enable)

This bit enables or disables A/D conversion start requests generated by the GTADTRB register compare matches during GTCNT counter up-counting.
The setting is invalid during the event count operation, and the A/D conversion start request is not generated.

ADTRBDEN Bit (GTADTRB Register Compare Match (Down-Counting) A/D Conversion Start Request Enable)

This bit enables or disables A/D conversion start requests generated by the GTADTRB register compare matches during GTCNT counter down-counting.
The setting is invalid during the event count operation, and the A/D conversion start request is not generated.

GRP[1:0] Bits (Output Stop Group Select)

Select the group to detect disabling of output from the GPTW to the POEG and to request of disabling of output from the POEG to the GPTW.
Each signal indicating detection of disabling in response to a dead-time error or simultaneous driving of outputs to the high or low level is output to the group selected in the GRP[1:0] bits while the respective output disable detection enable bits are 1.
Requests to disabling output from the POEG for the group selected in the GRP[1:0] bits can be monitored with the GTST.ODF flag.
Set the GRP[1:0] bits while both of the OAE and OBE bits in the GTIOR register are set to 0.

GRPDTE Bit (Dead Time Error Output Stop Detection Enable)

This bit enables or disables the output stop detection by a dead time error.
The dead time error output stop detection is not generated during the event count operation.

GRPABH Bit (Simultaneous High Output Stop Detection Enable)

This bit enables or disables the output stop detection when the GTIOCnA and GTIOCnB pins have become high simultaneously.

GRPABL Bit (Simultaneous Low Output Stop Detection Enable)

This bit enables or disables the output stop detection when the GTIOCnA and GTIOCnB pins have become low simultaneously.

GTINTPC Bit (Cycle Count End Interrupt Enable)

This bit enables or disables the request of the cycle count end interrupt (GTCEI).

24.2.16 General PWM Timer Status Register (GTST)

Address(es): GPTW0.GTST 000C 203Ch, GPTW1.GTST 000C 213Ch, GPTW2.GTST 000C 223Ch, GPTW3.GTST 000C 233Ch, GPTW4.GTST 000C 243Ch, GPTW5.GTST 000C 253Ch, GPTW6.GTST 000C 263Ch, GPTW7.GTST 000C 273Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PCF	OABLF	OABHF	DTEF	—	—	—	ODF	—	—	—	—	ADTRBDF	ADTRBUF	ADTRADF	ADTRAUF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TUCF	—	—	—	—	ITCNT[2:0]			TCFPU	TCFPO	TCFF	TCFE	TCFD	TCFC	TCFB	TCFA
Value after reset:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TCFA	Input Capture/Compare Match Flag A	0: No input capture/compare match of GTCCRA is generated. 1: An input capture/compare match of GTCCRA is generated.	R/(W) *1
b1	TCFB	Input Capture/Compare Match Flag B	0: No input capture/compare match of GTCCRB is generated. 1: An input capture/compare match of GTCCRB is generated.	R/(W) *1
b2	TCFC	Compare Match Flag C	0: No compare match of GTCCRC is generated. 1: An compare match of GTCCRC is generated.	R/(W) *1
b3	TCFD	Compare Match Flag D	0: No compare match of GTCCRD is generated. 1: An compare match of GTCCRD is generated.	R/(W) *1
b4	TCFE	Compare Match Flag E	0: No compare match of GTCCRE is generated. 1: An compare match of GTCCRE is generated.	R/(W) *1
5	TCFF	Compare Match Flag F	0: No compare match of GTCCRF is generated. 1: An compare match of GTCCRF is generated.	R/(W) *1
b6	TCFPO	Overflow Flag	0: No overflow (crest) occurred. 1: An overflow (crest) occurred.	R/(W) *1
b7	TCFPU	Underflow Flag	0: No underflow (trough) occurred. 1: An underflow (trough) occurred.	R/(W) *1
b10 to b8	ITCNT[2:0]	GTCIV/GTCIU Interrupt Skipping Count Counter	Counter for counting the number of times a timer interrupt has been skipped.	R
b14 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	TUCF	Count Direction Flag	0: The GTCNT counter counts downward. 1: The GTCNT counter counts upward.	R
b16	ADTRAUF	GTADTRA Register Compare Match (Up-Counting) A/D Conversion Start Request Flag	0: No GTADTRA register compare match has occurred in up-counting. 1: A GTADTRA register compare match has occurred in up-counting.	R/(W) *1
b17	ADTRADF	GTADTRA Register Compare Match (Down-Counting) A/D Conversion Start Request Flag	0: No GTADTRA register compare match has occurred in down-counting. 1: A GTADTRA register compare match has occurred in down-counting.	R/(W) *1
b18	ADTRBUF	GTADTRB Register Compare Match (Up-Counting) A/D Conversion Start Request Flag	0: No GTADTRB register compare match has occurred in up-counting. 1: A GTADTRB register compare match has occurred in up-counting.	R/(W) *1
b19	ADTRBDF	GTADTRB Register Compare Match (Down-Counting) A/D Conversion Start Request Flag	0: No GTADTRB register compare match has occurred in down-counting. 1: A GTADTRB register compare match has occurred in down-counting.	R/(W) *1
b23 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b24	ODF	Output Stop Request Flag	0: No output stop request has occurred. 1: An output stop request has occurred.	R
b27 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit Name	Description	R/W
b28	DTEF	Dead Time Error Flag	0: No dead time error has occurred. 1: A dead time error has occurred.	R
b29	OABHF	Simultaneous High Output Flag	0: No simultaneous generation of 1 both for the GTIOCA and GTIOCB pins has occurred. 1: A simultaneous generation of 1 both for the GTIOCA and GTIOCB pins has occurred.	R
b30	OABLF	Simultaneous Low Output Flag	0: No simultaneous generation of 0 both for the GTIOCA and GTIOCB pins has occurred. 1: A simultaneous generation of 0 both for the GTIOCA and GTIOCB pins has occurred.	R
b31	PCF	Cycle Count End Flag	0: No cycle count end has occurred. 1: A cycle count end has occurred.	R/(W) *1

Note 1. Only 0 can be written to clear the flag. When clearing the ADTRAUF, ADTRADF, ADTRBUF, or ADTRBDF flag, be sure to write 0 only to the target flag or flags for clearing and to write 1 to the other flags not for clearing.

The GTST register indicates the status of the GPTW.

TCFA Flag (Input Capture/Compare Match Flag A)

The TCFA flag indicates the status for the input capture or compare match of the GTCCRA register.
[Setting conditions]

- GTCNT = GTCCRA, when the GTCCRA register functions as a compare match register
- The GTCNT counter value is transferred to the GTCCRA register by the input capture signal when the GTCCRA register functions as an input capture register.

[Clearing condition]

- 0 is written to this flag.

TCFB Flag (Input Capture/Compare Match Flag B)

The TCFB flag indicates the status for the input capture or compare match of the GTCCRB register.
[Setting conditions]

- GTCNT = GTCCRB, when the GTCCRB register functions as a compare match register
- The GTCNT counter value is transferred to the GTCCRB register by the input capture signal when the GTCCRB register functions as an input capture register.

[Clearing condition]

- 0 is written to this flag.

TCFC Flag (Compare Match Flag C)

The TCFC flag indicates the status for the compare match of the GTCCRC register.
When the GTCCRC register performs buffer operation, the GTCCRC register doesn't perform compare match.
[Setting conditions]

- GTCNT = GTCCRC

[Clearing condition]

- 0 is written to this flag.

[Not comparing condition]

- GTCR.MD[2:0] = 001b (sawtooth-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (triangle-wave PWM mode 3)
- GTBER.CCRA[1:0] = 01b, 10b, 11b (the GTCCRC register performs buffer operation).

TCFD Flag (Compare Match Flag D)

The TCFD flag indicates the status for the compare match of the GTCCRD register.

When the GTCCRD register performs buffer operation, the GTCCRD register doesn't perform compare match.
[Setting conditions]

- GTCNT = GTCCRD

[Clearing condition]

- 0 is written to this flag.

[Not comparing condition]

- GTCR.MD[2:0] = 001b (sawtooth-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (triangle-wave PWM mode 3)
- GTBER.CCRA[1:0] = 01b, 10b, 11b (the GTCCRD register performs buffer operation).

TCFE Flag (Compare Match Flag E)

The TCFE flag indicates the status for the compare match of the GTCCRE register.
When the GTCCRE register performs buffer operation, the GTCCRE register doesn't perform compare match.
[Setting conditions]

- GTCNT = GTCCRE

[Clearing condition]

- 0 is written to this flag.

[Not comparing condition]

- GTCR.MD[2:0] = 001b (sawtooth-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (triangle-wave PWM mode 3)
- GTBER.CCRB[1:0] = 01b, 10b, 11b (the GTCCRE register performs buffer operation).

TCFF Flag (Compare Match Flag F)

The TCFF flag indicates the status for the compare match of the GTCCRF register.
When the GTCCRF register performs buffer operation, the GTCCRF register doesn't perform compare match.
[Setting conditions]

- GTCNT = GTCCRF

[Clearing condition]

- 0 is written to this flag.

[Not comparing condition]

- GTCR.MD[2:0] = 001b (sawtooth-wave one-shot pulse mode).
- GTCR.MD[2:0] = 110b (triangle-wave PWM mode 3)
- GTBER.CCRB[1:0] = 01b, 10b, 11b (the GTCCRF register performs buffer operation).

TCFPO Flag (Overflow Flag)

The TCFPO flag indicates when an overflow or crest has occurred.
[Setting conditions]

- In sawtooth-wave mode, an overflow (the GTCNT counter changes from the GTPR register value to 0 in up-counting) has occurred
- In triangle-wave mode, a crest (the GTCNT counter changes from the GTPR register value to GTPR – 1) has occurred
- In counting by hardware sources, an overflow (the GTCNT counter changes from the GTPR register value to 0 in up-counting) has occurred.

[Clearing condition]

- 0 is written to this flag.

TCFPU Flag (Underflow Flag)

The TCFPU flag indicates when an underflow or trough has occurred.
[Setting conditions]

- In sawtooth-wave mode, an underflow (the GTCNT counter changes from 0 to the GTPR register value in down-counting) has occurred
- In triangle-wave mode, a trough (the GTCNT counter changes from 0 to 1) has occurred
- In counting by hardware sources, an underflow (the GTCNT counter changes from 0 to the GTPR register value in down-counting) has occurred.

[Clearing condition]

- 0 is written to this flag.

ITCNT[2:0] Bits (GTCIV/GTCIU Interrupt Skipping Count Counter)

When the GTCIV/GTCIU interrupt skipping function is used (the GTITC.IVTC[1:0] bits are set to a value other than 00b), the counter is incremented by 1 every time the GTCIV/GTCIU interrupt source is generated.
These bits are operated independently from the extended interrupt skipping by the GTEITC register.
[Clearing conditions]

- The GTCIV/GTCIU interrupt skipping function is not used (the GTITC.IVTT[2:0] bits are 000b when the IVTC[1:0] bits are 00b).
- The GTCIV/GTCIU interrupt skipping count matches the specified count (the ITCNT[2:0] bits match the skipping count specified by the IVTT[2:0] bits).
- When the count operation is stopped.

TUCF Flag (Count Direction Flag)

This flag indicates the count direction of the GTCNT counter.
This flag indicates 1 when in up-counting and indicates 0 when in down-counting during the event count operation.

ADTRAUF Flag (GTADTRA Register Compare Match (Up-Counting) A/D Conversion Start Request Flag)

This status flag indicates generation of a GTADTRA register compare match in up-counting.
[Setting condition]

- The GTCNT counter matches the GTADTRA register in up-counting

[Clearing conditions]

- 0 is written to the ADTRAUF flag

ADTRADF Flag (GTADTRA Register Compare Match (Down-Counting) A/D Conversion Start Request Flag)

This status flag indicates generation of a GTADTRA register compare match in down-counting.
[Setting condition]

- The GTCNT counter matches the GTADTRA register in down-counting

[Clearing conditions]

- 0 is written to the ADTRADF flag

ADTRBUF Flag (GTADTRB Register Compare Match (Up-Counting) A/D Conversion Start Request Flag)

This status flag indicates generation of a GTADTRB register compare match in up-counting.
[Setting condition]

- The GTCNT counter matches the GTADTRB register in up-counting

[Clearing conditions]

- 0 is written to the ADTRBUF flag

ADTRBDF Flag (GTADTRB Register Compare Match (Down-Counting) A/D Conversion Start Request Flag)

This status flag indicates generation of a GTADTRB register compare match in down-counting.
[Setting condition]

- The GTCNT counter matches the GTADTRB register in down-counting

[Clearing conditions]

- 0 is written to the ADTRBDF flag

ODF Flag (Output Stop Request Flag)

This flag monitors the output stop request for the group selected by the GTINTAD.GRP[1:0] bits.
Even if a request to disable output is de-asserted after the request is made, releasing the control of negation on the PWM pins has to wait until the end of the PWM cycle.

DTEF Flag (Dead Time Error Flag)

This flag indicates that the timer output toggle point after the automatic addition of dead time has exceeded the count period.
This flag returns to 0 when the timer output toggle point after the automatic addition of dead time is back within the period. This flag can only be read from. (Writing 0 to clear the flag is not allowed.)
When an interrupt by the DTEF flag is enabled (when the GTINTAD.GRPDTE bit is 1), the DTEF flag is output as detection of disabling of output to the POEG, and an GDTE interrupt is generated whenever the DTEF flag changes from 0 to 1.
[Setting condition]

- When a change point of the waveform after the automatic setting of dead time has exceeded the count period (in the following cases)
  Up-counting in triangle-wave mode: GTCCRA register – GTDVU register ≤ 0
  Down-counting in triangle-wave mode: GTCCRA register – GTDVD register < 0
  Up-counting in sawtooth-wave one-shot pulse mode:
  When GTCCRA register – GTDVU register < 0, or GTCCRA register + GTDVD register > GTPR register
  Down-counting in sawtooth-wave one-shot pulse mode:
  When GTCCRA register + GTDVU register > GTPR register, or GTCCRA register – GTDVD register < 0

[Clearing condition]

- The timer output toggle point after the automatic addition of dead time is within the count period.

OABHF Flag (Simultaneous High Output Flag)

This flag indicates that the simultaneous output of 1 both from the GTIOCnA and GTIOCnB pins.
When one of the GTIOCnA and GTIOCnB pins changes to 0, the flag changes to 0.
This flag can only be read from. (Writing 0 to clear the flag is not allowed.)
When the output stop detection by the OABHF flag is enabled (the GTINTAD.GRPABH bit is 1), the OABHF flag is output as the output stop detection to the POEG. The GPTW does not have an interrupt to indicate that outputs have been simultaneous driven to the high level. Use the interrupt function in the POEG if this is necessary.
[Setting condition]

- The GTIOCnA and GTIOCnB pins output 1 simultaneously while both the OAE and OBE bits in the GTIOR register are set to 1.

[Clearing conditions]

- The GTIOCnA and GTIOCnB pins output different values while both the OAE and OBE bits are set to 1.

- The GTIOCnA and GTIOCnB pins output 0 simultaneously while both the OAE and OBE bits are set to 1.
- At least one of the OAE and OBE bits is set to 0.

OABLF Flag (Simultaneous Low Output Flag)

This flag indicates that the simultaneous output of 0 both from the GTIOCnA and GTIOCnB pins.
When one of the GTIOCnA and GTIOCnB pins changes to 1, the flag changes to 0.
This flag can only be read from. (Writing 0 to clear the flag is not allowed.)
When the output stop detection by the OABLF flag is enabled (the GTINTAD.GRPABL bit is 1), the OABLF flag is output as the output stop detection to the POEG. The GPTW does not have an interrupt to indicate that outputs have been simultaneous driven to the high level. Use the interrupt function in the POEG if this is necessary.
[Setting condition]

- The GTIOCnA and GTIOCnB pins output 0 simultaneously while both the OAE and OBE bits in the GTIOR register are set to 1.

[Clearing conditions]

- The GTIOCnA and GTIOCnB pins output different values while both the OAE and OBE bits are set to 1.
- The GTIOCnA and GTIOCnB pins output 1 simultaneously while both the OAE and OBE bits are set to 1.
- At least one of the OAE and OBE bits is set to 0.

Comparison for the output value to generate the OABHF or OABLF flag uses the value before the compare match output (PWM output) is masked by the output negate function. Even during the output negate condition, compare match operation continues internally, where the OABHF or OABLF flag is updated based on the operation results.

PCF Flag (Cycle Count End Flag)

This status flag indicates the end of the cycle count operation.
[Setting condition]

- The GTPC.PCEN bit is 1 and the GTPC.PCNT counter is 1 at the end of cycle.
- The GTPC.PCEN bit is 1 and the GTPC.PCNT counter is 0 at the count clock.

[Clearing conditions]

- 0 is written to the PCF flag

24.2.17 General PWM Timer Buffer Enable Register (GTBER)

Address(es): GPTW0.GTBER 000C 2040h, GPTW1.GTBER 000C 2140h, GPTW2.GTBER 000C 2240h, GPTW3.GTBER 000C 2340h, GPTW4.GTBER 000C 2440h, GPTW5.GTBER 000C 2540h, GPTW6.GTBER 000C 2640h, GPTW7.GTBER 000C 2740h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	ADTDB	ADTTB[1:0]		—	ADTDA	ADTTA[1:0]		—	CCRSWT	PR[1:0]		CCRB[1:0]		CCRA[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DBRTSADB	DBRTEADB	DBRTSADA	DBRTEADA	DBRTSCB	DBRTECB	DBRTSCA	DBRTECA	—	—	—	—	BD[3]	BD[2]	BD[1]	BD[0]
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BD[0]	GTCCRA/GTCCRB Registers Buffer Operation Disable	0: Buffer operation is enabled 1: Buffer operation is disabled	R/W
b1	BD[1]	GTPR Register Buffer Operation Disable		R/W
b2	BD[2]	GTADTRA/GTADTRB Registers Buffer Operation Disable		R/W
b3	BD[3]	GTDVU/GTDVD Registers Buffer Operation Disable		R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	DBRTECA	GTCCRA Register Double Buffer Repeat Operation Enable	0: GTCCRA register double buffer repeat operation is disabled 1: GTCCRA register double buffer repeat operation is enabled	R/W
b9	DBRTSCA	GTCCRA Register Double Buffer Repeat Operation Period Select	0: Repeats the transfer from the intermediate buffer to the GTCCRA register with trough as the cycle. 1: Repeats the transfer from the intermediate buffer to the GTCCRA register with crest as the cycle.	R/W
b10	DBRTECB	GTCCRB Register Double Buffer Repeat Operation Enable	0: GTCCRB register double buffer repeat operation is disabled 1: GTCCRB register double buffer repeat operation is enabled	R/W
b11	DBRTSCB	GTCCRB Register Double Buffer Repeat Operation Period Select	0: Repeats the transfer from the intermediate buffer to the GTCCRB register with trough as the cycle. 1: Repeats the transfer from the intermediate buffer to the GTCCRB register with crest as the cycle.	R/W
b12	DBRTEADA	GTADTRA Register Double Buffer Repeat Operation Enable	0: GTADTRA register double buffer repeat operation is disabled 1: GTADTRA register double buffer repeat operation is enabled	R/W
b13	DBRTSADA	GTADTRA Register Double Buffer Repeat Operation Period Select	0: Repeats the transfer from the intermediate buffer to the GTADTRA register with trough as the cycle. 1: Repeats the transfer from the intermediate buffer to the GTADTRA register with crest as the cycle.	R/W
b14	DBRTEADB	GTADTRB Register Double Buffer Repeat Operation Enable	0: GTADTRB register double buffer repeat operation is disabled 1: GTADTRB register double buffer repeat operation is enabled	R/W
b15	DBRTSADB	GTADTRB Register Double Buffer Repeat Operation Period Select	0: Repeats the transfer from the intermediate buffer to the GTADTRB register with trough as the cycle. 1: Repeats the transfer from the intermediate buffer to the GTADTRB register with crest as the cycle.	R/W

Bit	Symbol	Bit Name	Description	R/W
b17, b16	CCRA[1:0]	GTCCRA Register Buffer Operation	b17b16 0 0: Buffer operation is not performed 0 1: Single buffer operation (GTCCRA register ⇔ GTCCRC register) 1 x: Double buffer operation (GTCCRA register ⇔ GTCCRC register ⇔ GTCCRD register)	R/W
b19, b18	CCRB[1:0]	GTCCRB Register Buffer Operation	b19b18 0 0: Buffer operation is not performed 0 1: Single buffer operation (GTCCRB register ⇔ GTCCRE register) 1 x: Double buffer operation (GTCCRB register ⇔ GTCCRE register ⇔ GTCCRF register)	R/W
b21, b20	PR[1:0]	GTPR Register Buffer Operation	b21b20 0 0: Buffer operation is not performed 0 1: Single buffer operation (GTPBR register ⇒ GTPR register) 1 x: Double buffer operation (GTPDBR register ⇒ GTPBR register ⇒ GTPR register)	R/W
b22	CCRSWT	GTCCRA and GTCCRB Registers Forcible Buffer Operation	Writing 1 to this bit forcibly performs buffer transfer of GTCCRA and GTCCRB registers. This bit automatically returns to 0 after the writing of 1. This bit is read as 0.	R/W
b23	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b25, b24	ADTTA[1:0]	GTADTRA Register Buffer Transfer Timing Select	• Triangle waves or in complementary PWM mode b25b24 0 0: No transfer 0 1: Transfer at crest 1 0: Transfer at trough 1 1: Transfer at both crest and trough • Sawtooth waves b25b24 0 0: No transfer Values other than 0 0: Transfer at underflow (during down-counting), overflow (during up-counting), or counter clearing	R/W
b26	ADTDA	GTADTRA Register Double Buffer Operation	0: Single buffer operation (GTADTBRA register ⇒ GTADTRA register) 1: Double buffer operation (GTADTDBRA register ⇒ GTADTBRA register ⇒ GTADTRA register)	R/W
b27	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b29, b28	ADTTB[1:0]	GTADTRB Register Buffer Transfer Timing Select	• Triangle waves or in complementary PWM mode b29b28 0 0: No transfer 0 1: Transfer at crest 1 0: Transfer at trough 1 1: Transfer at both crest and trough • Sawtooth waves b29b28 0 0: No transfer Values other than 0 0: Transfer at underflow (during down-counting), overflow (during up-counting), or counter clearing	R/W
b30	ADTDB	GTADTRB Register Double Buffer Operation	0: Single buffer operation (GTADTBRB register ⇒ GTADTRB register) 1: Double buffer operation (GTADTDBRB register ⇒ GTADTBRB register ⇒ GTADTRB register)	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

The GTBER register makes settings for buffer operation.

Set the GTBER register except the BD[3:0] bits while the GTCNT counter is stopped.

BD[0] Bit (GTCCRA/GTCCRB Registers Buffer Operation Disable)

This bit disables buffer operation using the GTCCRA, GTCCRC, and GTCCRD registers together and buffer operation using the GTCCRB, GTCCRE, and GTCCRF registers together.

Even though the BD[0] bit is set to 0, buffer operation in the GTCCRB register is not performed if the GTDTCR.TDE bit is set to 1 in sawtooth-wave one-shot pulse mode or triangle-wave PWM mode. Instead, the compare match value for negative-phase waveform with dead time is set automatically.

In complementary PWM mode, it is valid only for the buffer operation of GTCCRC register and GTCCRE register. The buffer operation of GTCCRA registers cannot be disabled. Buffer operation of GTCCRE register and GTCCRF register is enabled/disabled by GTBER2.CP3DB bit. No buffer transfer to GTCCRB is performed in complementary PWM mode.

A value for the BD[0] bit in the channel related to the position of the bit written with 1 by the GTSECSR register can be set when 1 is written to the SBDCE or SBDCD bit in the GTSECR register.

When the DBRTECm (m = A, B) bit is 1, setting the BD[0] bit to 1 while the mode of operation is sawtooth-wave one-shot pulse mode, triangle-wave PWM mode 3, or double buffer operation with triangle waves results in transfer from the intermediate buffer to the GTCCRm register.

BD[1] Bit (GTPR Register Buffer Operation Disable)

This bit disables buffer operation using the GTPR, GTPBR, and GTPDBR registers together.

A value for the BD[1] bit in the channel related to the position of the bit written with 1 by the GTSECSR register can be set when 1 is written to the SBDPE or SBDPD bit in the GTSECR register.

In complementary PWM mode, the slave channel is also controlled by setting the BD[1] bit of the master channel.

BD[2] Bit (GTADTRA/GTADTRB Registers Buffer Operation Disable)

This bit disables buffer operation using the GTADTRA, GTADTBRA, and GTADTDBRA registers together and buffer operation using the GTADTRB, GTADTBRB, and GTADTDBRB registers together.

The setting is invalid during the event count operation, and the buffer operation using the GTADTRA and GTADTRB registers is not performed.

A value for the BD[2] bit in the channel related to the position of the bit written with 1 by the GTSECSR register can be set when 1 is written to the SBDAE or SBDAD bit in the GTSECR register.

When the DBRTECm (m = A, B) bit is 1, triangle-wave is used, the GTADTRm register is operated in a double buffer, and the buffer transfer timing is set to both crest and trough (GTBER.ADTTm[1:0] bits are 11b), Even if the BD[2] bit is set to 1, the buffer operation from the intermediate buffer to the GTADTRm register is performed.

BD[3] Bit (GTDVU/GTDVD Registers Buffer Operation Disable)

In sawtooth-wave PWM mode 1, sawtooth-wave one-shot pulse mode or triangle-wave PWM mode, this bit disables buffer operation using the GTDVU and GTDBU registers together and buffer operation using the GTDVD and GTDBD registers together.

Even though the BD[3] bit is set to 0, buffer operation in the GTDVD register is not performed if the GTDTCR.TDFER bit is set to 1. Instead, the value in the GTDVU register is set automatically.

In sawtooth-wave PWM mode 2 or complementary PWM mode, this bit is invalid and GTDVU and GTDVD registers don’t perform buffer operation.

The setting is invalid during the event count operation, and the buffer operation using the GTDVU and GTDVD registers is not performed.

A value for the BD[3] bit in the channel related to the position of the bit written with 1 by the GTSECSR register can be set when 1 is written to the SBDDE or SBDDD bit in the GTSECR register.

DBRTECm Bit (GTCCRm Register Double Buffer Repeat Operation Enable) (m = A, B)

This setting enables the operation to repeat a transfer to the GTCCRm register from the intermediate buffer by cycle during the buffer transfer disable period when performing the double buffer operation using the GTCCRm register. It is

valid in sawtooth-wave one-shot pulse mode, triangle-wave PWM mode 3, or triangle wave PWM mode 2 that double buffers the GTCCRm register (GTBER.CCRm[1] bit is 1).
The disabling period of buffer transfer indicates the period that buffer transfer is stopped by setting of the BD[0] bit (CPU writing or simultaneous buffer operation control by the GTSECSR register) and the period for buffer transfer extended skipping (except the case for skipping by counting both crests and troughs) by the GTEITLB register.
When the DBRTECm bit is 1, writing by the CPU to the GTCCRm register sets the same value in temporary register x (x = C, E). The value of the GTCCRx (x = C, E) register is also transferred to the temporary register x (x = C, E) by forcible buffer transfer.

DBRTSCm Bit (GTCCRm Register Double Buffer Repeat Operation Period Select) (m = A, B)

When GTCCRm double buffer repeat operation is permitted, select the period for which the repeat operation is to be performed. This is valid for triangular wave PWM mode 2.
When changing the prohibition/permission of buffer transfer during the counting operation of the GTCNT counter, change it in the same period as the period set by the DBRTSCm bit.

DBRTEADm Bit (GTADTRm Register Double Buffer Repeat Operation Enable) (m = A, B)

When operating the GTADTRm register in a double buffer operation, the operation of repeating the transfer from the intermediate buffer to the GTADTRm register in periodic units is permitted during the period when buffer transfer is prohibited.
Using the triangular wave and complementary PWM mode, the GTADTRm register is double-buffered (GTBER.ADTDm bit is 1), and the buffer transfer timing is both crest and trough (GTBER.ADTTm[1:0] bits are 11b). It is effective in the case.
The buffer transfer prohibition period is set for the BD[2] bits (CPU write or buffer operation control by the GTSECSR register), buffer transfer expansion thinning by the GTEITLB register (except when counting both crest and trough with a triangular wave and thinning out). It corresponds to any of the buffer transfer decimation by the A/D conversion start request compare match decimation selected in the GTADCMSS register.
When the ADTDm bit is 1 and the DBRTEADm bit is 1, when CPU writing is performed to the GTADTRm register, the same value is also written to the temporary register ADm (m = A, B).

DBRTSADm Bit (GTADTRA Register Double Buffer Repeat Operation Period Select) (m = A, B)

When GTADTRm double buffer repeat operation is permitted, select the period for which repeat operation is to be performed.
When changing disable or enable during the counting operation of the GTCNT counter, change it in the same cycle as the cycle set in the DBRTSADm bit.

CCRA[1:0] Bits (GTCCRA Register Buffer Operation)

These bits set buffer operation with the GTCCRA, GTCCRC, and GTCCRD registers combined. When buffer operation is restricted by the operating mode set in the GTCR register, the GTCR register setting is given priority.*1

CCRB[1:0] Bits (GTCCRB Register Buffer Operation)

These bits set buffer operation with the GTCCRB, GTCCRE, and GTCCRF registers combined. When buffer operation is restricted by the operating mode set in the GTCR register, the GTCR register setting is given priority.*1

PR[1:0] Bits (GTPR Register Buffer Operation)

These bits set buffer operation with the GTPR, GTPBR, and GTPDBR registers combined.

CCRSWT Bit (GTCCRA and GTCCRB Registers Forcible Buffer Operation)

Writing 1 to the CCRSWT bit forcibly performs buffer transfer of the GTCCRA and GTCCRB registers. This bit automatically returns to 0 after the writing of 1. This bit is read as 0.

This bit is valid only when counting is stopped with compare match operation specified.

ADTTA[1:0] Bits (GTADTRA Register Buffer Transfer Timing Select)

These bits set the transfer timing for buffer operation of the GTADTRA, GTADTBRA, and GTADTDBRA registers. The setting is invalid during the event count operation.

ADTDA Bit (GTADTRA Register Double Buffer Operation)

These bits set buffer operation with the GTADTRA, GTADTBRA, and GTADTDBRA registers combined. The setting is invalid during the event count operation.

ADTTB[1:0] Bits (GTADTRB Register Buffer Transfer Timing Select)

These bits set the transfer timing for buffer operation of the GTADTRB, GTADTBRB, and GTADTDBRB registers. The setting is invalid during the event count operation.

ADTDB Bit (GTADTRB Register Double Buffer Operation)

These bits set buffer operation with the GTADTRA, GTADTBRA, and GTADTDBRA registers combined. The setting is invalid during the event count operation.

Note 1. The buffer operation mode is fixed in sawtooth-wave one-shot pulse mode or triangle-wave PWM mode 3 (64-bit transfer at trough).

24.2.18 General PWM Timer Interrupt and A/D Conversion Start Request Skipping Setting Register (GTITC)

Address(es): GPTW0.GTITC 000C 2044h, GPTW1.GTITC 000C 2144h, GPTW2.GTITC 000C 2244h, GPTW3.GTITC 000C 2344h, GPTW4.GTITC 000C 2444h, GPTW5.GTITC 000C 2544h, GPTW6.GTITC 000C 2644h, GPTW7.GTITC 000C 2744h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	ADTBL	—	ADTAL	—	IVTT[2:0]			IVTC[1:0]		ITLF	ITLE	ITLD	ITLC	ITLB	ITLA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ITLA	GTCCRA Register Compare Match/Input Capture Interrupt Link	0: Not linked with GTCIV/GTCIU interrupt skipping function. 1: Linked with GTCIV/GTCIU interrupt skipping function.	R/W
b1	ITLB	GTCCRB Register Compare Match/Input Capture Interrupt Link	0: Not linked with GTCIV/GTCIU interrupt skipping function. 1: Linked with GTCIV/GTCIU interrupt skipping function.	R/W
b2	ITLC	GTCCRC Register Compare Match Interrupt Link	0: Not linked with GTCIV/GTCIU interrupt skipping function. 1: Linked with GTCIV/GTCIU interrupt skipping function.	R/W
b3	ITLD	GTCCRD Register Compare Match Interrupt Link	0: Not linked with GTCIV/GTCIU interrupt skipping function. 1: Linked with GTCIV/GTCIU interrupt skipping function.	R/W
b4	ITLE	GTCCRE Register Compare Match Interrupt Link	0: Not linked with GTCIV/GTCIU interrupt skipping function. 1: Linked with GTCIV/GTCIU interrupt skipping function.	R/W
b5	ITLF	GTCCRF Register Compare Match Interrupt Link	0: Not linked with GTCIV/GTCIU interrupt skipping function. 1: Linked with GTCIV/GTCIU interrupt skipping function.	R/W
b7, b6	IVTC[1:0]	GTCIV/GTCIU Interrupt Skipping Function Select	b7 b6 0 0: Skipping is not performed 0 1: Both overflow and underflow for sawtooth waves and crest for triangle waves are counted and skipped. 1 0: Both overflow and underflow for sawtooth waves and trough for triangle waves are counted and skipped. 1 1: Both overflow and underflow for sawtooth waves and both crest and trough for triangle waves are counted and skipped	R/W
b10 to b8	IVTT[2:0]	GTCIV/GTCIU Interrupt Skipping Count Select	b10 b8 0 0 0: Skipping is not performed 0 0 1: Skipping count of 1 0 1 0: Skipping count of 2 0 1 1: Skipping count of 3 1 0 0: Skipping count of 4 1 0 1: Skipping count of 5 1 1 0: Skipping count of 6 1 1 1: Skipping count of 7	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b12	ADTAL	GTADTRA Register A/D Conversion Start Request Link	0: Not linked with GTCIV/GTCIU interrupt skipping function. 1: Linked with GTCIV/GTCIU interrupt skipping function.	R/W
b13	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b14	ADTBL	GTADTRB Register A/D Conversion Start Request Link	0: Not linked with GTCIV/GTCIU interrupt skipping function. 1: Linked with GTCIV/GTCIU interrupt skipping function.	R/W
b31 to b15	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

GTITC sets the skipping function for the GTCNT counter overflow (GTPR compare match) interrupt (GTCIV)/ underflow interrupt (GTCIU) and also sets whether to link the other interrupts and A/D conversion start requests with the GTCIV/GTCIU interrupt skipping function. Note that detection of disabling of output to the POEG and dead-time error interrupts cannot be linked with the GTCIV or GTCIU interrupt skipping function. Additionally, if the interrupt skipping function is performed, the change in the status flag is also skipped.
The setting is invalid during the event count operation. The setting is operated independently from the extended interrupt skipping by the GTEITC register.

ITLA Bit (GTCCRA Register Compare Match/Input Capture Interrupt Link)

This bit specifies whether to link the GTCCRA compare match/input capture interrupt (GTCIA) with the GTCIV/ GTCIU interrupt skipping function.

ITLB Bit (GTCCRB Register Compare Match/Input Capture Interrupt Link)

This bit specifies whether to link the GTCCRB compare match/input capture interrupt (GTCIB) with the GTCIV/GTCIU interrupt skipping function.

ITLC Bit (GTCCRC Register Compare Match Interrupt Link)

This bit specifies whether to link the GTCCRC compare match interrupt (GTCIC) with the GTCIV/GTCIU interrupt skipping function.

ITLD Bit (GTCCRD Register Compare Match Interrupt Link)

This bit specifies whether to link the GTCCRD compare match interrupt (GTCID) with the GTCIV/GTCIU interrupt skipping function.

ITLE Bit (GTCCRE Register Compare Match Interrupt Link)

This bit specifies whether to link the GTCCRE compare match interrupt (GTCIE) with the GTCIV/GTCIU interrupt skipping function.

ITLF Bit (GTCCRF Register Compare Match Interrupt Link)

This bit specifies whether to link the GTCCRF compare match interrupt (GTCIF) with the GTCIV/GTCIU interrupt skipping function.

IVTC[1:0] Bits (GTCIV/GTCIU Interrupt Skipping Function Select)

These bits set the skipping function for the GTPR compare match (GTCNT counter overflow) interrupt (GTCIV)/ GTCNT counter underflow interrupt (GTCIU).

IVTT[2:0] Bits (GTCIV/GTCIU Interrupt Skipping Count Select)

These bits set the skipping count for the GTPR compare match (GTCNT counter overflow) interrupt (GTCIV)/GTCNT counter underflow interrupt (GTCIU).
When modifying the IVTT[2:0] bits, first set the IVTC[1:0] bits to 00b.

ADTAL Bit (GTADTRA Register A/D Conversion Start Request Link)

This bit specifies whether to link the A/D conversion start request, which is generated in response to a compare match with the GTCNT counter and the GTADTRA register, with the GTCIV/GTCIU interrupt skipping function.

ADTBL Bit (GTADTRB Register A/D Conversion Start Request Link)

This bit specifies whether to link the A/D conversion start request, which is generated in response to a compare match with the GTCNT counter and the GTADTRB register, with the GTCIV/GTCIU interrupt skipping function.

24.2.19 General PWM Timer Counter (GTCNT)

Address(es): GPTW0.GTCNT 000C 2048h, GPTW1.GTCNT 000C 2148h, GPTW2.GTCNT 000C 2248h, GPTW3.GTCNT 000C 2348h, GPTW4.GTCNT 000C 2448h, GPTW5.GTCNT 000C 2548h, GPTW6.GTCNT 000C 2648h, GPTW7.GTCNT 000C 2748h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The GTCNT counter is a 32-bit readable/writable counter. There is one GTCNT counter for each channel. Writing is available only when counting is stopped, and thus writing is disabled while counting is in progress (when CST = 1). The upper 16 bits are reserved in the products with 48 Kbytes of RAM, and these bits are read as 0.
Access in 8-bit or 16-bit units to the GTCNT counter is prohibited, and it should be accessed in 32-bit units.
Set the range of the GTCNT counter within the range of 0 ≤ GTCNT counter ≤ GTPR register.

24.2.20 General PWM Timer Compare Capture Register m (GTCCRm) (m = A to F)

Address(es): GPTW0.GTCCRA 000C 204Ch, GPTW1.GTCCRA 000C 214Ch, GPTW2.GTCCRA 000C 224Ch, GPTW3.GTCCRA 000C 234Ch, GPTW4.GTCCRA 000C 244Ch, GPTW5.GTCCRA 000C 254Ch, GPTW6.GTCCRA 000C 264Ch, GPTW7.GTCCRA 000C 274Ch,
GPTW0.GTCCRB 000C 2050h, GPTW1.GTCCRB 000C 2150h, GPTW2.GTCCRB 000C 2250h, GPTW3.GTCCRB 000C 2350h, GPTW4.GTCCRB 000C 2450h, GPTW5.GTCCRB 000C 2550h, GPTW6.GTCCRB 000C 2650h, GPTW7.GTCCRB 000C 2750h,
GPTW0.GTCCRC 000C 2054h, GPTW1.GTCCRC 000C 2154h, GPTW2.GTCCRC 000C 2254h, GPTW3.GTCCRC 000C 2354h, GPTW4.GTCCRC 000C 2454h, GPTW5.GTCCRC 000C 2554h, GPTW6.GTCCRC 000C 2654h, GPTW7.GTCCRC 000C 2754h,
GPTW0.GTCCRE 000C 2058h, GPTW1.GTCCRE 000C 2158h, GPTW2.GTCCRE 000C 2258h, GPTW3.GTCCRE 000C 2358h, GPTW4.GTCCRE 000C 2458h, GPTW5.GTCCRE 000C 2558h, GPTW6.GTCCRE 000C 2658h, GPTW7.GTCCRE 000C 2758h,
GPTW0.GTCCRD 000C 205Ch, GPTW1.GTCCRD 000C 215Ch, GPTW2.GTCCRD 000C 225Ch, GPTW3.GTCCRD 000C 235Ch, GPTW4.GTCCRD 000C 245Ch, GPTW5.GTCCRD 000C 255Ch, GPTW6.GTCCRD 000C 265Ch, GPTW7.GTCCRD 000C 275Ch,
GPTW0.GTCCRF 000C 2060h, GPTW1.GTCCRF 000C 2160h, GPTW2.GTCCRF 000C 2260h, GPTW3.GTCCRF 000C 2360h, GPTW4.GTCCRF 000C 2460h, GPTW5.GTCCRF 000C 2560h, GPTW6.GTCCRF 000C 2660h, GPTW7.GTCCRF 000C 2760h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Note 1. The value after reset is 1 for products with 64 Kbytes of RAM and 0 for products with 48 Kbytes of RAM.

The GTCCRm register is a 32-bit readable/writable register. The upper 16 bits are reserved in the products with 48 Kbytes of RAM, and these bits are read as 0.
Access in 8-bit or 16-bit units to the GTCCRm register is prohibited, and it should be accessed in 32-bit units.
The GTCCRA and GTCCRB registers are registers used for both output compare and input capture.
The GTCCRC and GTCCRE registers are compare match registers, and can also function as buffer registers for the GTCCRA and GTCCRB registers.
The GTCCRD and GTCCRF registers are compare match registers, and can also function as buffer registers for the GTCCRC and GTCCRE registers (double buffer registers for the GTCCRA and GTCCRB registers).

24.2.21 General PWM Timer Period Setting Register (GTPR)

Address(es): GPTW0.GTPR 000C 2064h, GPTW1.GTPR 000C 2164h, GPTW2.GTPR 000C 2264h, GPTW3.GTPR 000C 2364h, GPTW4.GTPR 000C 2464h, GPTW5.GTPR 000C 2564h, GPTW6.GTPR 000C 2664h, GPTW7.GTPR 000C 2764h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Note 1. The value after reset is 1 for products with 64 Kbytes of RAM and 0 for products with 48 Kbytes of RAM.

The GTPR register is a 32-bit readable/writable register that sets the maximum count value of the GTCNT counter. The upper 16 bits are reserved in the products with 48 Kbytes of RAM, and these bits are read as 0.
Access in 8-bit or 16-bit units to the GTPR register is prohibited, and it should be accessed in 32-bit units.
The setting is invalid in the sawtooth-wave PWM mode 2. In sawtooth-wave mode except the sawtooth-wave PWM mode 2, the value of (GTPR register + 1) is the count period. In triangle-wave mode or complementary PWM mode, the value of (GTPR register value × 2) is the count period.
When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.

24.2.22 General PWM Timer Period Setting Buffer Register (GTPBR)

Address(es): GPTW0.GTPBR 000C 2068h, GPTW1.GTPBR 000C 2168h, GPTW2.GTPBR 000C 2268h, GPTW3.GTPBR 000C 2368h, GPTW4.GTPBR 000C 2468h, GPTW5.GTPBR 000C 2568h, GPTW6.GTPBR 000C 2668h, GPTW7.GTPBR 000C 2768h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Note 1. The value after reset is 1 for products with 64 Kbytes of RAM and 0 for products with 48 Kbytes of RAM.

The GTPBR register is a 32-bit readable/writable register that functions as a buffer register for the GTPR register. The upper 16 bits are reserved in the products with 48 Kbytes of RAM, and these bits are read as 0.
Access in 8-bit or 16-bit units to the GTPBR register is prohibited, and it should be accessed in 32-bit units.
The setting is invalid in the sawtooth-wave PWM mode 2. When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.

24.2.23 General PWM Timer Period Setting Double-Buffer Register (GTPDBR)

Address(es): GPTW0.GTPDBR 000C 206Ch, GPTW1.GTPDBR 000C 216Ch, GPTW2.GTPDBR 000C 226Ch, GPTW3.GTPDBR 000C 236Ch, GPTW4.GTPDBR 000C 246Ch, GPTW5.GTPDBR 000C 256Ch, GPTW6.GTPDBR 000C 266Ch, GPTW7.GTPDBR 000C 276Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Note 1. The value after reset is 1 for products with 64 Kbytes of RAM and 0 for products with 48 Kbytes of RAM.

The GTPDBR register is a 32-bit readable/writable register that functions as a buffer register for the GTPBR register (a double buffer register for the GTPR register). The upper 16 bits are reserved in the products with 48 Kbytes of RAM, and these bits are read as 0.
Access in 8-bit or 16-bit units to the GTPDBR register is prohibited, and it should be accessed in 32-bit units.
The setting is invalid in the sawtooth-wave PWM mode 2. When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.

24.2.24 A/D Conversion Start Request Timing Register m (GTADTRm) (m = A, B)

Address(es): GPTW0.GTADTRA 000C 2070h, GPTW1.GTADTRA 000C 2170h, GPTW2.GTADTRA 000C 2270h, GPTW3.GTADTRA 000C 2370h, GPTW4.GTADTRA 000C 2470h, GPTW5.GTADTRA 000C 2570h, GPTW6.GTADTRA 000C 2670h, GPTW7.GTADTRA 000C 2770h,
GPTW0.GTADTRB 000C 207Ch, GPTW1.GTADTRB 000C 217Ch, GPTW2.GTADTRB 000C 227Ch, GPTW3.GTADTRB 000C 237Ch, GPTW4.GTADTRB 000C 247Ch, GPTW5.GTADTRB 000C 257Ch, GPTW6.GTADTRB 000C 267Ch, GPTW7.GTADTRB 000C 277Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Note 1. The value after reset is 1 for products with 64 Kbytes of RAM and 0 for products with 48 Kbytes of RAM.

The GTADTRm register is 32-bit readable/writable register that set the timing of A/D conversion start request generation. The upper 16 bits are reserved in the products with 48 Kbytes of RAM, and these bits are read as 0.
Access in 8-bit or 16-bit units to the GTADTRm register is prohibited, and it should be accessed in 32-bit units.
When the GTADTRm register value matches the GTCNT counter value, an A/D conversion start request is generated. In complementary PWM mode, A/D conversion start request is generated when the GTCNT counter of the master channel matches this register.

24.2.25 A/D Conversion Start Request Timing Buffer Register m (GTADTBRm) (m = A, B)

Address(es): GPTW0.GTADTBRA 000C 2074h, GPTW1.GTADTBRA 000C 2174h, GPTW2.GTADTBRA 000C 2274h, GPTW3.GTADTBRA 000C 2374h, GPTW4.GTADTBRA 000C 2474h, GPTW5.GTADTBRA 000C 2574h, GPTW6.GTADTBRA 000C 2674h, GPTW7.GTADTBRA 000C 2774h, GPTW0.GTADTBRB 000C 2080h, GPTW1.GTADTBRB 000C 2180h, GPTW2.GTADTBRB 000C 2280h, GPTW3.GTADTBRB 000C 2380h, GPTW4.GTADTBRB 000C 2480h, GPTW5.GTADTBRB 000C 2580h, GPTW6.GTADTBRB 000C 2680h, GPTW7.GTADTBRB 000C 2780h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Note 1. The value after reset is 1 for products with 64 Kbytes of RAM and 0 for products with 48 Kbytes of RAM.

The GTADTBRm register is 32-bit readable/writable register that function as buffer register for the GTADTRm register. The upper 16 bits are reserved in the products with 48 Kbytes of RAM, and these bits are read as 0.
Access in 8-bit or 16-bit units to the GTADTBRm register is prohibited, and it should be accessed in 32-bit units.

24.2.26 A/D Conversion Start Request Timing Double-Buffer Register m (GTADTDBRm) (m = A, B)

Address(es): GPTW0.GTADTDBRA 000C 2078h, GPTW1.GTADTDBRA 000C 2178h, GPTW2.GTADTDBRA 000C 2278h, GPTW3.GTADTDBRA 000C 2378h, GPTW4.GTADTDBRA 000C 2478h, GPTW5.GTADTDBRA 000C 2578h, GPTW6.GTADTDBRA 000C 2678h, GPTW7.GTADTDBRA 000C 2778h, GPTW0.GTADTDBRB 000C 2084h, GPTW1.GTADTDBRB 000C 2184h, GPTW2.GTADTDBRB 000C 2284h, GPTW3.GTADTDBRB 000C 2384h, GPTW4.GTADTDBRB 000C 2484h, GPTW5.GTADTDBRB 000C 2584h, GPTW6.GTADTDBRB 000C 2684h, GPTW7.GTADTDBRB 000C 2784h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Note 1. The value after reset is 1 for products with 64 Kbytes of RAM and 0 for products with 48 Kbytes of RAM.

The GTADTDBRm register is 32-bit readable/writable register that function as buffer register for the GTADTBRm register (double buffer register for the GTADTRm register). The upper 16 bits are reserved in the products with 48 Kbytes of RAM, and these bits are read as 0.
Access in 8-bit or 16-bit units to the GTADTDBRm register is prohibited, and it should be accessed in 32-bit units.

24.2.27 General PWM Timer Dead Time Control Register (GTDTCR)

Address(es): GPTW0.GTDTCR 000C 2088h, GPTW1.GTDTCR 000C 2188h, GPTW2.GTDTCR 000C 2288h, GPTW3.GTDTCR 000C 2388h, GPTW4.GTDTCR 000C 2488h, GPTW5.GTDTCR 000C 2588h, GPTW6.GTDTCR 000C 2688h, GPTW7.GTDTCR 000C 2788h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	TDFER	—	—	TDBDE	TDBUE	—	—	—	TDE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TDE	Negative-Phase Waveform Setting	0: GTCCRB register is set without using GTDVU and GTDVD registers. 1: GTDVU and GTDVD registers are used to set the compare match value for negative-phase waveform with dead time automatically in GTCCRB register.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	TDBUE	GTDVU Register Buffer Operation Enable	0: GTDVU register buffer operation is disabled 1: GTDVU register buffer operation is enabled	R/W
b5	TDBDE	GTDVD Register Buffer Operation Enable	0: GTDVD register buffer operation is disabled 1: GTDVD register buffer operation is enabled	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	TDFER	GTDVD Register Setting	0: GTDVU and GTDVD registers are set separately. 1: The value written to GTDVU register is automatically set to GTDVD register.	R/W
b31 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTDTCR register enables automatic setting of a compare match value for negative-phase waveform with dead time. The setting is invalid during the event count operation.

TDE Bit (Negative-Phase Waveform Setting)

This bit sets whether to use the GTDVU and GTDVD registers. When the GTDVU and GTDVD registers are used, the compare match value for a negative-phase waveform with dead time that was obtained by the compare match value of a positive-phase waveform (GTCCRA register) and the dead time value (GTDVU and GTDVD registers) is automatically set in the GTCCRB register.

The TDE bit setting is ignored in sawtooth-wave PWM mode, and the GTCCRB register is not automatic setting.

The automatically set GTCCRB register value has the following upper and lower limit values.

- Triangle wave PWM mode
  Upper limit value: the value set in the GTPR register – 1
  Lower limit value: 0000 0001h in up-counting, 0000 0000h in down-counting
- Sawtooth-wave one-shot pulse mode
  Upper limit value: the value set in the GTPR register
  Lower limit value: 0000 0000h

If the obtained GTCCRB register value is not within the range between the upper and lower limits, the upper or lower limit is set in the GTCCRB register, and the GTST.DTEF flag becomes 1. However, if the obtained GTCCRB register value exceeds the upper limit in triangle-wave PWM mode, the DTEF flag becomes 0.

TDBUE Bit (GTDVU Register Buffer Operation Enable)

This bit enables buffer operation with the GTDVU and GTDBU registers combined.
The timing of buffer transfer is at troughs in triangle-wave mode, and at overflows or underflows in sawtooth-wave mode.

TDBDE Bit (GTDVD Register Buffer Operation Enable)

This bit enables buffer operation with the GTDVD and GTDBD registers combined.
The buffer transfer timing is the trough in triangle-wave mode, and at an overflow or underflow in sawtooth-wave mode.
When this bit and the TDFER bit are set to 1 simultaneously, the TDFER bit setting is given priority.

TDFER Bit (GTDVD Register Setting)

This bit sets whether or not the value written to the GTDVU register is also set to the GTDVD register automatically.

24.2.28 General PWM Timer Dead Time Value Register m (GTDVm) (m = U, D)

Address(es): GPTW0.GTDVU 000C 208Ch, GPTW1.GTDVU 000C 218Ch, GPTW2.GTDVU 000C 228Ch, GPTW3.GTDVU 000C 238Ch, GPTW4.GTDVU 000C 248Ch, GPTW5.GTDVU 000C 258Ch, GPTW6.GTDVU 000C 268Ch, GPTW7.GTDVU 000C 278Ch, GPTW0.GTDVD 000C 2090h, GPTW1.GTDVD 000C 2190h, GPTW2.GTDVD 000C 2290h, GPTW3.GTDVD 000C 2390h, GPTW4.GTDVD 000C 2490h, GPTW5.GTDVD 000C 2590h, GPTW6.GTDVD 000C 2690h, GPTW7.GTDVD 000C 2790h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Note 1. The value after reset is 1 for products with 64 Kbytes of RAM and 0 for products with 48 Kbytes of RAM.

The GTDVm register is 32-bit readable/writable register that set the dead time for generating PWM waveforms with dead time. The upper 16 bits are reserved in the products with 48 Kbytes of RAM, and these bits are read as 0.
Access in 8-bit or 16-bit units to the GTDVm register is prohibited, and it should be accessed in 32-bit units.
The GTDVU register is used for up-counting. The GTDVD register is used for down-counting.
The setting is invalid in sawtooth-wave PWM mode 2.
In complementary PWM mode, the GTDVD register is invalid and the GTDVU register is used as the dead time value during both up-counting and down-counting. No matter which GTDVU register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.
In Triangle-waves, The GTDVU register is used for up-counting. The GTDVD register is used for down-counting. In the case of a sawtooth-waves, the GTDVU register controls the front dead time and the GTDVD register controls the rear dead time, regardless of whether the count is up or down.
The setting which the value for the GTDVm register is equal to or larger than the value for the GTPR register is prohibited.

- GTDVU > 0
- GTDVU < GTPR/2
- GTDVU + GTPR ≤ FFFF FFFFh

In addition, when using the automatic dead time setting function, do not set a value that makes a change point of the waveform exceeding the count period. The change point of the negative-phase waveform, which is automatically

calculated, is obtained by reading the GTCCRB register. When using the GTDVm register, writing to the GTCCRB register is prohibited. When setting this register to 0, waveforms without dead time are output. When the GTDTCR.TDFER bit is 1, the writing to the GTDVD register has no effect. At this time, when the GTDVD register is read, the value for the GTDVU register is read.

24.2.29 General PWM Timer Dead Time Value Buffer Register m (GTDBm) (m = U, D)

Address(es): GPTW0.GTDBU 000C 2094h, GPTW1.GTDBU 000C 2194h, GPTW2.GTDBU 000C 2294h, GPTW3.GTDBU 000C 2394h, GPTW4.GTDBU 000C 2494h, GPTW5.GTDBU 000C 2594h, GPTW6.GTDBU 000C 2694h, GPTW7.GTDBU 000C 2794h,
GPTW0.GTDBD 000C 2098h, GPTW1.GTDBD 000C 2198h, GPTW2.GTDBD 000C 2298h, GPTW3.GTDBD 000C 2398h, GPTW4.GTDBD 000C 2498h, GPTW5.GTDBD 000C 2598h, GPTW6.GTDBD 000C 2698h, GPTW7.GTDBD 000C 2798h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Note 1. The value after reset is 1 for products with 64 Kbytes of RAM and 0 for products with 48 Kbytes of RAM.

The GTDBm register is 32-bit readable/writable register that function as buffer register for the GTDVm register. The upper 16 bits are reserved in the products with 48 Kbytes of RAM, and these bits are read as 0.
Access in 8-bit or 16-bit units to the GTDBm register is prohibited, and it should be accessed in 32-bit units.

24.2.30 General PWM Timer Output Protection Function Status Register (GTSOS)

Address(es): GPTW0.GTSOS 000C 209Ch, GPTW1.GTSOS 000C 219Ch, GPTW2.GTSOS 000C 229Ch, GPTW3.GTSOS 000C 239Ch, GPTW4.GTSOS 000C 249Ch, GPTW5.GTSOS 000C 259Ch, GPTW6.GTSOS 000C 269Ch, GPTW7.GTSOS 000C 279Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SOS[1:0]	
Value after reset:	0	0	0	0	0	0	x	x	0	0	0	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b1, b0	SOS[1:0]	Output Protection Function Status	b1 b0 0 0: Normal operation 0 1: Protected state (GTCCRA = 0 is set during transfer at trough or crest) 1 0: Protected state (GTCCRA ≥ GTPR is set during transfer at trough) 1 1: Protected state (GTCCRA ≥ GTPR is set during transfer at crest)	R
b7 to b2	—	Reserved	These bits are read as 0.	R
b9, b8	—	Reserved	The read value is undefined.	R
b31 to b10	—	Reserved	These bits are read as 0.	R

The GTSOS register is a status register that indicates the status of the output protection function. The output protection function is enabled only when the dead time is automatically set (the GTDTCR.TDE bit is 1) in triangle-wave mode.

SOS[1:0] Bits (Output Protection Function Status)

This bit indicates the status of the output protection function in triangle-wave PWM mode. For details of the output protection function, see section 24.8.4, Output Protection Function for GTIOCnm Pin Output (n = 0 to 7; m = A, B).

24.2.31 General PWM Timer Output Protection Function Temporary Release Register (GTSOTR)

Address(es): GPTW0.GTSOTR 000C 20A0h, GPTW1.GTSOTR 000C 21A0h, GPTW2.GTSOTR 000C 22A0h, GPTW3.GTSOTR 000C 23A0h, GPTW4.GTSOTR 000C 24A0h, GPTW5.GTSOTR 000C 25A0h, GPTW6.GTSOTR 000C 26A0h, GPTW7.GTSOTR 000C 27A0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SOTR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SOTR	Output Protection Function Temporary Release	0: Protected state is not released 1: Protected state is released	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTSOTR register temporarily releases the protected state of GTIOCnB pin output (n = 0 to 7) when output protection has been set.

The protected state can be released only for the case of the GTSOS.SOS[1:0] bits are 10b (protected state in which GTCCRA register ≥ GTPR register has occurred during transfer at trough). The protected state cannot be released for any other case.

SOTR Bit (Output Protection Function Temporary Release)

This bit sets whether to temporarily release the protected state of the GTIOCnB pin output in an output protected state. After the SOTR bit has been set to 1, the output protection function is canceled from the first trough. After the SOTR bit has been set to 0, output protection is resumed from the first trough.

24.2.32 General PWM Timer A/D Conversion Start Request Signal Monitoring Register (GTADSMR)

Address(es): GPTW0.GTADSMR 000C 20A4h, GPTW1.GTADSMR 000C 21A4h, GPTW2.GTADSMR 000C 22A4h, GPTW3.GTADSMR 000C 23A4h, GPTW4.GTADSMR 000C 24A4h, GPTW5.GTADSMR 000C 25A4h, GPTW6.GTADSMR 000C 26A4h, GPTW7.GTADSMR 000C 27A4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	ADSMEN1	—	—	—	—	—	—	ADSMS1[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	ADSMEN0	—	—	—	—	—	—	ADSMS0[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	ADSMS0[1:0]	A/D Conversion Start Request Signal Monitor 0 Selection	b1 b0 0 0: A/D conversion start request signal generated by the GTADTRA register during up-counting 0 1: A/D conversion start request signal generated by the GTADTRA register during down-counting 1 0: A/D conversion start request signal generated by the GTADTRB register during up-counting 1 1: A/D conversion start request signal generated by the GTADTRB register during down-counting	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	ADSMEN0	A/D Conversion Start Request Signal Monitor 0 Output Enabling	0: Output of A/D conversion start request signal monitor 0 is disabled. 1: Output of A/D conversion start request signal monitor 0 is enabled.	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b17, b16	ADSMS1[1:0]	A/D Conversion Start Request Signal Monitor 1 Selection	b17b16 0 0: A/D conversion start request signal generated by the GTADTRA register during up-counting 0 1: A/D conversion start request signal generated by the GTADTRA register during down-counting 1 0: A/D conversion start request signal generated by the GTADTRB register during up-counting 1 1: A/D conversion start request signal generated by the GTADTRB register during down-counting	R/W
b23 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b24	ADSMEN1	A/D Conversion Start Request Signal Monitor 1 Output Enabling	0: Output of A/D conversion start request signal monitor 1 is disabled. 1: Output of A/D conversion start request signal monitor 1 is enabled.	R/W
b31 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to control monitors for the A/D conversion start request signal that is synchronized with a frame.

ADSMSk[1:0] Bits (A/D Conversion Start Request Signal Monitor k Selection) (k = 0, 1)

These bits are used to select A/D conversion start request signal synchronized with a frame which is monitored by the GTADSMk pin.

In triangle-wave PWM mode or complementary PWM mode, the following settings are prohibited:

- Set the ADSMSk[1:0] bits to 00b (A/D conversion start request during up-counting) when GTADTRA = 0
- Set the ADSMSk[1:0] bits to 10b (A/D conversion start request during up-counting) when GTADTRB = 0

- Set the ADSMSk[1:0] bits to 01b (A/D conversion start request during down-counting) when GTADTRA = GTPR
- Set the ADSMSk[1:0] bits to 11b (A/D conversion start request during down-counting) when GTADTRB = GTPR

ADSMENk Bit (A/D Conversion Start Request Signal Monitor k Output Enabling) (k = 0, 1)

This bit enables or disables the monitor output to the GTADSMk pin.
When the output is disabled, the GTADSMk pin goes to the low level.

When the bit is 1, the signal on the GTADSMk pin goes to the high level on assertion of the signal to request to the start of A/D conversion selected by the ADSMSk[1:0] bits and returns to the low level at the end of the current cycle of the timer for the channel that generated the given signal to request the start of A/D conversion. When the counter stops, the value when the counter stopped is retained for output. Set the ADSMENk bit to 0 to output the low level.
When a signal to request the start of A/D conversion is generated at the end of a timer period, the generation of this signal has priority in terms of monitoring output and the output remains at the high level till the end of the next period.
When the output of the same A/D conversion start request signal monitoring output is enabled for multiple channels, ORed signals will be output from the GPTW.

24.2.33 General PWM Timer Extended Interrupt Skipping Counter Control Register (GTEITC)

Address(es): GPTW0.GTEITC 000C 20A8h, GPTW1.GTEITC 000C 21A8h, GPTW2.GTEITC 000C 22A8h, GPTW3.GTEITC 000C 23A8h, GPTW4.GTEITC 000C 24A8h, GPTW5.GTEITC 000C 25A8h, GPTW6.GTEITC 000C 26A8h, GPTW7.GTEITC 000C 27A8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	EITCNT2[3:0]				EITCNT2IV[3:0]				EIVTT2[3:0]				—	—	EIVTC2[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	EITCNT1[3:0]				—	—	—	—	EIVTT1[3:0]				—	—	EIVTC1[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	EIVTC1[1:0]	Extended Interrupt Skipping Counter 1 Count Source Select	b1 b0 0 0: Not counted (not skipped) 0 1: Counting both at overflow or underflow in sawtooth-wave mode, and counting crests in triangle-wave mode 1 0: Counting both at overflow or underflow in sawtooth-wave mode, and counting troughs in triangle-wave mode 1 1: Counting both at overflow or underflow in sawtooth-wave mode, and counting both crests and troughs in triangle-wave mode	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7 to b4	EIVTT1[3:0]	Extended Interrupt Skipping 1 Skipping Count Setting	Skipping count for the extended interrupt skipping 1	R/W
b11 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b12	EITCNT1[3:0]	Extended Interrupt Skipping Counter 1	The extended interrupt skipping counter 1	R
b17, b16	EIVTC2[1:0]	Extended Interrupt Skipping Counter 2 Count Source Select	b17b16 0 0: Not counted (not skipped) 0 1: Counting both at overflow or underflow in sawtooth-wave mode, and counting crests in triangle-wave mode 1 0: Counting both at overflow or underflow in sawtooth-wave mode, and counting troughs in triangle-wave mode 1 1: Counting both at overflow or underflow in sawtooth-wave mode, and counting both crests and troughs in triangle-wave mode	R/W
b19, b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23 to b20	EIVTT2[3:0]	Extended Interrupt Skipping 2 Skipping Count Setting	Skipping count for the extended interrupt skipping 2	R/W
b27 to b24	EITCNT2IV[3:0]	Extended Interrupt Skipping Counter 2 Initial Value	The extended interrupt skipping counter 2 initial value	R/W *1
b31 to b28	EITCNT2[3:0]	Extended Interrupt Skipping Counter 2	The extended interrupt skipping counter 2	R

Note 1. The EITCNT2IV[3:0] bits are only writable when the value other than 00b is written to the EIVTC2[1:0] bits that have been 00b.

The GTEITC register sets the extended interrupt skipping function to skip the interrupts, A/D conversion start requests, and buffer transfers independently by counting at overflow and underflow in the GTCNT counter.
The setting is operated independently from the interrupt skipping by the GTITC, GTADCMSC register.
The setting is invalid during the event count operation.

Access in 8-bit units to GTEITC is prohibited.

EIVTCk[1:0] Bits (Extended Interrupt Skipping k Count Source Select) (k = 1, 2)

These bits select the way of counting for the extended interrupt skipping counter k. Setting only with these bits does not skip the interrupts, A/D conversion start requests, and buffer transfers.
Skipping function for the interrupt, A/D conversion start request, and buffer transfer, all of which are a target of skipping, is set individually with the GTEITLI1, GTEITLI2, and GTEITLB registers.

EIVTTk[3:0] Bits (The Extended Interrupt Skipping k Skipping Count Setting) (k = 1, 2)

A count for the period with continuous skipping is set as a skipping count, where a period is from a generation of a count source selected by the EIVTCk[1:0] bits to the next generation of the count source. When the count source is generated while the EIVTTk[3:0] bits match the EITCNTk[3:0] bits, the EITCNTk[3:0] bits are cleared.
When these bits are 0000b, skipping is not performed.

EITCNT1[3:0] Bits (Extended Interrupt Skipping Counter 1)

The counting is incremented by 1 every time a count source (overflow/underflow/crest/trough) selected by the EIVTC1[1:0] bits is generated.
Counting is performed periodically within the range between 0 and the EIVTT1[3:0] bits.
Even if the GTCNT counter is stopped, the value is not cleared, and value at stop of the GTCNT counter is retained.
[Clearing conditions]

- 00b is written to the EIVTC1[1:0] bits
- 0000b is written to the EIVTT1[3:0] bits
- A count source (overflow/underflow/crest/trough) selected by the EIVTC1[1:0] bits is generated when the extended interrupt skipping 1 skipping count set by the EIVTT1[3:0] bits match the value for the EITCNT1[3:0]

EITCNT2IV[3:0] Bits (Extended Interrupt Skipping Counter 2 Initial Value)

These bits are the value of the initial value for the extended interrupt skipping counter 2. Writing to the EITCNT2IV[3:0] is performed only when the writing value to the EIVTC2[1:0] bits are other than 00b and when the GTEITC register is written by the access of upper 16 bits or 32 bits while the EITCNT2[3:0] bits are set not to count (EIVTC2[1:0] bits are 00b). When the EITCNT2IV[3:0] bits are written, the value written to the EITCNT2IV[3:0] bits is written to the EITCNT2[3:0] bits simultaneously.
The writing to the EITCNT2IV[3:0] bits are ignored when the EITCNT2[3:0] bits are set to count (EIVTC2[1:0] bits are other than 00b) or perform the setting of not to count (00b is written to the EIVTC2[1:0] bits).
The EITCNT2IV[3:0] bits are not reset by the writing 00b to the EIVTC2[1:0] bits.

EITCNT2[3:0] Bits (Extended Interrupt Skipping Counter 2)

The counting is incremented by 1 every time a count source (overflow/underflow/crest/trough) selected by the EIVTC2[1:0] bits is generated.
Counting is performed periodically within the range between 0 and the EIVTT2[3:0] bits.
Even if the GTCNT counter is stopped, the value is not cleared, and value at stop of the GTCNT counter is retained.
Setting of the initial value for the EITCNT2[3:0] bits are performed only when the writing value to the EIVTC2[1:0] bits are other than 00b and when the GTEITC register is written by the access of upper 16 bits or 32 bits while the extended interrupt skipping counter 2 is set as not to count (00b is written to the EIVTC2[1:0] bits).
When the initial value is set, the written value to the EITCNT2IV[3:0] bits is written to the EITCNT2[3:0] bits as the initial value.
[Clearing conditions]

- 00b is written to the EIVTC2[1:0] bits
- 0000b is written to the EIVTT2[3:0] bits

- The value other than 00b is written to the EIVTC2[1:0] bits and the 0000b is written to the EITCNT2IV[3:0] bits simultaneously while 00b is set to the EIVTC2[1:0] bits
- A count source (overflow/underflow/crest/trough) selected by the EIVTC2[1:0] bits is generated when the extended interrupt skipping 2 skipping count set by the EIVTT2[3:0] bits match the value for the EITCNT2[3:0]

24.2.34 General PWM Timer Extended Interrupt Skipping Setting Register 1 (GTEITLI1)

Address(es): GPTW0.GTEITLI1 000C 20ACh, GPTW1.GTEITLI1 000C 21ACh, GPTW2.GTEITLI1 000C 22ACh, GPTW3.GTEITLI1 000C 23ACh, GPTW4.GTEITLI1 000C 24ACh, GPTW5.GTEITLI1 000C 25ACh, GPTW6.GTEITLI1 000C 26ACh, GPTW7.GTEITLI1 000C 27ACh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	EITLU[2:0]			—	EITLV[2:0]			—	EITLF[2:0]			—	EITLE[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	EITLD[2:0]			—	EITLC[2:0]			—	EITLB[2:0]			—	EITLA[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	EITLA[2:0]	GTCCRA Register Compare Match/ Input Capture Interrupt Extended Skipping Function Select	See Table 24.6.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6 to b4	EITLB[2:0]	GTCCRB Register Compare Match/ Input Capture Interrupt Extended Skipping Function Select	See Table 24.6.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10 to b8	EITLC[2:0]	GTCCRC Register Compare Match Interrupt Extended Skipping Function Select	See Table 24.6.	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b14 to b12	EITLD[2:0]	GTCCRD Register Compare Match Interrupt Extended Skipping Function Select	See Table 24.6.	R/W
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b18 to b16	EITLE[2:0]	GTCCRE Register Compare Match Interrupt Extended Skipping Function Select	See Table 24.6.	R/W
b19	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b22 to b20	EITLF[2:0]	GTCCRF Register Compare Match Interrupt Extended Skipping Function Select	See Table 24.6.	R/W
b23	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b26 to b24	EITLV[2:0]	Overflow Interrupt Extended Skipping Function Select	See Table 24.6.	R/W
b27	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b30 to b28	EITLU[2:0]	Underflow Interrupt Extended Skipping Function Select	See Table 24.6.	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

The GTEITLI1 register sets the extended skipping function for interrupts such as compare match/input capture, overflow, and underflow.

Setting only with this register does not perform skipping. The GTEITC register should be set so that a corresponding extended interrupt skipping counter operates the counting.

The setting is operated independently from the interrupt skipping by the GTITC register.

The setting is invalid during the event count operation.

EITLm[2:0] Bits (GTCCRm Register Compare Match/Input Capture Interrupt Extended Skipping Function Select) (m = A, B)

These bits select the extended interrupt skipping function to skip the compare match/input capture interrupt (GTCIm) in the GTCCRm register.
Refer to Table 24.6.

EITLx[2:0] Bits (GTCCRx Register Compare Match Interrupt Extended Skipping Function Select) (x = C, D, E, F)

These bits select the extended interrupt skipping function to skip the compare match interrupt (GTCIx) in the GTCCRx register.
Refer to Table 24.6.

EITLV[2:0] Bits (Overflow Interrupt Extended Skipping Function Select)

These bits select the extended interrupt skipping function to skip the interrupt at overflow.
Refer to Table 24.6.

EITLU[2:0] Bits (Underflow Interrupt Extended Skipping Function Select)

These bits select the extended interrupt skipping function to skip the interrupt at underflow.
Refer to Table 24.6.

Table 24.6 Setting the Function Select for the GTEITLI1 Register

EITLy[2:0] Bits	Function
0 0 0	Do not perform an extended interrupt skipping
0 0 1	Skip an interrupt in the period when the value for the extended interrupt skipping counter 1 is other than 0 (An interrupt is output in the period of EITCNT1 bit = 0)
0 1 0	Skip an interrupt in the period when the value for the extended interrupt skipping counter 2 is other than 0 (An interrupt is output in the period of EITCNT2 bit = 0)
0 1 1	Skip an interrupt in the period the value for the extended interrupt skipping counter 1 or 2 is other than 0 (An interrupt is output in the period of EITCNT1 bit = 0 and EITCNT2 bit = 0)
1 0 0	Do not set this value
1 0 1	Skip an interrupt in the period when the value for the extended interrupt skipping counter 1 is other than the skipping count (An interrupt is output in the period of EITCNT1 bit = EIVTT1 bit)
1 1 0	Skip an interrupt in the period when the value for the extended interrupt skipping counter 2 is other than the skipping count (An interrupt is output in the period of EITCNT2 bit = EIVTT2 bit)
1 1 1	Skip an interrupt in the period when the value for the extended interrupt skipping counter 1 or 2 is other than the skipping count (An interrupt is output in the period of EITCNT1 bit = EIVTT1 bit and EITCNT2 bit = EIVTT2 bit)

y = A, B, C, D, E, F, V, U; k = 1, 2

Note: When the intended skipping counter is set as not to count (EIVTCk[1:0]bits = 00b or EIVTTk[3:0] bits = 0000b), skipping is not performed.

Note: When the EITLy[2:0] bits are set to 011b or 111b, or when one of the skipping counters is set as not to count, skipping is not performed.

24.2.35 General PWM Timer Extended Interrupt Skipping Setting Register 2 (GTEITLI2)

Address(es): GPTW0.GTEITLI2 000C 20B0h, GPTW1.GTEITLI2 000C 21B0h, GPTW2.GTEITLI2 000C 22B0h, GPTW3.GTEITLI2 000C 23B0h, GPTW4.GTEITLI2 000C 24B0h, GPTW5.GTEITLI2 000C 25B0h, GPTW6.GTEITLI2 000C 26B0h, GPTW7.GTEITLI2 000C 27B0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	EADTBL[2:0]			—	EADTAL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	EADTAL[2:0]	GTADTRA A/D Conversion Start Request Extended Skipping Function Select	See Table 24.7.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6 to b4	EADTBL[2:0]	GTADTRB A/D Conversion Start Request Extended Skipping Function Select	See Table 24.7.	R/W
b31 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTEITLI2 register sets the extended skipping function for A/D conversion start requests.
Setting only with this register does not perform skipping. The GTEITC register should be set so that a corresponding extended interrupt skipping counter operates the counting.
The setting is operated independently from the interrupt skipping by the GTITC register.
The setting is invalid during the event count operation.

EADTmL[2:0] Bits (GTADTRm A/D Conversion Start Request Extended Skipping Function Select) (m = A, B)

These bits select the extended interrupt skipping function to skip A/D conversion start requests for the compare match in the GTADTRm register.
Refer to Table 24.7.

Table 24.7 Setting the Function Select for the GTEITLI2 Register

EADTmL[2:0] Bits	Function
0 0 0	Do not perform an extended interrupt skipping
0 0 1	Skip an A/D conversion start request in the period when the value for the extended interrupt skipping counter 1 is other than 0 (An A/D conversion start request is output in the period of EITCNT1 bit = 0)
0 1 0	Skip an A/D conversion start request in the period when the value for the extended interrupt skipping counter 2 is other than 0 (An A/D conversion start request is output in the period of EITCNT2 bit = 0)
0 1 1	Skip an A/D conversion start request in the period when the value for the extended interrupt skipping counter 1 or 2 is other than 0 (An A/D conversion start request is output in the period of EITCNT1 bit = 0 and EITCNT2 bit = 0)
1 0 0	Do not set this value
1 0 1	Skip an A/D conversion start request in the period when the value for the extended interrupt skipping counter 1 is other than the skipping count (An A/D conversion start request is output in the period of EITCNT1 bit = EIVTT1 bit)
1 1 0	Skip an A/D conversion start request in the period when the value for the extended interrupt skipping counter 2 is other than the skipping count (An A/D conversion start request is output in the period of EITCNT2 bit = EIVTT2 bit)
1 1 1	Skip an A/D conversion start request in the period when the value for the extended interrupt skipping counter 1 or 2 is other than the skipping count (An A/D conversion start request is output in the period of EITCNT1 bit = EIVTT1 bit and EITCNT2 bit = EIVTT2 bit)

m = A, B; k = 1, 2

Note: When the intended skipping counter is set as not to count (EIVTCk[1:0]bits = 00b or EIVTTk[3:0] bits = 0000b), skipping is not performed.

Note: When the EADTmL[2:0] bits are set to 011b or 111b, or when one of the skipping counters is set as not to count, skipping is not performed.

24.2.36 General PWM Timer Extended Buffer Transfer Skipping Setting Register (GTEITLB)

Address(es): GPTW0.GTEITLB 000C 20B4h, GPTW1.GTEITLB 000C 21B4h, GPTW2.GTEITLB 000C 22B4h, GPTW3.GTEITLB 000C 23B4h, GPTW4.GTEITLB 000C 24B4h, GPTW5.GTEITLB 000C 25B4h, GPTW6.GTEITLB 000C 26B4h, GPTW7.GTEITLB 000C 27B4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	EBTLDVD[2:0]			—	EBTLDVU[2:0]			—	EBTLADB[2:0]			—	EBTLADA[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	EBTLPR[2:0]			—	EBTLCB[2:0]			—	EBTLCA[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	EBTLCA[2:0]	GTCCRA Register Buffer Transfer Extended Skipping Function Select	See Table 24.8.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6 to b4	EBTLCB[2:0]	GTCCRB Register Buffer Transfer Extended Skipping Function Select	See Table 24.8.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10 to b8	EBTLPR[2:0]	GTPR Register Buffer Transfer Extended Skipping Function Select	See Table 24.8.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b18 to b16	EBTLADA[2:0]	GTADTRA Register Buffer Transfer Extended Skipping Function Select	See Table 24.8.	R/W
b19	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b22 to b20	EBTLADB[2:0]	GTADTRB Register Buffer Transfer Extended Skipping Function Select	See Table 24.8.	R/W
b23	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b26 to b24	EBTLDVU[2:0]	GTDVU Register Buffer Transfer Extended Skipping Function Select	See Table 24.8.	R/W
b27	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b30 to b28	EBTLDVD[2:0]	GTDVD Register Buffer Transfer Extended Skipping Function Select	See Table 24.8.	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

The GTEITLB register sets the extended skipping function for buffer transfers.
Setting only with this register does not perform skipping. The GTEITC register should be set so that a corresponding extended interrupt skipping counter operates the counting.
The setting is operated independently from the interrupt skipping by the GTITC register.
The buffer transfer from the GTOLBR register to the GTIOR.GTIOA[4:0], GTIOB[4:0] bits is not target of the extended buffer transfer skipping function.
The setting is invalid during the event count operation.

EBTLCA[2:0] Bits (GTCCRA Register Buffer Transfer Extended Skipping Function Select)

These bits select the extended buffer transfer skipping function to skip buffer transfers (transfers among the GTCCRA, GTCCRC, GTCCRD registers and a temporary register A) in the GTCCRA register.
An extended skipping of buffer transfers in the GTCCRA register is valid for forcible buffer transfers by the GTBER.CCRSWT bit while the count operation is stopped. Forcible buffer transfers in the GTCCRA register should be

performed in the condition of not performing the extended buffer transfer skipping.
The buffer transfer between the GTCCRC, GTCCRE register, and GTCCRA register in complementary PWM mode cannot be skipped.

EBTLCB[2:0] Bits (GTCCRB Register Buffer Transfer Extended Skipping Function Select)

These bits select the extended buffer transfer skipping function to skip buffer transfers (transfers among the GTCCRB, GTCCRE, GTCCRF registers and a temporary register B) in the GTCCRB register.
An extended skipping of buffer transfers in the GTCCRB register is valid for forcible buffer transfers by the GTBER.CCRSWT bit while the count operation is stopped. Forcible buffer transfers in the GTCCRB register should be performed in the condition of not performing the extended buffer transfer skipping.

EBTLPR[2:0] Bits (GTPR Register Buffer Transfer Extended Skipping Function Select)

These bits select the extended buffer transfer skipping function to skip buffer transfers (transfers among the GTPR, GTPBR, and GTPDBR registers) in the GTPR register.
When skipping GTPR buffer transfers in complementary PWM mode, GTEITC register setting of slave channel should be matched to the master channel so that the buffer transfer timing of the slave channel matches the master channel.

EBTLADm[2:0] Bits (GTADTRm Register Buffer Transfer Extended Skipping Function Select) (m = A, B)

These bits select the extended buffer transfer skipping function to skip buffer transfers (transfers among the GTADTRm, GTADTBRm, and GTADTDBRm registers) in the GTADTRm register.

EBTLDVm[2:0] Bits (GTDVm Register Buffer Transfer Extended Skipping Function Select) (m = U, D)

These bits select the extended buffer transfer skipping function to skip buffer transfers (transfers between the GTDVm and GTDBm registers) in the GTDVm register.

Table 24.8 Setting the Function Select for the GTEITLB Register

EBTLx[2:0] Bits	Function
0 0 0	Do not perform an extended interrupt skipping
0 0 1	Skip a buffer transfer in the period when the value for the extended interrupt skipping counter 1 is other than 0 (A buffer transfer is output in the period of EITCNT1 bit = 0)
0 1 0	Skip a buffer transfer in the period when the value for the extended interrupt skipping counter 2 is other than 0 (A buffer transfer is output in the period of EITCNT2 bit = 0)
0 1 1	Skip a buffer transfer in the period when the value for the extended interrupt skipping counter 1 or 2 is other than 0 (A buffer transfer is output in the period of EITCNT1 bit = 0 and EITCNT2 bit = 0)
1 0 0	Do not set this value
1 0 1	Skip a buffer transfer in the period when the value for the extended interrupt skipping counter 1 is other than the skipping count (A buffer transfer is output in the period of EITCNT1 bit = EIVTT1 bit)
1 1 0	Skip a buffer transfer in the period when the value for the extended interrupt skipping counter 2 is other than the skipping count (A buffer transfer is output in the period of EITCNT2 bit = EIVTT2 bit)
1 1 1	Skip a buffer transfer in the period when the value for the extended interrupt skipping counter 1 or 2 is other than the skipping count (A buffer transfer is output in the period of EITCNT1 bit = EIVTT1 bit and EITCNT2 bit = EIVTT2 bit)

x = CA, CB, PR, ADA, ADB, DVU, DVD; k = 1, 2
Note: When the intended skipping counter is set as not to count (EIVTCk[1:0]bits = 00b or EIVTTk[3:0] bits = 0000b), skipping is not performed.
Note: When the EBTLx[2:0] bits are set to 011b or 111b, or when one of the skipping counters is set as not to count, skipping is not performed.

24.2.37 General PWM Timer Inter Channel Logical Operation Function Setting Register (GTICLF)

Address(es): GPTW0.GTICLF 000C 20B8h, GPTW1.GTICLF 000C 21B8h, GPTW2.GTICLF 000C 22B8h, GPTW3.GTICLF 000C 23B8h, GPTW4.GTICLF 000C 24B8h, GPTW5.GTICLF 000C 25B8h, GPTW6.GTICLF 000C 26B8h, GPTW7.GTICLF 000C 27B8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	ICLFSELD[5:0]						—	ICLFB[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	ICLFSELC[5:0]						—	ICLFA[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	ICLFA[2:0]	GTIOCnA Output Logical Operation Function Select	b2 b0 0 0 0: A (no delay) 0 0 1: NOT A (no delay) 0 1 0: C (1 PCLKC delay) 0 1 1: NOT C (1 PCLKC delay) 1 0 0: A AND C (1 PCLKC delay)*2 1 0 1: A OR C (1 PCLKC delay)*2 1 1 0: A EXOR C (1 PCLKC delay)*2 1 1 1: A NOR C (1 PCLKC delay)*2	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b9 to b4	ICLFSELC[5:0]	Inter Channel Signal C Select*1*2	b9 b4 0 0 0 0 0 0: GTIOC0A 0 0 0 0 0 1: GTIOC0B 0 0 0 0 1 0: GTIOC1A 0 0 0 0 1 1: GTIOC1B 0 0 0 1 0 0: GTIOC2A 0 0 0 1 0 1: GTIOC2B 0 0 0 1 1 0: GTIOC3A 0 0 0 1 1 1: GTIOC3B 0 0 1 0 0 0: GTIOC4A 0 0 1 0 0 1: GTIOC4B 0 0 1 0 1 0: GTIOC5A 0 0 1 0 1 1: GTIOC5B 0 0 1 1 0 0: GTIOC6A 0 0 1 1 0 1: GTIOC6B 0 0 1 1 1 0: GTIOC7A 0 0 1 1 1 1: GTIOC7B Settings other than above are prohibited.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b18 to b16	ICLFB[2:0]	GTIOCnB Output Logical Operation Function Select	b18 b16 0 0 0: B (no delay) 0 0 1: NOT B (no delay) 0 1 0: D (1 PCLKC delay) 0 1 1: NOT D (1 PCLKC delay) 1 0 0: B AND D (1 PCLKC delay)*3 1 0 1: B OR D (1 PCLKC delay)*3 1 1 0: B EXOR D (1 PCLKC delay)*3 1 1 1: B NOR D (1 PCLKC delay)*3	R/W
b19	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit Name	Description	R/W
b25 to b20	ICLFSELD[5:0]	Inter Channel Signal D Select*1*3	b25 b20 0 0 0 0 0 0: GTIOC0A 0 0 0 0 0 1: GTIOC0B 0 0 0 0 1 0: GTIOC1A 0 0 0 0 1 1: GTIOC1B 0 0 0 1 0 0: GTIOC2A 0 0 0 1 0 1: GTIOC2B 0 0 0 1 1 0: GTIOC3A 0 0 0 1 1 1: GTIOC3B 0 0 1 0 0 0: GTIOC4A 0 0 1 0 0 1: GTIOC4B 0 0 1 0 1 0: GTIOC5A 0 0 1 0 1 1: GTIOC5B 0 0 1 1 0 0: GTIOC6A 0 0 1 1 0 1: GTIOC6B 0 0 1 1 1 0: GTIOC7A 0 0 1 1 1 1: GTIOC7B Settings other than above are prohibited.	R/W
b31 to b26	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. The signal before performing output disable control is selected.
Note 2. When channel's own GTIOCnA is selected, C is treated as 1.
Note 3. When channel's own GTIOCnB is selected, D is treated as 1.

The GTICLF register sets the logical operation function between compare match outputs.
The logical operation is performed with the signals that the duty 0%/100% control is performed after compare match control.
The output disable control is performed with the signal after logical operation.
Access in 8-bit units to the GTICLF register is prohibited.

ICLFm[2:0] Bits (GTIOCnm Output Logical Operation Function Select) (n = 0 to 7) (m = A, B)

These bits select the logical operation function between signals before performing output disable control for GTIOCnm.
When the same signal to operate logical function AND, OR, EXOR and NOR is selected, one signal is treated as 1.

ICLFSELk[5:0] Bits (Inter Channel Signal k Select) (k = C, D)

These bits select the signal k that the logical operation is performed with the signal before performing output disable control for GTIOCnm.

24.2.38 General PWM Timer Cycle Count Register (GTPC)

Address(es): GPTW0.GTPC 000C 20BCh, GPTW1.GTPC 000C 21BCh, GPTW2.GTPC 000C 22BCh, GPTW3.GTPC 000C 23BCh, GPTW4.GTPC 000C 24BCh, GPTW5.GTPC 000C 25BCh, GPTW6.GTPC 000C 26BCh, GPTW7.GTPC 000C 27BCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	PCNT[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	ASTP	—	—	—	—	—	—	—	PCEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PCEN	Cycle Count Function Enable	0: Cycle count function is disabled. 1: Cycle count function is enabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	ASTP	Automatic Stop Function Enable	0: Automatic function is disabled. 1: Automatic function is enabled.	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b27 to b16	PCNT[11:0]	Cycle Counter	Counter for number of cycles.	R/W
b31 to b28	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTPC register counts the number of period.

PCEN Bit (Cycle Count Function Enable)

This bit enables or disables cycle count function.
Writing is available when counting is both in progress and stopped.
When 1 is written to either the GTSECR.SPCE bit or the GTSECR.SPCD bit, the value is simultaneously set to the PCEN bit in the channels set to 1 by the GTSECSR register.

ASTP Bit (Automatic Stop Function Enable)

This bit enables or disables the GTCNT counter automatic stopping after finishing counting the number of period.
When the PCEN bis is 0, writing is available.
When the PCEN bit is 1, writing is disabled.
When the PCEN bit is 1, the ASTP bit is 1, and the PCNT counter is stopped at PCNT = 0, the GTCNT counter is also stopped. When the ASTP bit is 0, the GTCNT counter continues to count.

PCNT[11:0] Bits (Cycle Counter)

This counter counts the number of cycles.
When the PCEN bis is 0, writing the number of cycles is available.
When the PCEN bit is 1, writing is disabled, and down-counting is performed at the end of cycle. In sawtooth-wave mode, the end of cycle refers to overflow, underflow, or GTCNT counter clearing. In triangle-wave mode or complementary PWM mode, it refers to trough.
When the PCNT counter is 1 at the end of cycle, it becomes 0 and counting is stopped.
When the GTCNT counter is stopped while cycle count function is enabled, the PCNT counter keeps its value. When the GTCNT counter restarts counting and the PCEN bit is 1, the PCNT counter restarts down-counting from the hold value.
When the PCEN bit is changed from 0 to 1 while the PCNT counter is 0 and the ASTP bit is 1, the GTCNT counter is

stopped at the count clock immediately after that.

24.2.39 General PWM Timer A/D Conversion Start Request Compare Match Skipping Control Register (GTADCMSC)

Address(es): GPTW0.GTADCMSC 000C 20C0h, GPTW1.GTADCMSC 000C 21C0h, GPTW2.GTADCMSC 000C 22C0h, GPTW3.GTADCMSC 000C 23C0h, GPTW4.GTADCMSC 000C 24C0h, GPTW5.GTADCMSC 000C 25C0h, GPTW6.GTADCMSC 000C 26C0h, GPTW7.GTADCMSC 000C 27C0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ADCMSCNT2[3:0]				ADCMSCNT2IV[3:0]				ADCMST2[3:0]				—	—	ADCMSC2[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ADCMSCNT1[3:0]				ADCMSCNT1IV[3:0]				ADCMST1[3:0]				—	—	ADCMSC1[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	ADCMSC1[1:0]	A/D Conversion Start Request Compare Match Skipping Counter 1 Count Source Select	b1 b0 0 0: Not counted (not skipped) 0 1: Counting GTADTRA register compare match 1 0: Counting GTADTRB register compare match 1 1: Counting both GTADTRA register compare match and GTADTRB register compare match	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7 to b4	ADCMST1[3:0]	A/D Conversion Start Request Compare Match Skipping 1 Skipping Count Setting	Skipping count for the A/D conversion start request compare match skipping 1.	R/W
b11 to b8	ADCMSCNT1IV[3:0]	A/D Conversion Start Request Compare Match Skipping Counter 1 Initial Value	A/D Conversion Start Request Compare Match Skipping Counter 1 Initial Value	R/W *1
b15 to b12	ADCMSCNT1[3:0]	A/D Conversion Start Request Compare Match Skipping Counter 1	A/D Conversion Start Request Compare Match Skipping Counter 1	R/W
b17, b16	ADCMSC2[1:0]	A/D Conversion Start Request Compare Match Skipping Counter 2 Count Source Select	b1 b0 0 0: Not counted (not skipped) 0 1: Counting GTADTRA register compare match 1 0: Counting GTADTRB register compare match 1 1: Counting both GTADTRA register compare match and GTADTRB register compare match	R/W
b19, b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23 to b20	ADCMST2[3:0]	A/D Conversion Start Request Compare Match Skipping 2 Skipping Count Setting	Skipping count for the A/D conversion start request compare match skipping 2	R/W
b27 to b24	ADCMSCNT2IV[3:0]	A/D Conversion Start Request Compare Match Skipping Counter 2 Initial Value	A/D Conversion Start Request Compare Match Skipping Counter 2 Initial Value	R/W *2
b31 to b28	ADCMSCNT2[3:0]	A/D Conversion Start Request Compare Match Skipping Counter 2	A/D Conversion Start Request Compare Match Skipping Counter 2	R/W

Note 1. Writing is possible only when the ADCMSC1[1:0] bits are 00b and a value other than 00b is written to the ADCMSC1[1:0] bits.
Note 2. Writing is possible only when the ADCMSC2[1:0] bits are 00b and a value other than 00b is written to the ADCMSC2[1:0] bits.

The GTADCMSC register is a register that controls the skipping counter of the A/D conversion start request compare match skipping function that counts compare matches of GTADTRA register and GTADTRB register, and skipping A/D

conversion start request and buffer transfer independently.
This register setting is operated independently from the interrupt skipping by the GTITC register or GTEITC register.
Access in 8-bit units to GTADCMSC is prohibited.

ADCMSCk[1:0] Bits (A/D Conversion Start Request Compare Match Skipping Counter k Count Source Select) (k = 1, 2)

These bits select the way of counting for the A/D conversion start request compare match skipping counter k.
Setting only with these bits does not skip the A/D conversion start requests and buffer transfers. Skipping function for the A/D conversion start request and buffer transfer which are a target of skipping is set individually with the GTADCMSS register.

ADCMSTk[3:0] Bits (A/D Conversion Start Request Compare Match Skipping k Skipping Count Setting) (k = 1, 2)

A count for the period with continuous skipping is set as a skipping count, where a period is from a generation of a count source selected by the ADCMSCk[1:0] bits to the next generation of the count source.
When the count source is generated while the ADCMSTk[3:0] bits match the ADCMSCNTk[3:0] bits, the ADCMSCNTk[3:0] bits are cleared.
When these bits are 0000b, skipping is not performed.

ADCMSCNTkIV[3:0] Bits (A/D Conversion Start Request Compare Match Skipping Counter k Initial Value) (k = 1, 2)

These bits are the value of the initial value for the A/D conversion start request compare match skipping counter k.
Writing to the ADCMSCNTkIV[3:0] is performed only when the writing value to the ADCMSCk[1:0] bits are other than 00b and when the GTADCMSC register is written by the access of 16 bits or 32 bits while the ADCMSCNTk[3:0] bits are set not to count (ADCMSCk[1:0] bits are 00b). When the ADCMSCNTkIV[3:0] bits are written, the value written to the ADCMSCNTkIV[3:0] bits is written to the ADCMSCNTk[3:0] bits simultaneously.
The writing to the ADCMSCNTkIV[3:0] bits are ignored when the ADCMSCNTk[3:0] bits are set to count (ADCMSCk[1:0] bits are other than 00b) or perform the setting of not to count (00b is written to the ADCMSCk[1:0] bits).
The ADCMSCNTkIV[3:0] bits are not reset by the writing 00b to the ADCMSCk[1:0] bits.

ADCMSCNTk[3:0] Bits (A/D Conversion Start Request Compare Match Skipping Counter k) (k = 1, 2)

The counting is incremented by 1 every time a count source selected by the ADCMSCk[1:0] bits is generated.
Counting is performed periodically within the range between 0 and the ADCMSTk[3:0] bits.
Even if the GTCNT counter is stopped, the value is not cleared, and value at stop of the GTCNT counter is retained.
Setting of the initial value for the ADCMSCNTk[3:0] bits are performed only when the writing value to the ADCMSCk[1:0] bits are other than 00b and when the GTADCMSC register is written by the access of 16 bits or 32 bits while the A/D conversion start request compare match skipping counter k is set as not to count (ADCMSCk[1:0] bits are 00b).
When the initial value is set, the written value to the ADCMSCNTkIV[3:0] bits is written to the ADCMSCNTk[3:0] bits as the initial value.
[Clearing condition]

- 00b is written to the ADCMSCk[1:0] bits.
- 0000b is written to the ADCMSTk[3:0] bits.
- The value other than 00b is written to the ADCMSCk[1:0] bits and the 0000b is written to the ADCMSCNTkIV[3:0] bits simultaneously while 00b is set to the ADCMSCk[1:0] bits.
- A count source selected by the ADCMSCk[1:0] bits is generated when the extended interrupt skipping k skipping count set by the ADCMSTk[3:0] bits match the value for the ADCMSCNTk[3:0].

24.2.40 General PWM Timer A/D Conversion Start Request Compare Match Skipping Setting Register (GTADCMSS)

Address(es): GPTW0.GTADCMSS 000C 20C4h, GPTW1.GTADCMSS 000C 21C4h, GPTW2.GTADCMSS 000C 22C4h, GPTW3.GTADCMSS 000C 23C4h, GPTW4.GTADCMSS 000C 24C4h, GPTW5.GTADCMSS 000C 25C4h, GPTW6.GTADCMSS 000C 26C4h, GPTW7.GTADCMSS 000C 27C4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	ADCMBSB[2:0]			—	ADCMBSA[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	ADCMSBL[2:0]			—	ADCMSAL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	ADCMSAL[2:0]	GTADTRA Register A/D Conversion Start Request Compare Match Skipping Function Select	See Table 24.9.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6 to b4	ADCMSBL[2:0]	GTADTRB Register A/D Conversion Start Request Compare Match Skipping Function Select	See Table 24.9.	R/W
b15 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b18 to b16	ADCMBSA[2:0]	GTADTRA Register Buffer Transfer by A/D Conversion Start Request Compare Match Skipping Function Select	See Table 24.10.	R/W
b19	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b22 to b20	ADCMBSB[2:0]	GTADTRA Register Buffer Transfer by A/D Conversion Start Request Compare Match Skipping Function Select	See Table 24.10.	R/W
b31 to b23	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTADCMSC register is a register that controls the skipping counter of the A/D conversion start request compare match skipping function that counts compare matches of GTADTRA register and GTADTRB register, and skipping A/D conversion start request and buffer transfer independently.
This register setting is operated independently from the interrupt skipping by the GTITC register or GTEITC register.
Access in 8-bit units to GTADCMSC is prohibited.

ADCMSmL[2:0] Bits (GTADTRm Register A/D Conversion Start Request Compare Match Skipping Function Select) (m = A, B)

These bits select the A/D conversion start request compare match skipping function of the GTADTRm (m = A, B) register.
Refer to Table 24.9.

Table 24.9 Setting of GTADTRm Register A/D Conversion Start Request Compare Match Skipping Function Select bit

ADCMSmL[2:0] Bits	Function
0 0 0	Do not perform an A/D conversion start request compare match skipping
0 0 1	Skip an A/D conversion start request in the period when the value for the A/D conversion start request compare match skipping counter 1 is other than 0000b (An A/D conversion start request is output in the period of the ADCMSCNT1[3:0] bits = 0000b)
0 1 0	Skip an A/D conversion start request in the period when the value for the A/D conversion start request compare match skipping counter 2 is other than 0000b (An A/D conversion start request is output in the period of the ADCMSCNT2[3:0] bits = 0000b)
0 1 1	Skip an A/D conversion start request in the period when the value for the A/D conversion start request compare match skipping counter 1 or 2 is other than 0000b (An A/D conversion start request is output in the period of the ADCMSCNT1[3:0] bits = 0000b and the ADCMSCNT2[3:0] bits = 0000b)
1 0 0	Setting prohibited
1 0 1	Skip an A/D conversion start request in the period when the value for the A/D conversion start request compare match skipping counter 1 is other than the skipping count (An A/D conversion start request is output in the period of the ADCMSCNT1[3:0] bits = the ADCMST1[3:0] bits)
1 1 0	Skip an A/D conversion start request in the period when the value for the A/D conversion start request compare match skipping counter 2 is other than the skipping count (An A/D conversion start request is output in the period of the ADCMSCNT2[3:0] bits = the ADCMST2[3:0] bits)
1 1 1	Skip an A/D conversion start request in the period when the value for the A/D conversion start request compare match skipping counter 1 or 2 is other than the skipping count (An A/D conversion start request is output in the period of the ADCMSCNT1[3:0] bits = the ADCMST1[3:0] bits and the ADCMSCNT2[3:0] bits = the ADCMST2[3:0] bits)

m = A, B

Note: When the intended skipping counter is set as not to count (the ADCMSCk[1:0] bits = 00b or the ADCMSTk[3:0] bits = 0000b), skipping is not performed. (k = 1, 2)

Note: When the ADCMSmL[2:0] bits are set to 011b or 111b, and when one of the skipping counters is set as not to count, skipping is not performed.

ADCMBSm[2:0] Bits (GTADTRm Register Buffer Transfer by A/D Conversion Start Request Compare Match Skipping Function Select) (m = A, B)

These bits select the A/D conversion start request comparison match skip function for buffer transfers in the GTADTRm register (transfers between the GTADTRm register, GTADTBRm register, and GTADTDBRm register).
Refer to Figure 24.8.

Table 24.10 Setting of the GTADTRm Register Buffer Transfer by A/D Conversion Start Request Compare Match Skipping Function Select bit

ADCMBSm[2:0] Bits	Function
0 0 0	Do not perform an GTADTRm register buffer transfer by A/D conversion start request compare match skipping
0 0 1	Skip a buffer transfer in the period when the value for the A/D conversion start request compare match skipping counter 1 is other than 0 (Buffer is transferred in the period of the ADCMSCNT1[3:0] bits = 0000b)
0 1 0	Skip a buffer transfer in the period when the value for the A/D conversion start request compare match skipping counter 2 is other than 0 (Buffer is transferred in the period of the ADCMSCNT2[3:0] bits = 0000b)
0 1 1	Skip a buffer transfer in the period when the value for the A/D conversion start request compare match skipping counter 1 or 2 is other than 0 (Buffer is transferred in the period of the ADCMSCNT1[3:0] bits = 0000b and the ADCMSCNT2[3:0] bits = 0000b)
1 0 0	Setting prohibited
1 0 1	Skip a buffer transfer in the period when the value for the A/D conversion start request compare match skipping counter 1 is other than the skipping count (Buffer is transferred in the period of the ADCMSCNT1[3:0] bits = the ADCMST1[3:0] bits)
1 1 0	Skip a buffer transfer in the period when the value for the A/D conversion start request compare match skipping counter 2 is other than the skipping count (Buffer is transferred in the period of the ADCMSCNT2[3:0] bits = the ADCMST2[3:0] bits)
1 1 1	Skip a buffer transfer in the period when the value for the A/D conversion start request compare match skipping counter 1 or 2 is other than the skipping count (A Buffer is transferred in the period of the ADCMSCNT1[3:0] bits = the ADCMST1[3:0] bits and the ADCMSCNT2[3:0] bits = the ADCMST2[3:0] bits)

m = A, B

Note: When the intended skipping counter is set as not to count (the ADCMSCk[1:0] bits = 00b or the ADCMSTk[3:0] bits = 0000b), skipping is not performed. (k = 1, 2)

Note: When the ADCMBSm[2:0] bits are set to 011b or 111b, and when one of the skipping counters is set as not to count, skipping is not performed.

24.2.41 General PWM Timer Operation Enable Bit Simultaneous Control Channel Select Register (GTSECSR)

Address(es): GPTW0.GTSECSR 000C 20D0h, GPTW1.GTSECSR 000C 21D0h, GPTW2.GTSECSR 000C 22D0h, GPTW3.GTSECSR 000C 23D0h, GPTW4.GTSECSR 000C 24D0h, GPTW5.GTSECSR 000C 25D0h, GPTW6.GTSECSR 000C 26D0h, GPTW7.GTSECSR 000C 27D0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	SECSEL7	SECSEL6	SECSEL5	SECSEL4	SECSEL3	SECSEL2	SECSEL1	SECSEL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SECSEL0	Channel 0 Operation Enable Bit Simultaneous Control Channel Select	0: Disable simultaneous control. 1: Enable simultaneous control.	R/W
b1	SECSEL1	Channel 1 Operation Enable Bit Simultaneous Control Channel Select		R/W
b2	SECSEL2	Channel 2 Operation Enable Bit Simultaneous Control Channel Select		R/W
b3	SECSEL3	Channel 3 Operation Enable Bit Simultaneous Control Channel Select		R/W
b4	SECSEL4	Channel 4 Operation Enable Bit Simultaneous Control Channel Select		R/W
b5	SECSEL5	Channel 5 Operation Enable Bit Simultaneous Control Channel Select		R/W
b6	SECSEL6	Channel 6 Operation Enable Bit Simultaneous Control Channel Select		R/W
b7	SECSEL7	Channel 7 Operation Enable Bit Simultaneous Control Channel Select		R/W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTSECSR register selects an intended channel for updating an operation enable bit by the GTSECR register.
A bit position for the GTSECSR register indicates a channel number. The GTSECSR register of each channel is a common register, and writing 1 to a bit in the GTSECSR register in any channel and updating it changes a channel, related to the position of the bit written with 1 by the GTSECSR register, to be simultaneously controlled of the operation enable bit by the GTSECR register.
Access in 8-bit or 16-bit units to the GTSECSR register is prohibited, and it should be accessed in 32-bit units.

SECSELn Bit (Operation Enable Bit Simultaneous Control Channel Select) (n = 0 to 7)

This bit enables or disables the simultaneous control of operation enable in channel n.
When the bit is set to 1, the simultaneous control is enabled, and disabled when the bit is 0.

24.2.42 General PWM Timer Operation Enable Bit Simultaneous Control Register (GTSECR)

Address(es): GPTW0.GTSECR 000C 20D4h, GPTW1.GTSECR 000C 21D4h, GPTW2.GTSECR 000C 22D4h, GPTW3.GTSECR 000C 23D4h, GPTW4.GTSECR 000C 24D4h, GPTW5.GTSECR 000C 25D4h, GPTW6.GTSECR 000C 26D4h, GPTW7.GTSECR 000C 27D4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	SSCD	SPCD	—	—	—	—	—	—	SSCE	SPCE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	SBDDD	SBDAD	SBDPD	SBDCD	—	—	—	—	SBDDE	SBDAE	SBDPE	SBDCE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SBDCE	GTCCR Register Buffer Operation Simultaneous Enable	0: Disable simultaneous enabling GTCCR buffer operations 1: Enable GTCCR register buffer operations simultaneously	R/W
b1	SBDPE	GTPR Register Buffer Operation Simultaneous Enable	0: Disable simultaneous enabling GTPR buffer operations 1: Enable GTPR register buffer operations simultaneously	R/W *1
b2	SBDAE	GTADTR Register Buffer Operation Simultaneous Enable	0: Disable simultaneous enabling GTADTR buffer operations 1: Enable GTADTR register buffer operations simultaneously	R/W
b3	SBDDE	GTDV Register Buffer Operation Simultaneous Enable	0: Disable simultaneous enabling GTDV buffer operations 1: Enable GTDV register buffer operations simultaneously	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	SBDCD	GTCCR Register Buffer Operation Simultaneous Disable	0: Disable simultaneous disabling GTCCR buffer operations 1: Disable GTCCR register buffer operations simultaneously	R/W
b9	SBDPD	GTPR Register Buffer Operation Simultaneous Disable	0: Disable simultaneous disabling GTPR buffer operations 1: Disable GTPR register buffer operations simultaneously	R/W *1
b10	SBDAD	GTADTR Register Buffer Operation Simultaneous Disable	0: Disable simultaneous disabling GTADTR buffer operations 1: Disable GTADTR register buffer operations simultaneously	R/W
b11	SBDDD	GTDV Register Buffer Operation Simultaneous Disable	0: Disable simultaneous disabling GTDV buffer operations 1: Disable GTDV register buffer operations simultaneously	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	SPCE	Cycle Count Function Simultaneous Enable	0: Disable simultaneous enabling cycle count function 1: Enable cycle count function simultaneously	R/W
b17	SSCE	Synchronous Set/Clear Simultaneous Enable	0: Disable simultaneous enabling synchronous set/clear 1: Enable synchronous set/clear simultaneously	R/W *1
b23 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b24	SPCD	Cycle Count Function Simultaneous Disable	0: Disable simultaneous disabling cycle count function 1: Disable cycle count function simultaneously	R/W
b25	SSCD	Synchronous Set/Clear Simultaneous Disable	0: Disable simultaneous disabling synchronous set/clear 1: Disable synchronous set/clear simultaneously	R/W *1
b31 to b26	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.

The GTSECR register simultaneously updates the value for operation enable bits of a channel set by the GTSECSR register.

Writing 1 to a bit in the GTSECR register in any channel and updating it updates an operation enable bit for all channels, related to the position of the bit written with 1 by the all GTSECR registers. Setting enable and disable bits for the same operation enable bit to 1 in the GTSECR is prohibited.

A bit written to 1 is automatically cleared. When the GTSECR is read, 0 is read.

Access in 8-bit or 16-bit units to the GTSECR register is prohibited, and it should be accessed in 32-bit units.

SBDCE Bit (GTCCR Register Buffer Operation Simultaneous Enable)

When 1 is written to this bit, 0 is simultaneously set to a GTBER.BD[0] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTCCRA, GTCCRC, and GTCCRD registers and using the GTCCRB, GTCCRE, and GTCCRF registers are enabled.
Simultaneous setting of SBDCE and SBDCD bits to 1 is prohibited.

SBDPE Bit (GTPR Register Buffer Operation Simultaneous Enable)

When 1 is written to this bit, 0 is simultaneously set to a GTBER.BD[1] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTPR, GTPBR, and GTPDBR registers are enabled.
Simultaneous setting of SBDPE and SBDPD bits to 1 is prohibited.

SBDAE Bit (GTADTR Register Buffer Operation Simultaneous Enable)

When 1 is written to this bit, 0 is simultaneously set to a GTBER.BD[2] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTADTRA, GTADTBRA, and GTADTDBRA registers and using the GTADTRB, GTADTBRB, and GTADTDBRB registers are enabled.
Simultaneous setting of SBDAE and SBDAD bits to 1 is prohibited.

SBDDE Bit (GTDV Register Buffer Operation Simultaneous Enable)

When 1 is written to this bit, 0 is simultaneously set to a GTBER.BD[3] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTDVU and GTDBU registers and using the GTDVD and GTDBD registers are enabled.
Simultaneous setting of SBDDE and SBDDD bits to 1 is prohibited.

SBDCD Bit (GTCCR Register Buffer Operation Simultaneous Disable)

When 1 is written to this bit, 1 is simultaneously set to a GTBER.BD[0] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTCCRA, GTCCRC, and GTCCRD registers and using the GTCCRB, GTCCRE, and GTCCRF registers are disabled.
Simultaneous setting of SBDCE and SBDCD bits to 1 is prohibited.

SBDPD Bit (GTPR Register Buffer Operation Simultaneous Disable)

When 1 is written to this bit, 1 is simultaneously set to a GTBER.BD[1] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTPR, GTPBR, and GTPDBR registers are disabled.
Simultaneous setting of SBDPE and SBDPD bits to 1 is prohibited.

SBDAD Bit (GTADTR Register Buffer Operation Simultaneous Disable)

When 1 is written to this bit, 1 is simultaneously set to a GTBER.BD[2] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTADTRA, GTADTBRA, and GTADTDBRA registers and using the GTADTRB, GTADTBRB, and GTADTDBRB registers are disabled.
Simultaneous setting of SBDAE and SBDAD bits to 1 is prohibited.

SBDDD Bit (GTDV Register Buffer Operation Simultaneous Disable)

When 1 is written to this bit, 1 is simultaneously set to a GTBER.BD[3] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTDVU and GTDBU registers and using the GTDVD and GTDBD registers are disabled.
Simultaneous setting of SBDDE and SBDDD bits to 1 is prohibited.

SPCE Bit (Cycle Count Function Simultaneous Enable)

Writing 1 to this bit simultaneously sets any GTPC.PCEN bits to 1 for channels with simultaneous control enabled by the setting 1 in the GTSECSR register and enables the cycle-counting function.
Simultaneous setting of SPCE and SPCD bits to 1 is prohibited.

SSCE Bit (Synchronous Set/Clear Simultaneous Enable)

Writing 1 to this bit simultaneously sets any GTCR.SSCEN bits to 1 for channels with simultaneous control enabled by the setting 1 in the GTSECSR register and enables the synchronous set/clear function.
Simultaneous setting of SSCE and SSCD bits to 1 is prohibited.

SPCD Bit (Cycle Count Function Simultaneous Disable)

Writing 1 to this bit simultaneously sets any GTPC.PCEN bits to 0 for channels with simultaneous control enabled by the setting 1 in the GTSECSR register and disables the cycle-counting function.
Simultaneous setting of SPCE and SPCD bits to 1 is prohibited.

SSCD Bit (Synchronous Set/Clear Simultaneous Disable)

Writing 1 to this bit simultaneously sets any GTCR.SSCEN bits to 0 for channels with simultaneous control enabled by the setting 1 in the GTSECSR register and disables the synchronous set/clear function.
Simultaneous setting of SSCE and SSCD bits to 1 is prohibited.

24.2.43 General PWM Timer Buffer Enable Register 2 (GTBER2)

Address(es): GPTW0.GTBER2 000C 20E0h, GPTW1.GTBER2 000C 21E0h, GPTW2.GTBER2 000C 22E0h, GPTW3.GTBER2 000C 23E0h, GPTW4.GTBER2 000C 24E0h, GPTW5.GTBER2 000C 25E0h, GPTW6.GTBER2 000C 26E0h, GPTW7.GTBER2 000C 27E0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	OLTTB[1:0]		OLTTA[1:0]		CPBTD	CP3DB	—	—	CPTDV	CPTADB	CPTADA	CPTPR	CPTCB	CPTCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	CMTADB	CMTADA	—	CMTCB[1:0]		CMTCA[1:0]		—	—	CCTDV	CCTADB	CCTADA	CCTPR	CCTCB	CCTCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CCTCA	Counter Clear Source GTCCRA Register Buffer Transfer Disable	0: Enable GTCCRA register buffer transfer by counter clear 1: Disable GTCCRA register buffer transfer by counter clear	R/W
b1	CCTCB	Counter Clear Source GTCCRB Register Buffer Transfer Disable	0: Enable GTCCRB register buffer transfer by counter clear 1: Disable GTCCRB register buffer transfer by counter clear	R/W
b2	CCTPR	Counter Clear Source GTPR Register Buffer Transfer Disable	0: Enable GTPR register buffer transfer by counter clear 1: Disable GTPR register buffer transfer by counter clear	R/W
b3	CCTADA	Counter Clear Source GTADTRA Register Buffer Transfer Disable	0: Enable GTADTRA register buffer transfer by counter clear 1: Disable GTADTRA register buffer transfer by counter clear	R/W
b4	CCTADB	Counter Clear Source GTADTRB Register Buffer Transfer Disable	0: Enable GTADTRB register buffer transfer by counter clear 1: Disable GTADTRB register buffer transfer by counter clear	R/W
b5	CCTDV	Counter Clear Source GTDVU/GTDVD Register Buffer Transfer Disable	0: Enable GTDVU/GTDVD register buffer transfer by counter clear 1: Disable GTDVU/GTDVD register buffer transfer by counter clear	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	CMTCA[1:0]	Compare Match Source GTCCRA Register Buffer Transfer Enable	b9 b8 0 0: Disable GTCCRA register Buffer Transfer by compare match of GTCCRA register and GTCCRB register 0 1: Enable GTCCRA register Buffer Transfer by compare match of GTCCRA register 1 0: Enable GTCCRA register Buffer Transfer by compare match of GTCCRB register 1 1: Enable GTCCRA register Buffer Transfer by compare match of GTCCRA register and GTCCRB register	R/W
b11, b10	CMTCB[1:0]	Compare Match Source GTCCRB Register Buffer Transfer Enable	b11 b10 0 0: Disable GTCCRB register Buffer Transfer by compare match of GTCCRA register and GTCCRB register 0 1: Enable GTCCRB register Buffer Transfer by compare match of GTCCRA register 1 0: Enable GTCCRB register Buffer Transfer by compare match of GTCCRB register 1 1: Enable GTCCRB register Buffer Transfer by compare match of GTCCRA register and GTCCRB register	R/W
b12	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b13	CMTADA	Compare Match Source GTADTRA Register Buffer Transfer Enable	0: Disable GTADTRA register buffer transfer by compare match of GTADTRA register 1: Enable GTADTRA register buffer transfer by compare match of GTADTRA register	R/W

Bit	Symbol	Bit Name	Description	R/W
b14	CMTADB	Compare Match Source GTADTRB Register Buffer Transfer Enable	0: Disable GTADTRB register buffer transfer by compare match of GTADTRB register 1: Enable GTADTRB register buffer transfer by compare match of GTADTRB register	R/W
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b16	CPTCA	Overflow/Underflow Source GTCCRA Register Buffer Transfer Disable	0: Enable GTCCRA register buffer transfer by overflow/underflow 1: Disable GTCCRA register buffer transfer by overflow/underflow	R/W
b17	CPTCB	Overflow/Underflow Source GTCCRB Register Buffer Transfer Disable	0: Enable GTCCRB register buffer transfer by overflow/underflow 1: Disable GTCCRB register buffer transfer by overflow/underflow	R/W
b18	CPTPR	Overflow/Underflow Source GTPR Register Buffer Transfer Disable	0: Enable GTPR register buffer transfer by overflow/underflow 1: Disable GTPR register buffer transfer by overflow/underflow	R/W
b19	CPTADA	Overflow/Underflow Source GTADTRA Register Buffer Transfer Disable	0: Enable GTADTRA register buffer transfer by overflow/ underflow 1: Disable GTADTRA register buffer transfer by overflow/ underflow	R/W
b20	CPTADB	Overflow/Underflow Source GTADTRB Register Buffer Transfer Disable	0: Enable GTADTRB register buffer transfer by overflow/ underflow 1: Disable GTADTRB register buffer transfer by overflow/ underflow	R/W
b21	CPTDV	Overflow/Underflow Source GTDVU/GTDVD Register Buffer Transfer Disable	0: Enable GTDVU/GTDVD register buffer transfer by overflow/ underflow 1: Disable GTDVU/GTDVD register buffer transfer by overflow/ underflow	R/W
b23, b22	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b24	CP3DB	Complementary PWM Mode 3,4 Double Buffer Select*1	0: Disable double buffer function in complementary PWM mode 3, 4 1: Enable double buffer function in complementary PWM mode 3, 4	R/W
b25	CPBTD	Complementary PWM Mode Buffer Transfer Disable*1	0: Enable buffer transfer from the temporary register to the GTCCRC and GTPBR register 1: Disable buffer transfer from the temporary register to the GTCCRC and GTPBR register	R/W
b27, b26	OLTTA[1:0]	GTIOCnA Output Level Buffer Transfer Timing Select*1	• Triangle waves, complementary PWM mode b27 b26 0 0: No transfer 0 1: Transfer at crest 1 0: Transfer at trough 1 1: Transfer at both crest and trough • Sawtooth waves b27 b26 0 0: No transfer 0 1: Transfer at the end of period 1 0: Transfer by compare match of the GTCCRA register 1 1: Setting prohibited	R/W
b29, b28	OLTTB[1:0]	GTIOCnB Output Level Buffer Transfer Timing Select*1	• Triangle waves, complementary PWM mode b29 b28 0 0: No transfer 0 1: Transfer at crest 1 0: Transfer at trough 1 1: Transfer at both crest and trough • Sawtooth waves b29 b28 0 0: No transfer 0 1: Transfer at the end of period 1 0: Transfer by compare match of the GTCCRB register 1 1: Setting prohibited	R/W
b31, b30	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Not available in GPTW3, GPTW7.

The GTBER2 register makes settings for buffer operation.
Set the CP3DB bit and OLTTm[1:0] (m = A, B) bits when the GTCNT counter is stopped.

CCTCm Bit (Counter Clear Source GTCCRm Register Buffer Transfer Disable) (m = A, B)

This bit disables buffer transfer by counter clear using the GTCCRm, GTCCRn, and GTCCRx registers together (m = A, B; n = C, E; x = D, F).
This bit is effective when the GTBER.BD[0] bit is 0 (buffer operation enabled) and the buffer operation is selected by the GTBER.CCRm[1:0] bits with sawtooth-waves.
The setting is invalid in triangle waves or complementary PWM mode. Even if the CCTCm bit is 0, buffer transfer is not performed by counter clear.
If there is a conflict with the CMTCm bit setting, the CCTCm bit setting has priority.
The setting is invalid during the event count operation.

CCTPR Bit (Counter Clear Source GTPR Register Buffer Transfer Disable)

This bit disables buffer transfer by counter clear using the GTPR, GTPBR, and GTPDBR registers together.
This bit is effective when the GTBER.BD[1] bit is 0 (buffer operation enabled) and the buffer operation is selected by the GTBER.PR[1:0] bits with sawtooth-waves.
The setting is invalid in triangle waves or complementary PWM mode. Even if the CCTPR bit is 0, buffer transfer is not performed by counter clear.
The setting is invalid during the event count operation.

24.2.44 General PWM Timer Output Level Buffer Register (GTOLBR)

Address(es): GPTW0.GTOLBR 000C 20E4h, GPTW1.GTOLBR 000C 21E4h, GPTW2.GTOLBR 000C 22E4h, GPTW3.GTOLBR 000C 23E4h, GPTW4.GTOLBR 000C 24E4h, GPTW5.GTOLBR 000C 25E4h, GPTW6.GTOLBR 000C 26E4h, GPTW7.GTOLBR 000C 27E4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	GTIOBB[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	GTIOAB[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	GTIOAB[4:0]	GTIOA Buffer	GTIOA[4:0] Register Buffer	R/W
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b20 to b16	GTIOBB[4:0]	GTIOB Buffer	GTIOB[4:0] Register Buffer	R/W
b31 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTOLBR register is a buffer register for the GTIOR.GTIOA[4:0] bits and GTIOR.GTIOB[4:0] bits.

GTIOmB[4:0] Bits (GTIOm Buffer) (m = A, B)

These bits are buffer bits of GTIOR.GTIOm[4:0] bits.

These bits are transferred to the GTIOR.GTIOm[4:0] bits at the transfer timing selected by the GTBER2.OLTTm[1:0] (m = A, B) bits.

24.2.45 General PWM Timer Inter Channel Cooperation Input Capture Control Register (GTICCR)

Address(es): GPTW0.GTICCR 000C 20ECh, GPTW1.GTICCR 000C 21ECh, GPTW2.GTICCR 000C 22ECh, GPTW3.GTICCR 000C 23ECh, GPTW4.GTICCR 000C 24ECh, GPTW5.GTICCR 000C 25ECh, GPTW6.GTICCR 000C 26ECh, GPTW7.GTICCR 000C 27ECh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ICBGRP[1:0]		—	—	—	—	—	ICBCLK	ICBFPU	ICBFPO	ICBFF	ICBFE	ICBFD	ICBFC	ICBFB	ICBFA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ICAGRP[1:0]		—	—	—	—	—	ICACLK	ICAFPU	ICAFPO	ICAFF	ICAFE	ICAFD	ICAFC	ICAFB	ICAFA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ICAFA	Enable GTCCRA Register Compare Match/Input Capture as Source of Other Channel GTCCRA Input Capture	0: Disable GTCCRA register comparison match/input capture transfer, which is the GTCCRA input capture source of other channels 1: Enable GTCCRA register comparison match/input capture transfer, which is the GTCCRA input capture source of other channels	R/W
b1	ICAFB	Enable GTCCRB Register Compare Match/Input Capture as Source of Other Channel GTCCRA Input Capture	0: Disable GTCCRB register comparison match/input capture transfer, which is the GTCCRA input capture source of other channels 1: Enable GTCCRB register comparison match/input capture transfer, which is the GTCCRA input capture source of other channels	R/W
b2	ICAFC	Enable GTCCRC Register Compare Match as Source of Other Channel GTCCRA Input Capture	0: Disable GTCCRC register comparison match, which is the GTCCRA input capture source of other channels 1: Enable GTCCRC register comparison match, which is the GTCCRA input capture source of other channels	R/W
b3	ICAFD	Enable GTCCRD Register Compare Match as Source of Other Channel GTCCRA Input Capture	0: Disable GTCCRD register comparison match, which is the GTCCRA input capture source of other channels 1: Enable GTCCRD register comparison match, which is the GTCCRA input capture source of other channels	R/W
b4	ICAFE	Enable GTCCRE Register Compare Match as Source of Other Channel GTCCRA Input Capture	0: Disable GTCCRE register comparison match, which is the GTCCRA input capture source of other channels 1: Enable GTCCRE register comparison match, which is the GTCCRA input capture source of other channels	R/W
b5	ICAFF	Enable GTCCRF Register Compare Match as Source of Other Channel GTCCRA Input Capture	0: Disable GTCCRF register comparison match, which is the GTCCRA input capture source of other channels 1: Enable GTCCRF register comparison match, which is the GTCCRA input capture source of other channels	R/W
b6	ICAFPO	Enable Overflow as Source of Other Channel GTCCRA Input Capture	0: Disable overflow for sawtooth wave and trough for triangular wave and complementary PWM mode, which are the GTCCRA input capture source of other channels 1: Enable overflow for sawtooth wave and trough for triangular wave and complementary PWM mode, which are the GTCCRA input capture source of other channels	R/W
b7	ICAFPU	Enable Underflow as Source of Other Channel GTCCRA Input Capture	0: Disable underflow for sawtooth wave and trough for triangular wave and complementary PWM mode, which are the GTCCRA input capture source of other channels 1: Enable underflow for sawtooth wave and trough for triangular wave and complementary PWM mode, which are the GTCCRA input capture source of other channels	R/W

Bit	Symbol	Bit Name	Description	R/W
b8	ICACLK	Enable Count Clock as Source of Other Channel GTCCRA Input Capture	0: Disable count clock, which are the GTCCRA input capture source of other channels 1: Enable count clock, which are the GTCCRA input capture source of other channels	R/W
b13 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15, b14	ICAGRP[1:0]	GTCCRA Input Capture Group Select	b15b14 0 0: Group A 0 1: Group B 1 0: Group D 1 1: Group D	R/W
b16	ICBFA	Enable GTCCRA Register Compare Match/Input Capture as Source of Other Channel GTCCRB Input Capture	0: Disable GTCCRA register comparison match/input capture transfer, which is the GTCCRB input capture source of other channels 1: Enable GTCCRA register comparison match/input capture transfer, which is the GTCCRB input capture source of other channels	R/W
b17	ICBFB	Enable GTCCRB Register Compare Match/Input Capture as Source of Other Channel GTCCRB Input Capture	0: Disable GTCCRB register comparison match/input capture transfer, which is the GTCCRB input capture source of other channels 1: Enable GTCCRB register comparison match/input capture transfer, which is the GTCCRB input capture source of other channels	R/W
b18	ICBFC	Enable GTCCRC Register Compare Match as Source of Other Channel GTCCRB Input Capture	0: Disable GTCCRC register comparison match, which is the GTCCRB input capture source of other channels 1: Enable GTCCRC register comparison match, which is the GTCCRB input capture source of other channels	R/W
b19	ICBFD	Enable GTCCRD Register Compare Match as Source of Other Channel GTCCRB Input Capture	0: Disable GTCCRD register comparison match, which is the GTCCRB input capture source of other channels 1: Enable GTCCRD register comparison match, which is the GTCCRB input capture source of other channels	R/W
b20	ICBFE	Enable GTCCRE Register Compare Match as Source of Other Channel GTCCRB Input Capture	0: Disable GTCCRE register comparison match, which is the GTCCRB input capture source of other channels 1: Enable GTCCRE register comparison match, which is the GTCCRB input capture source of other channels	R/W
b21	ICBFF	Enable GTCCRF Register Compare Match as Source of Other Channel GTCCRB Input Capture	0: Disable GTCCRF register comparison match, which is the GTCCRB input capture source of other channels 1: Enable GTCCRF register comparison match, which is the GTCCRB input capture source of other channels	R/W
b22	ICBFPO	Enable Overflow as Source of Other Channel GTCCRB Input Capture	0: Disable overflow for sawtooth wave and trough for triangular wave and complementary PWM mode, which are the GTCCRB input capture source of other channels 1: Enable overflow for sawtooth wave and trough for triangular wave and complementary PWM mode, which are the GTCCRB input capture source of other channels	R/W
b23	ICBFPU	Enable Underflow as Source of Other Channel GTCCRB Input Capture	0: Disable underflow for sawtooth wave and trough for triangular wave and complementary PWM mode, which are the GTCCRB input capture source of other channels 1: Enable underflow for sawtooth wave and trough for triangular wave and complementary PWM mode, which are the GTCCRB input capture source of other channels	R/W
b24	ICBCLK	Enable Count Clock as Source of Other Channel GTCCRB Input Capture	0: Disable count clock, which are the GTCCRB input capture source of other channels 1: Enable count clock, which are the GTCCRB input capture source of other channels	R/W
b29 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31, b30	ICBGRP[1:0]	GTCCRB Input Capture Group Select	b31b30 0 0: Group A 0 1: Group B 1 0: Group D 1 1: Group D	R/W

The GTICCR register is a register that controls input capture by inter channel link.
For channels that perform input capture in cooperation, the input capture factor set in the GTICASR and GTICASR registers is invalid.

ICAFm Bit (Enable GTCCRm Register Compare Match/Input Capture as Source of Other Channel GTCCRA Input Capture) (m = A, B)

Enables/disables the use of compare match/input capture of GTCCRm register as the input capture source of other channel's GTCCRA register.

ICAFn Bit (Enable GTCCRn Register Compare Match as Source of Other Channel GTCCRA Input Capture) (n = C, D, E, F)

Enables/disables the use of compare match of GTCCRn register as the input capture source of other channel's GTCCRA register.

ICAFPO Bit (Enable Overflow as Source of Other Channel GTCCRA Input Capture)

Enable/disable to use the overflow of sawtooth-waves, the crest of triangle-waves or complementary PWM mode as the input capture source of other channel's GTCCRA register.

ICAFPU Bit (Enable Underflow as Source of Other Channel GTCCRA Input Capture)

Enable/disable to use the underflow of sawtooth-waves, the trough of triangle-waves or complementary PWM mode as the input capture source of other channel's GTCCRA register.

ICACLK Bit (Enable Count Clock as Source of Other Channel GTCCRA Input Capture)

Enable/disable to use count clock as the input capture source of other channel's GTCCRA register.

ICAGRP[1:0] Bits (GTCCRA Input Capture Group Select)

Select the group of input capture by inter channel cooperation for GTCCRA register.
For channels that accept input capture of the GTCCRA register due to input capture sources from other channels, set the GTICASR.ASOC bit to 1 and select the group of inter channel cooperation with the ICAGRP[1:0] bits.

ICBFm Bit (Enable GTCCRm Register Compare Match/Input Capture as Source of Other Channel GTCCRB Input Capture) (m = A, B)

Enables/disables the use of compare match/input capture of GTCCRm register as the input capture source of other channel's GTCCRB register.

ICBFn Bit (Enable GTCCRn Register Compare Match as Source of Other Channel GTCCRB Input Capture) (n = C, D, E, F)

Enables/disables the use of compare match of GTCCRn register as the input capture source of other channel's GTCCRB register.

ICBFPO Bit (Enable Overflow as Source of Other Channel GTCCRB Input Capture)

Enable/disable to use the overflow of sawtooth-waves, the crest of triangle-waves or complementary PWM mode as the input capture source of other channel's GTCCRB register.

ICBFPU Bit (Enable Underflow as Source of Other Channel GTCCRB Input Capture)

Enable/disable to use the underflow of sawtooth-waves, the trough of triangle-waves or complementary PWM mode as the input capture source of other channel's GTCCRB register.

ICBCLK Bit (Enable Count Clock as Source of Other Channel GTCCRB Input Capture)

Enable/disable to use count clock as the input capture source of other channel's GTCCRB register.

ICBGRP[1:0] Bits (GTCCRB Input Capture Group Select)

Select the group of input capture by inter channel cooperation for GTCCRB register.

For channels that accept input capture of the GTCCRB register due to input capture sources from other channels, set the GTICBSR.BSOC bit to 1 and select the group of inter channel cooperation with the ICBGRP[1:0] bits.

24.2.46 Output Phase Switching Control Register (OPSCR)

Address(es): GPTW.OPSCR 000C 2B00h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	NFCS[1:0]		NFEN	—	—	GODF	GRP[1:0]		—	—	ALIGN	RV	INV	N	P	FB
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	EN	—	W	V	U	—	WF	VF	UF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	UF	Input Phase Software Setting	These bits set the input phase from software settings. Setting these bits is valid when FB = 1.	R/W
b1	VF			R/W
b2	WF			R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	U	Input U-Phase Monitor	This bit monitors the state of the input phase. • FB = 0 External input that are synchronized by PCLKC • FB = 1 The software settings (UF, VF, WF bit)	R
b5	V	Input V-Phase Monitor		R
b6	W	Input W-Phase Monitor		R
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b8	EN	Output Phase Enable	0: Do not output (Hi-Z external pin) 1: Output*1	R/W
b15-b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	FB	External Feedback Signal Enable	This bit selects the input phase from software settings and external input. 0: The external input 1: The software setting (UF, VF, WF bit)	R/W
b17	P	Positive-Phase Output (P) Control	0: Level signal output 1: PWM signal output (PWM of GPTW0)	R/W
b18	N	Negative-Phase Output (N) Control	0: Level signal output 1: PWM signal output (PWM of GPTW0)	R/W
b19	INV	Output Phase Invert Control	0: Positive logic (active-high) output 1: Negative logic (active-low) output	R/W
b20	RV	Output Phase Rotation Direction Reversal Control	0: Positive rotation 1: Reverse rotation	R/W
b21	ALIGN	Input Phase Alignment	0: Input phase aligned to PCLKC 1: Input phase aligned to the falling edge of PWM	R/W
b23, b22	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b25, b24	GRP[1:0]	Output Disabled Source Selection	b25b24 0 0: Group A output disable source 0 1: Group B output disable source 1 0: Group C output disable source 1 1: Group D output disable source	R/W
b26	GODF	Group Output Disable Function	0: This bit function is ignored 1: When the signal value of the source selected by the GRP[1:0] bits becomes High, the EN bit is cleared	R/W
b28, b27	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b29	NFEN	External Input Noise Filter Enable	0: The noise filter for the external input is disabled. 1: The noise filter for the external input is enabled.	R/W

Bit	Symbol	Bit Name	Description	R/W
b31, b30	NFCS[1:0]	External Input Noise Filter Clock Select	b31b30 0 0: PCLKC/1 0 1: PCLKC/4 1 0: PCLKC/16 1 1: PCLKC/64	R/W

Note 1. When the GODF bit is 1 and the signal value of the source selected by the GRP[1:0] bits becomes High, the EN bit is set to 0.

The OPSCR register sets the output of the signal waveform required for brushless DC motor control.

UF, VF, WF Bits (Input Phase Software Setting)

The UF, VF, and WF bits set the input phase from the software settings. When the FB bit is 1, these bits are valid. The set value of the UF, VF, and WF bit takes the place of the U, V, W external input.

U, V, W Bits (Input m-Phase Monitor) (m = U, V, W)

When the FB bit is 0, external inputs that are synchronized by PCLKC are monitored by these bits. When the FB bit is 1, the U, V, and W bits can read the UF, VF, and WF bits.

EN Bit (Output Phase Enable)

The EN bit controls the output enable signal of output phase (positive phase/negative phase).
When the EN bit is 1, the signal waveform is output.
When the EN bit is 0, first set FB, VF, and WF (software setting is selected), P, N, INV, RV, ALIGN, GRP[1:0], GODF, NFEN, and NFCS bits. Then, set this bit to 1. The EN bit should be set when output disable request doesn't occur from POEG. Also when GODF is 1 and the signal value selected in the GRP[1:0] bits are high, the EN bit is set to 0. And 1 by software to be written, the EN bit remains at 0.
For the return, after clearing the output disable request by software, set the EN bit to 1.

- EN bit priority order (conflict)

When 1 write by software and set to 0 by the output disable request has been conflicting for EN bit, set to 0 by the Output Disable Request is activated.

FB Bit (External Feedback Signal Enable)

The FB bit selects the input phase from the software settings (UF, VF, and WF bits) and external input such as a Hall element.

P Bit (Positive-Phase Output (P) Control)

The P bit selects one of the level signal output or PWM signal output for the positive-phase output (GTOUUP, GTOVUP and GTOWUP pins).

N Bit (Negative-Phase Output (N) Control)

The P bit selects one of the level signal output or PWM signal output for the negative-phase output (GTOULO, GTOVLO and GTOWLO pins).

INV Bit (Output Phase Invert Control)

The INV bit selects one of the positive logic (active-high) output or negative logic (active-low) output for the output phase.

RV Bit (Output Phase Rotation Direction Reversal Control)

The RV bit reverses the direction of rotation of the motor by inverting the input phase.

ALIGN Bit (Input Phase Alignment)

The ALIGN bit selects the PCLKC or PWM for the sampling of the input phase (input phase is specified in the FB bit).

When ALIGN bit is 0, input phase is aligned to PCLKC.

Note: When the chopping is performed, there are cases where the PWM width of output is shorter than the width of the PWM used to chop just before or after switching of output phase, depending on the phase difference between the phase output switch timing and the phase of PWM.

When ALIGN bit is 1, input phase is aligned with the falling edge of PWM.

GRP[1:0] Bits (Output Disabled Source Selection)

The GRP[1:0] bits select the output disable source.

When the GODF bit is 0, set the GRP[1:0] bits.

GODF Bit (Group Output Disable Function)

When the GODF bit is 1 and the signal value selected by the GRP[1:0] bits are high, the EN bit is set to 0.

When the GODF bit is 0, this bit is ignored.

Set the GODF bit while there is not output disable request from POEG.

NFEN Bit (External Input Noise Filter Enable)

The NFEN bit disables or enables the external input noise filter.

Note: Set this bit during the EN bit is 0 to avoid generation of unintentional internal edge caused by this bit switching.

NFCS[1:0] Bits (External Input Noise Filter Clock Select)

The NFCS[1:0] bits select the clock for the external input noise filter.

1. Set the NFCS[1:0] bits.
2. Wait for 2 cycles.
3. Set the EN bit to 1.

24.3 Operation

24.3.1 Basic Operation

The timer in each channel performs cycle count operation by count clock or hardware source. The GTCNT counter performs up-counting, or down-counting. The timer period is controlled by the GTPR or GTCCRm (m = A to F) registers. When the GTCNT counter value matches the values for GTCCRA or GTCCRB registers, the GTIOCnA or GTIOCnB pin outputs (n = 0 to 7) can be changed respectively. Also, the GTCCRA or GTCCRB registers can be used as the input capture registers by hardware source. The GTCCRC and GTCCRD registers can be performed as buffer registers for GTCCRA register, and the GTCCRE and GTCCRF registers can be performed as buffer registers for GTCCRB register.

24.3.1.1 Counter Operation

(1) Count Start/Count Stop

The counter in each channel starts counting operation when the corresponding GTCR.CST bit is set to 1, and stops counting when the bit is set to 0.
CST bit value can be changed by the following sources:

- Writing to the GTCR register
- Writing 1 to the bit corresponding to the channel number for the GTSTR register while the GTSSR.CSTRT bit is 1
- Writing 1 to the bit corresponding to the channel number for the GTSTP register while the GTPSR.CSTOP bit is 1
- Hardware source specified by the GTSSR register
- Hardware source specified by the GTPSR register
- Completion of the cycle count operation while the GTPC.ASTP bit is 1.

(2) Cycle Count Operation (In Up-Counting by the Count Clock)

The counter in each channel starts up-counting when the corresponding CST bit is set to 1 while the GTUPSR register and the GTDNSR register are 0000 0000h. When the value of the GTCNT counter changes from that of the GTPR register to 0000 0000h (an overflow) or the GTCCRm (m = A to F) value selected by the GTCSR.CSCMSC[2:0] bits to 0000 0000h in sawtooth-wave PWM mode 2 while the GTINTAD.GTINTPR[0] bit is 1, a GTCIV interrupt request is also issued. After GTCNT counter overflows, up-counting is resumed from 0000 0000h.
Figure 24.4 shows an example of cycle count operation in up-counting by count clock.

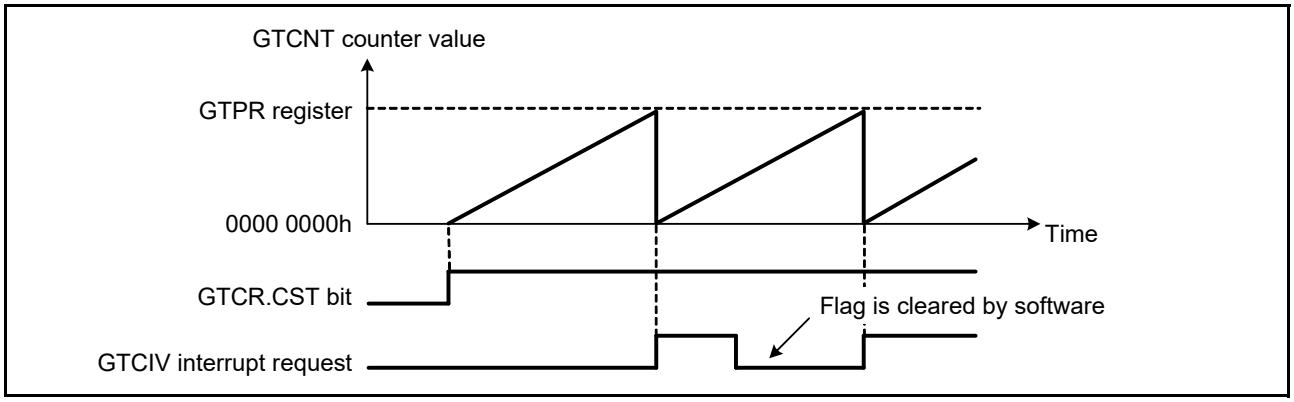


Figure 24.4 Example of Cycle Count Operation (In Up-Counting by the Count Clock)

Figure 24.5 shows an example for setting cycle count operation in up-counting.

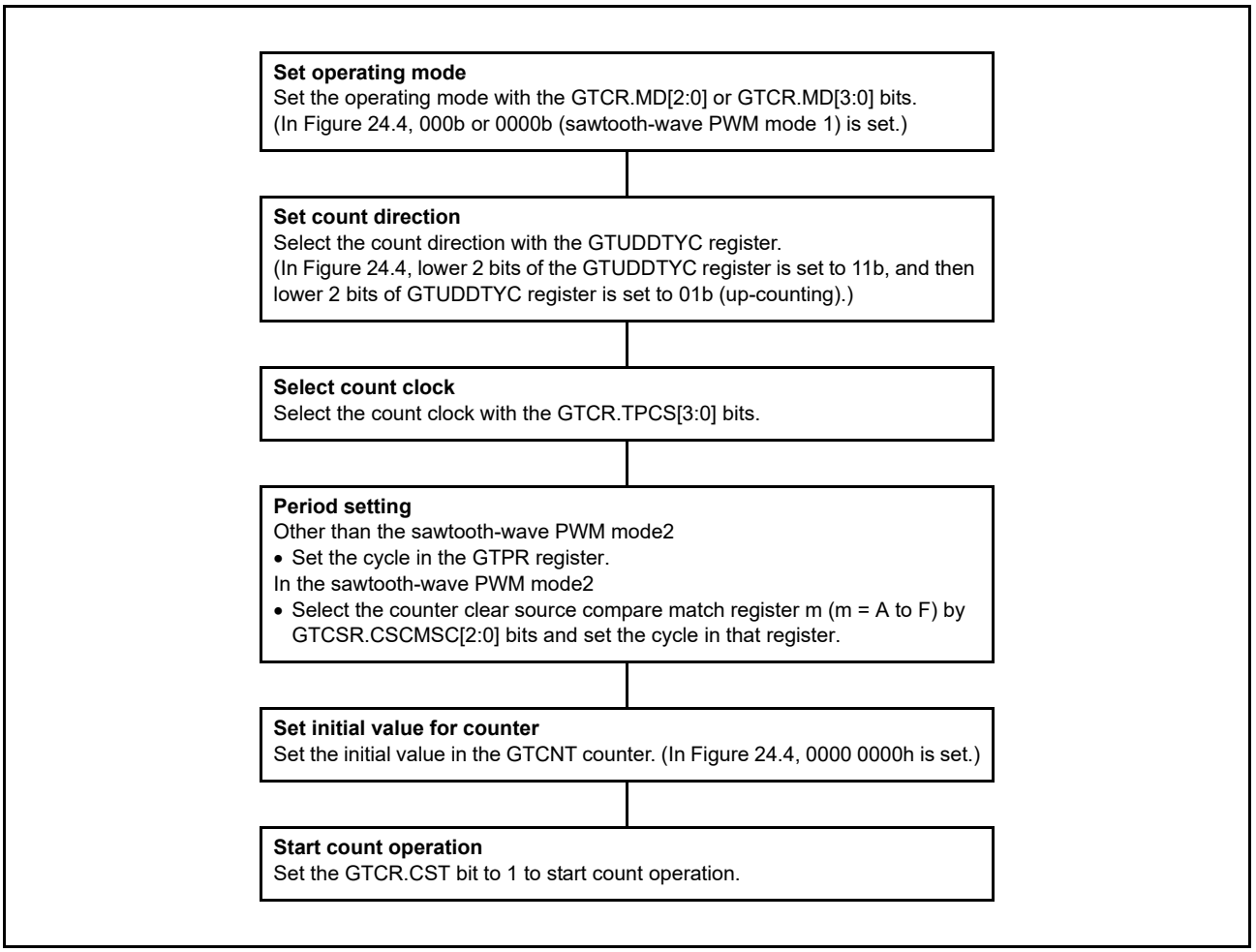


Figure 24.5 Example for Setting Cycle Count Operation (In Up-Counting by the Count Clock)

(3) Cycle Count Operation (In Down-Counting by the Count Clock)

The counter in each channel can start down-counting when the corresponding GTUDDTYC.UD bit is set in a state of the GTUPSR register and the GTDNSR register as 0000 0000h. When the value of the GTCNT counter changes from 0000 0000h to that of the GTPR register (an underflow) while the GTINTAD.GTINTPR[1] bit is 1, a GTCIU interrupt request is also issued. After the GTCNT counter underflows, down-counting is resumed from the GTPR register value.
Figure 24.6 shows an example of cycle count operation in down-counting by count clock.

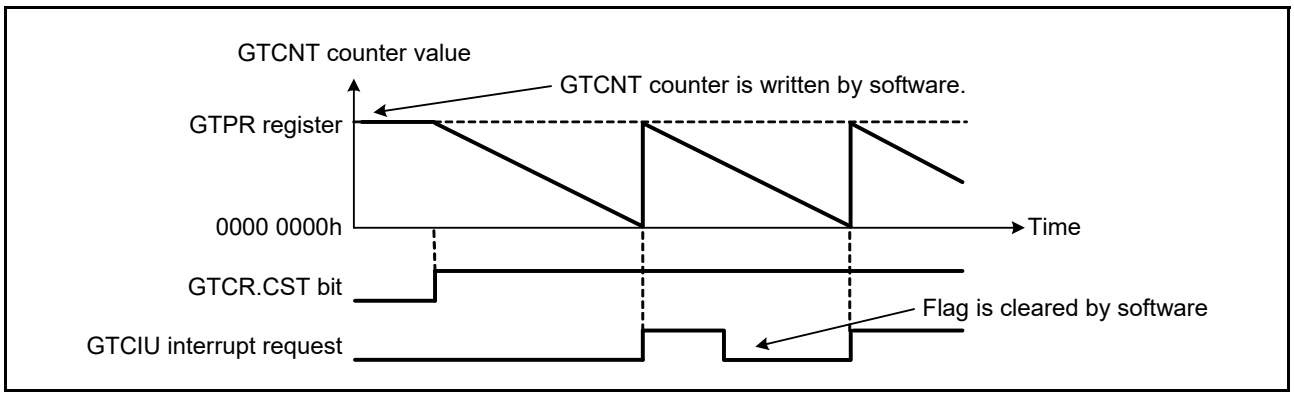


Figure 24.6 Example of Cycle Count Operation (In Down-Counting by the Count Clock)

Figure 24.7 shows an example for setting cycle count operation in down-counting by count clock.

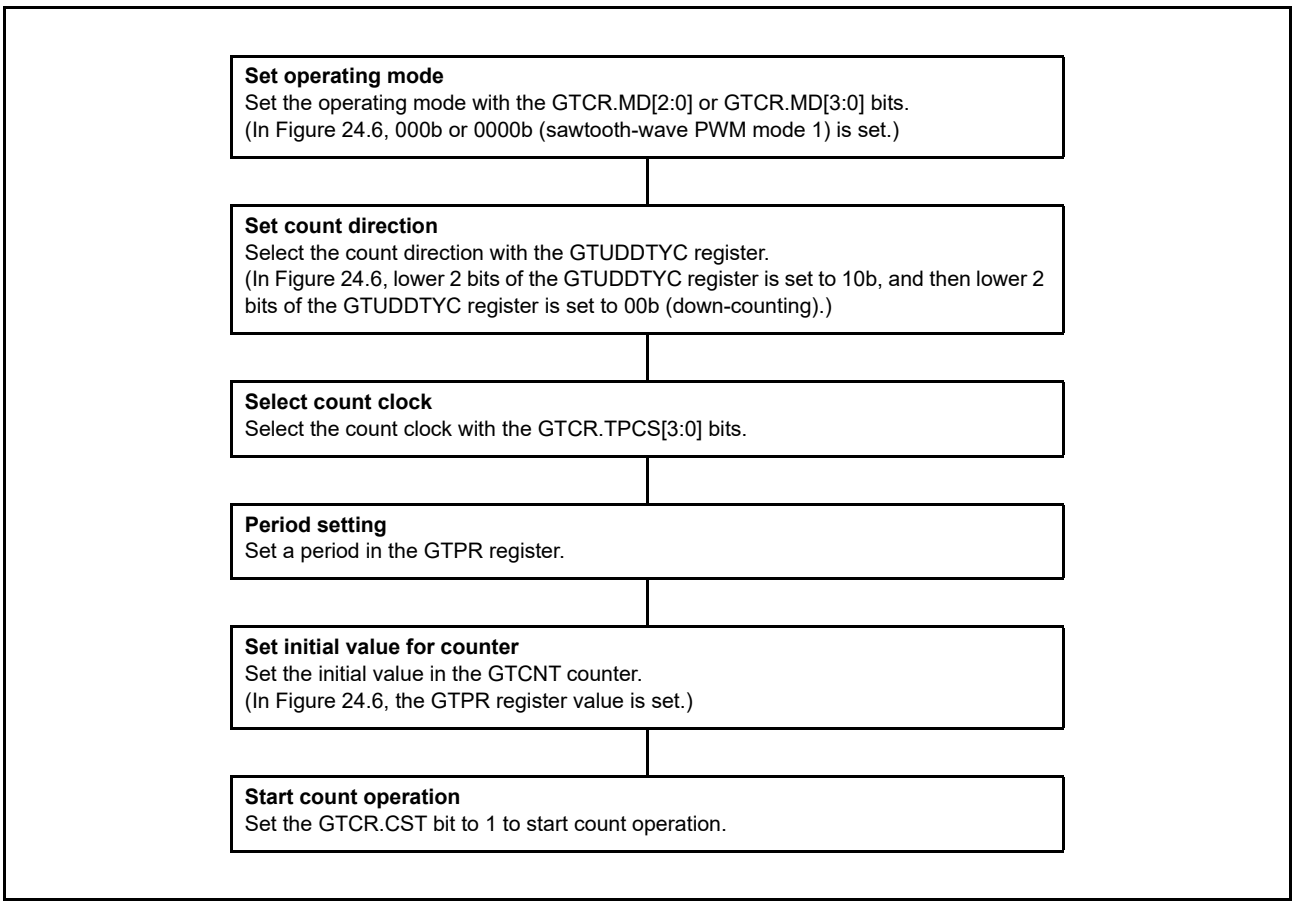


Figure 24.7 Example for Setting Cycle Count Operation (In Down-Counting by the Count Clock)

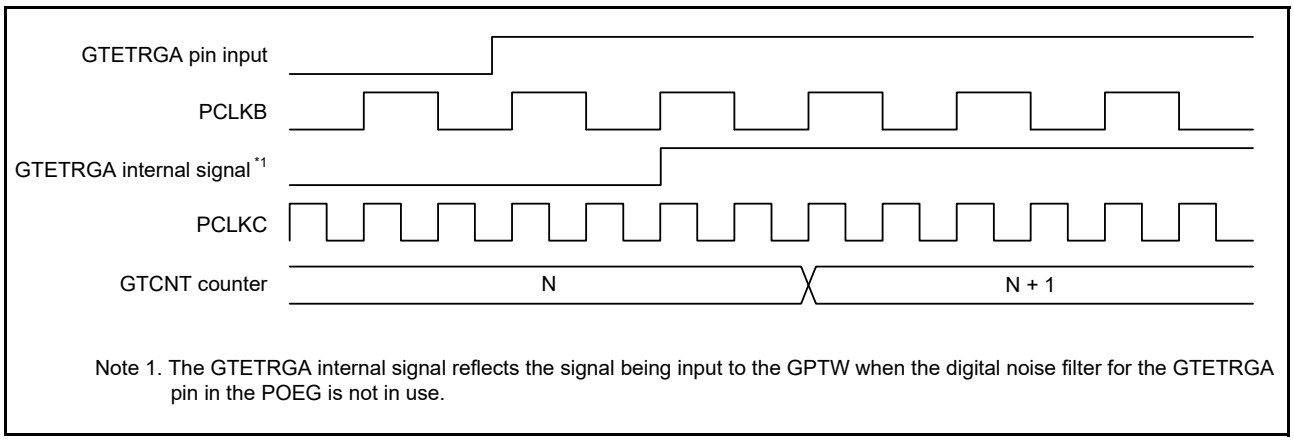
(4) Event Count Operation (In Up-Counting by Hardware Source)

The counter in each channel can start up-counting by hardware source by setting the GTUPSR register. When the GTUPSR register is set, the count clock selected by the GTCR.TPCS[3:0] bits and the count direction set by the GTUDDTYC.UD bit are invalid. If the hardware source in up-counting and the hardware source in down-counting are generated at the same time, the GTCNT counter value will not change.

Operation due to overflow in up-counting by hardware source is the same as the cycle count operation by the count clock.

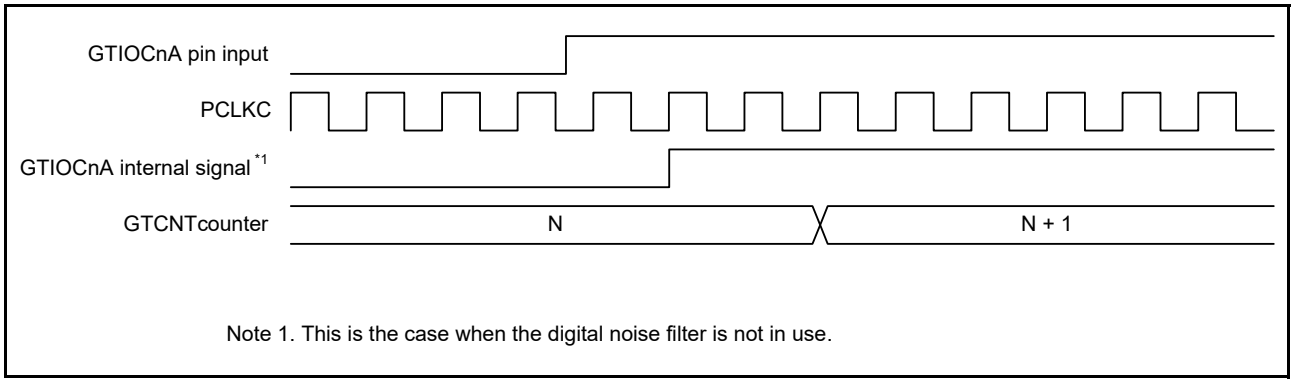
To perform up-counting by hardware source, start count operation by setting the GTCR.CST bit to 1. As the starting the count operation is synchronized with the count clock selected by TPCS[3:0] bits, set the CST bit to 1, and wait until the first cycle of the count clock elapses to start up-counting operation. To start up-counting after one PCLKC from setting the CST bit to 1, set the TPCS[3:0] bits to 0000b.

Figure 24.8 and Figure 24.9 show examples of up-counting operation by hardware source (edge of GTETRGA pin input and the rising edge of GTIOCnA pin input).



Note 1. The GTETRGA internal signal reflects the signal being input to the GPTW when the digital noise filter for the GTETRGA pin in the POEG is not in use.

Figure 24.8 Example of Event Count Operation (Up-Counting of Rising Edges of the Input on the GTETRGA Pin)



Note 1. This is the case when the digital noise filter is not in use.

Figure 24.9 Example of Event Count Operation (Up-Counting of Rising Edges of the Input on the GTIOCnA Pin)

Figure 24.10 shows an example of event counting by a hardware source (ELC event input).
This is an example when the ELC selectively outputs the event signal output to the ELC by the compare match to the GPTW0.GTCCRA register as the event factor A to the GPTW, and performs the event count operation of the GPTW1.GTCNT counter by the signal.
The GPTW0 compare match A signal synchronized with PCLKC is synchronized with PCLKB in ELC, and GPTW event factor A is output after 1 clock with PCLKB.

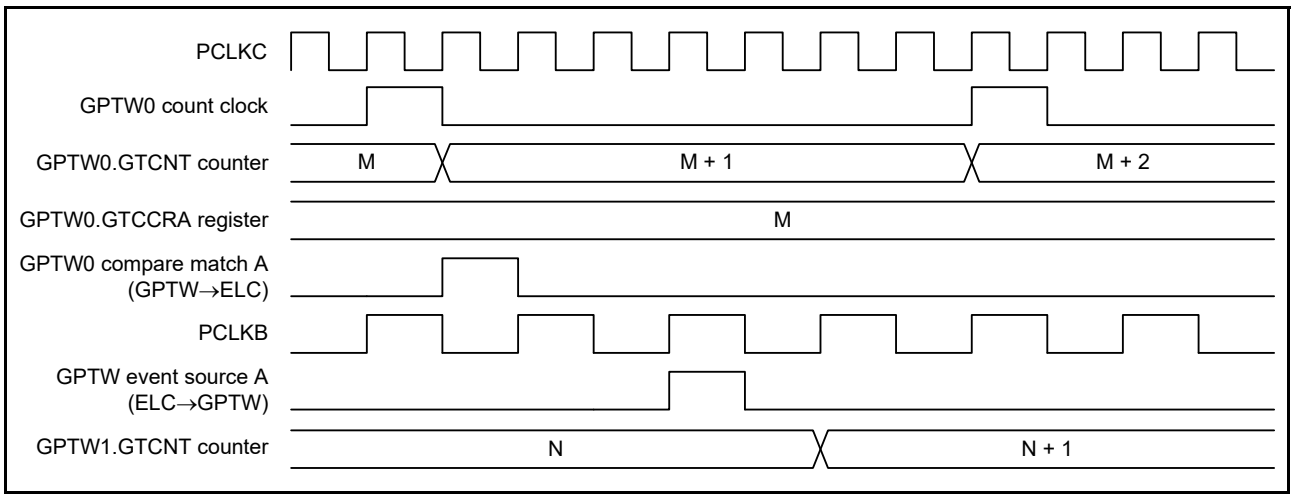


Figure 24.10 Example of Event Count Operation (Up-Counting the Number of Event Signals Input from the ELCA)

Figure 24.11 shows an example for setting cycle count operation in up-counting by hardware source.

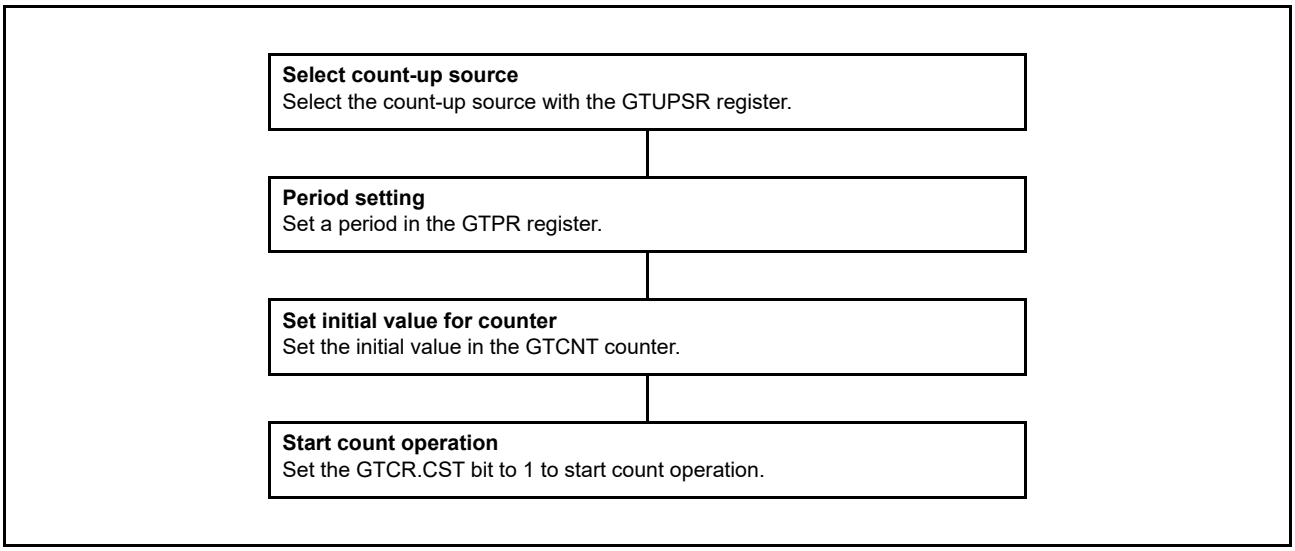


Figure 24.11 Example for Setting Event Count Operation (In Up-Counting by Hardware Source)

(5) Event Count Operation (In Down-Counting by Hardware Source)

The counter in each channel can start down-counting by hardware source by setting the GTDNSR register. When the GTDNSR register is set, the count clock selected by the GTCR.TPCS[3:0] bits and the count direction set by the GTUDDTYC.UD bit are invalid. If the hardware source in up-counting and the hardware source in down-counting are generated at the same time, the GTCNT counter value will not change.

Operation due to underflow in down-counting by hardware source is the same as the cycle count operation by the count clock.

To perform down-counting by hardware source, start count operation by setting the GTCR.CST bit to 1. As the starting the count operation is synchronized with the count clock selected by TPCS[3:0] bits, set the CST bit to 1, and wait until the first cycle of the count clock elapses to start down-counting operation. To start down-counting after one PCLKC from setting the CST bit to 1, set the TPCS[3:0] bits to 0000b.

Figure 24.12 shows an example of down-counting operation by hardware source.

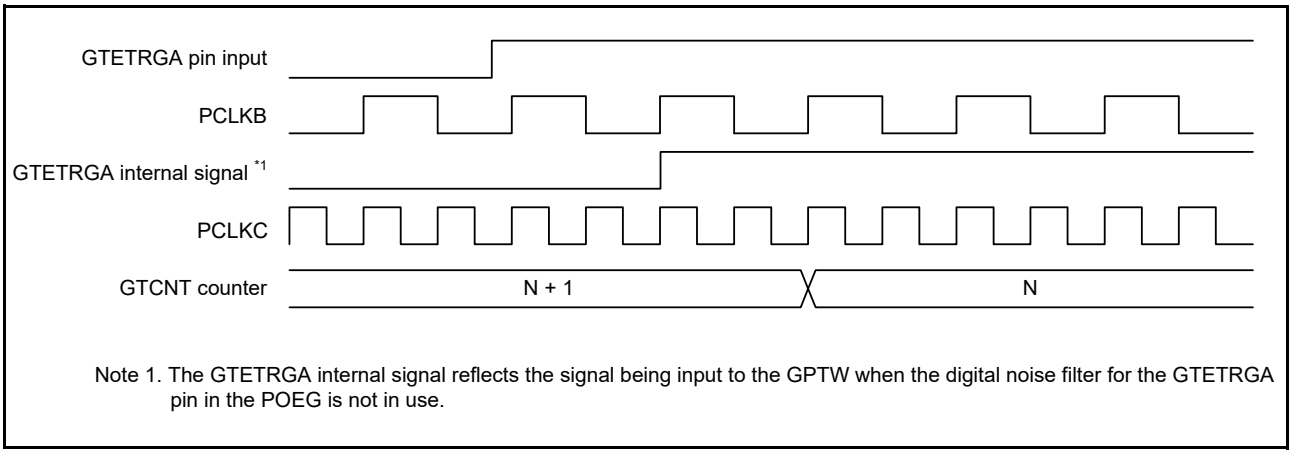


Figure 24.12 Example of Event Count Operation (Down-Counting of Rising Edges of the Input on the GTETRGA Pin)

Figure 24.13 shows an example for setting cycle count operation in down-counting by hardware source.

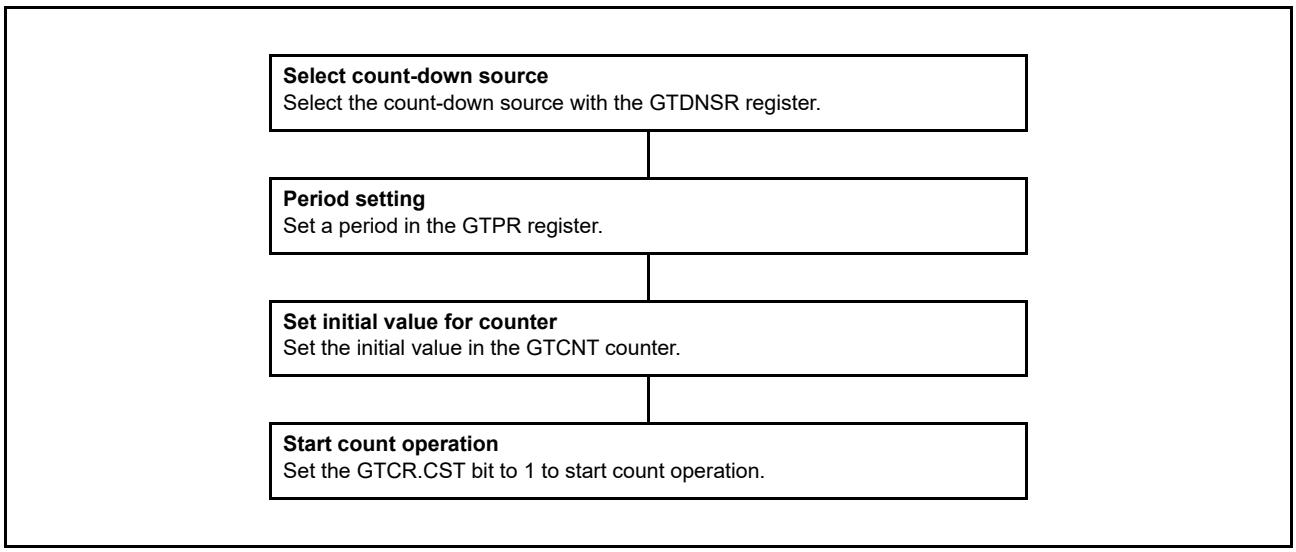


Figure 24.13 Example for Setting Event Count Operation (In Down-Counting by Hardware Source)

(6) Counter Clearing Operation

The counter in each channel can be cleared by the following sources:

- Writing to the GTCNT counter
- Writing 1 to the bit corresponding to the channel number for the GTCLR register while the GTCSR.CCLR bit is 1
- Hardware source specified by the GTCSR register

Write access during counting (when CST = 1) is disabled.
GTCLR register can be written to and cleared by hardware source whether the count operation is performed (GTCR.CST bit = 1) or is stopped (CST bit = 0). When the count direction flag is set as decrement (GTST.TUCF flag = 0) in sawtooth-wave mode (except sawtooth-wave PWM mode 2) selected with GTCR.MD[2:0] or GTCR.MD[3:0] bits, the GTCNT counter value for writing to the GTCLR register and for clearing by hardware source becomes the GTPR register value whether the counter is in operation or is stopped. The GTCNT counter value becomes 0000 0000h for other settings.
When the event count operation is set (at least one of bits for the GTUPSR or the GTDNSR register is set as 1), writing to the GTCLR register and clearing by hardware source are performed right after the clear source is generated (the operation is based on PCLKC). For other settings, clearing operation is synchronized with the count clock selected with the GTCR.TPCS[3:0] bits.

24.3.1.2 Waveform Output by Compare Match

Compare match refers to when the GTCNT counter value matches the GTCCRA or GTCCRB register value. The output of the corresponding GTIOCnA or GTIOCnB pin (n = 0 to 7) can be set to be driven low, high, or toggled in synchronization with the counter clock (including in the case of event counting) after the match.
In addition, the GTIOCnA or GTIOCnB pin output can be driven low, high, or toggled at the “end of the cycle” which is determined by the GTPR register or GTCCRm selected as a counter clear source by the GTCSR.CSCMSC[2:0] bits in sawtooth-wave PWM mode 2 (n = 0 to 7, m = A to F). The end of the cycle is as follows:

- For sawtooth waves (except sawtooth-wave PWM mode 2) in up-counting: When the GTCNT counter value changes from the GTPR register value to 0000 0000h (overflow)
- For sawtooth waves (except sawtooth-wave PWM mode 2) in down-counting: When the GTCNT counter value changes from 0000 0000h to the GTPR register value (underflow)
- For sawtooth-wave PWM mode 2 and GTCCRm register (m = A to F) selected as a counter clear source by the GTCSR.CSCMSC[2:0] bits. When the GTCNT counter value changes from the GTCCRm register value to 0000 0000h
- For sawtooth waves with the GTCNT counter clear
- For triangle waves: When the GTCNT counter value changes from 0000 0000h to 0000 0001h (trough)

(1) Low Output and High Output

Figure 24.14 shows an example of low output and high output operation by a compare match of the GTCNT counter and GTCCRA register and of the GTCNT counter and GTCCRB register.
In this example, the GPTW0.GTCNT counter performs up-counting, and settings have been made so that high is output from the GTIOC0A pin by a GPTW0.GTCCRA register compare match, and low is output from the GTIOC0B pin by a GPTW0.GTCCRB register compare match. The pin level does not change when the specified level and pin level match.

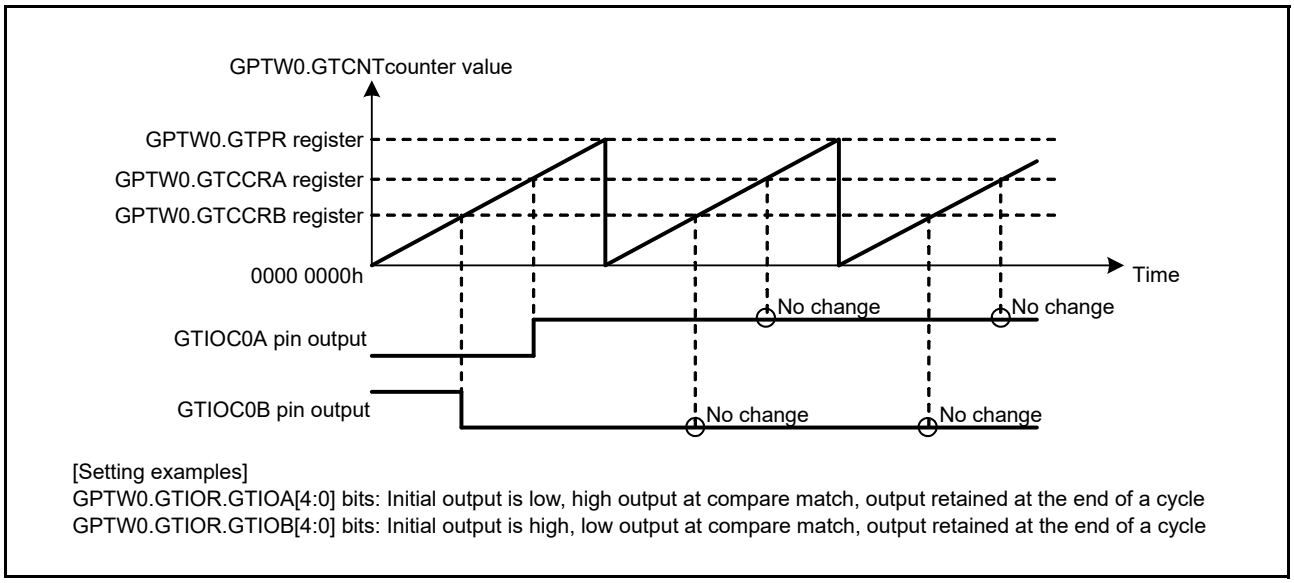
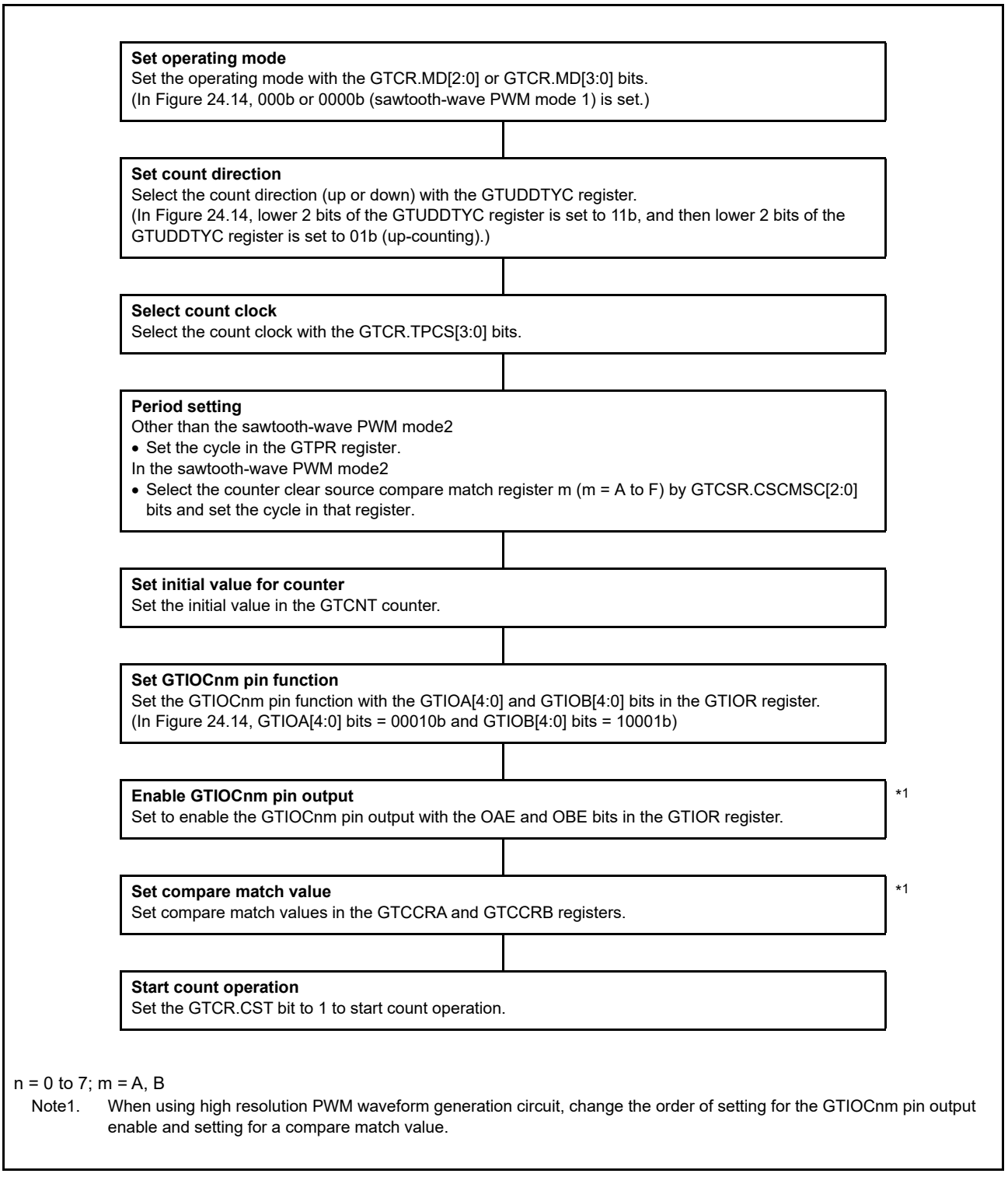


Figure 24.14 Example of Low Output and High Output Operation

Figure 24.15 shows an example for setting Low/High output operation.



n = 0 to 7; m = A, B

Note1. When using high resolution PWM waveform generation circuit, change the order of setting for the GTIOCnm pin output enable and setting for a compare match value.

Figure 24.15 Example for Setting Low/High Output Operation

(2) Toggled Output

Figure 24.16 and Figure 24.17 show examples of toggle output operations by compare match between the GPTW0.GTCNT counter and the GTCCRA and GTCCRB registers.

In Figure 24.16, the GPTW0.GTCNT counter performs up-counting, and settings have been made so that the GTIOC0A pin output by a GPTW0.GTCCRA register compare match and GTIOC0B pin output by a GPTW0.GTCCRB register compare match are toggled.

In Figure 24.17, the GPTW0.GTCNT counter performs up-counting, and settings have been made so that the GTIOC0A pin output is toggled by a compare match of GPTW0.GTCCRA register and the GTIOC0B pin output is toggled at the end of the cycle.

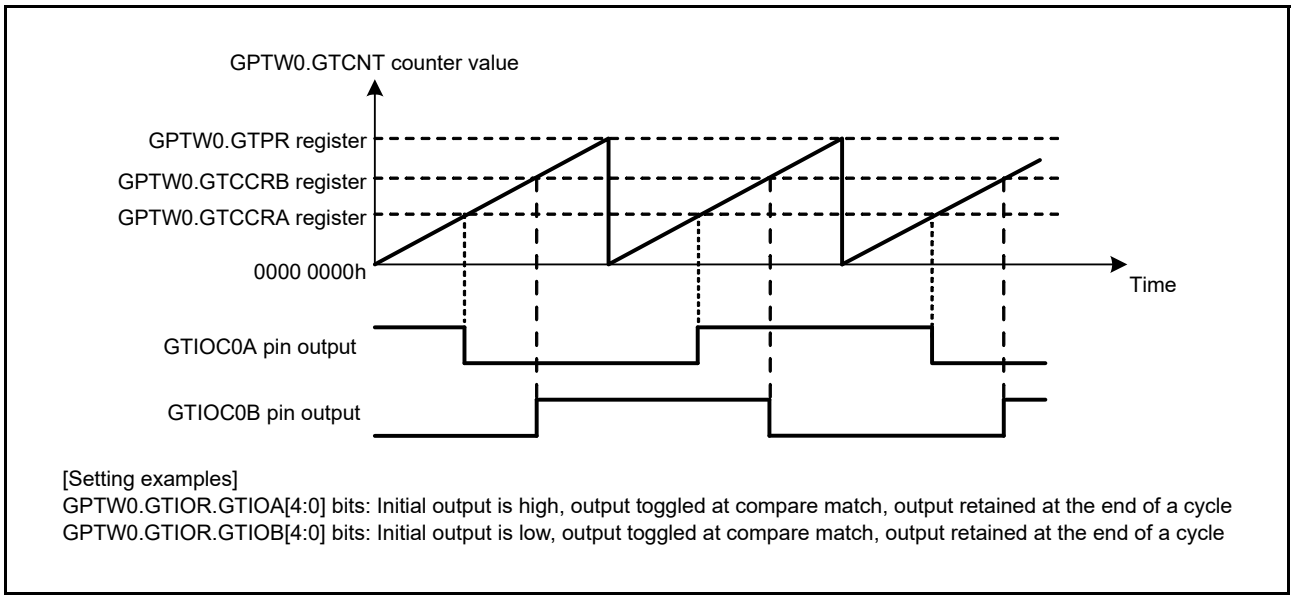


Figure 24.16 Example of Toggled Output Operation (1)

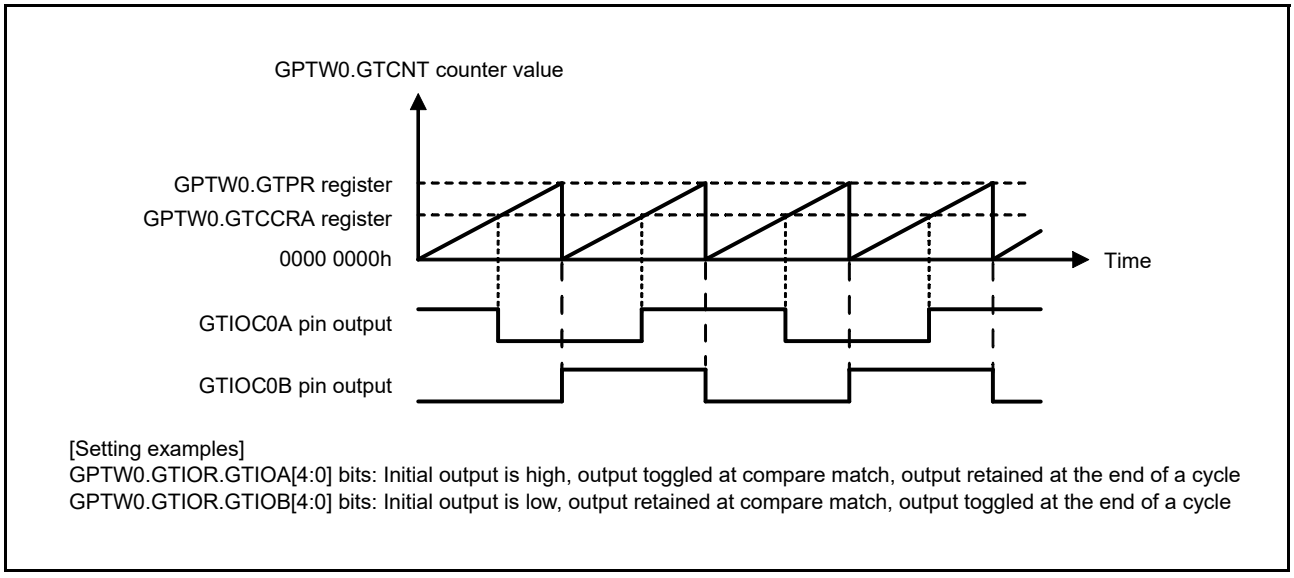
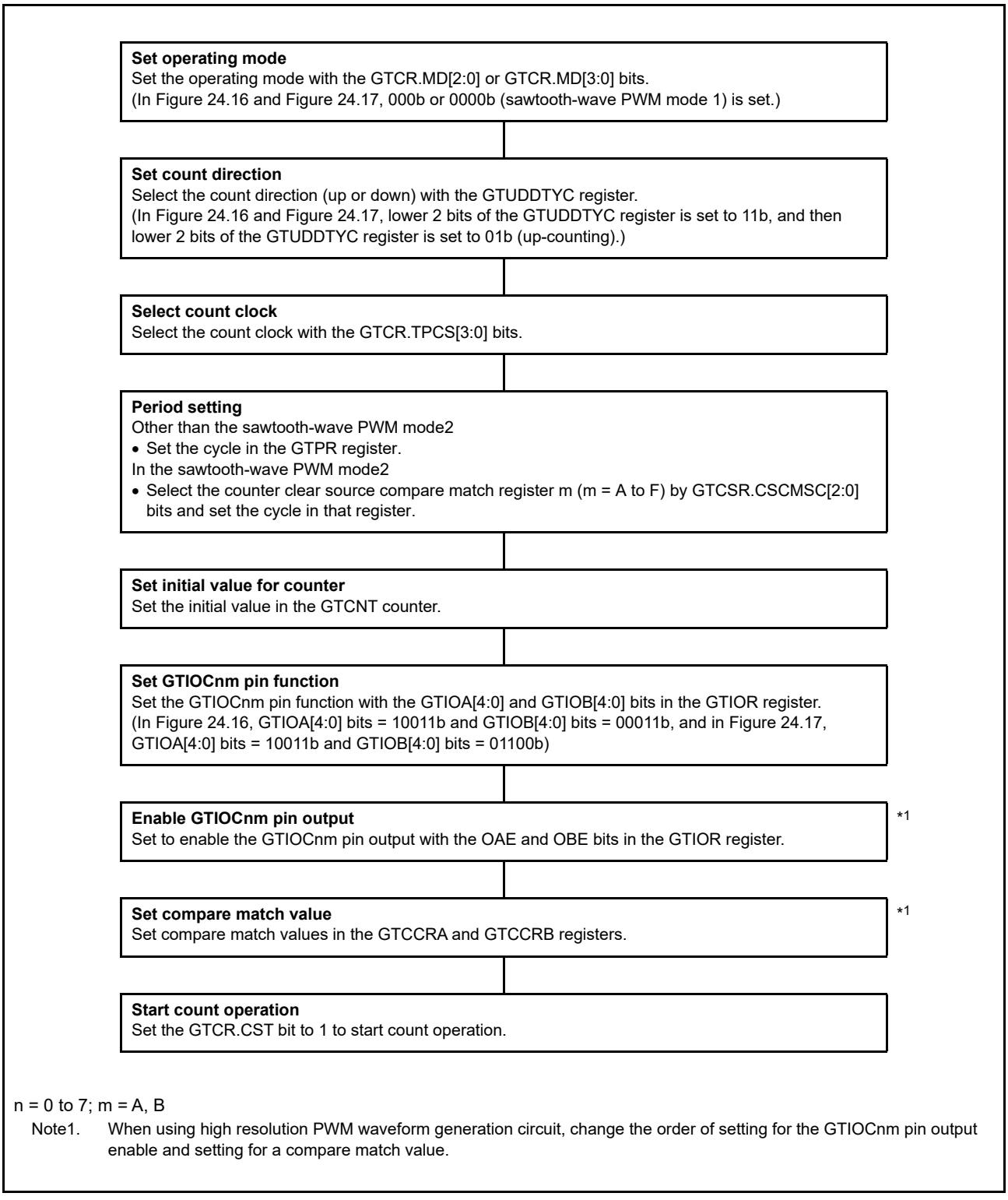


Figure 24.17 Example of Toggled Output Operation (2)

Figure 24.18 shows an example for setting toggled output operation.



n = 0 to 7; m = A, B

Note1. When using high resolution PWM waveform generation circuit, change the order of setting for the GTIOCnm pin output enable and setting for a compare match value.

Figure 24.18 Example for Setting Toggled Output Operation

24.3.1.3 Input Capture Function

Through detecting hardware sources selected by the GTICASR and GTICBSR registers, the GTCNT counter values can be transferred to the GTCCRA and GTCCRB registers respectively.

Figure 24.19 shows an example of input capture function operation.

In this example, the GPTW0.GTCNT counter performs up-counting, and settings have been made so that an input capture is performed at both edges of the GTIOC0A pin input and at the rising edge of the GTIOC0B pin input.

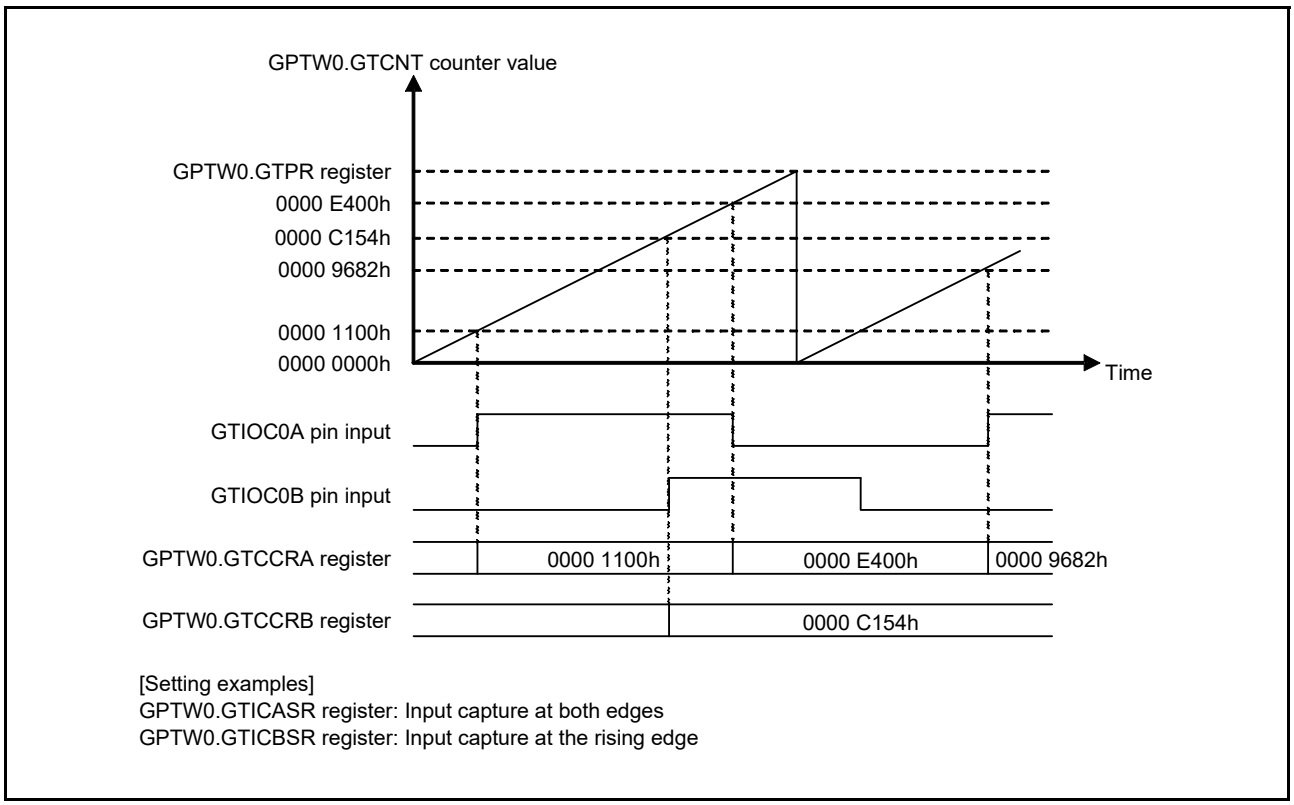


Figure 24.19 Example of Input Capture Operation

Figure 24.20 shows an example for setting input capture operation.

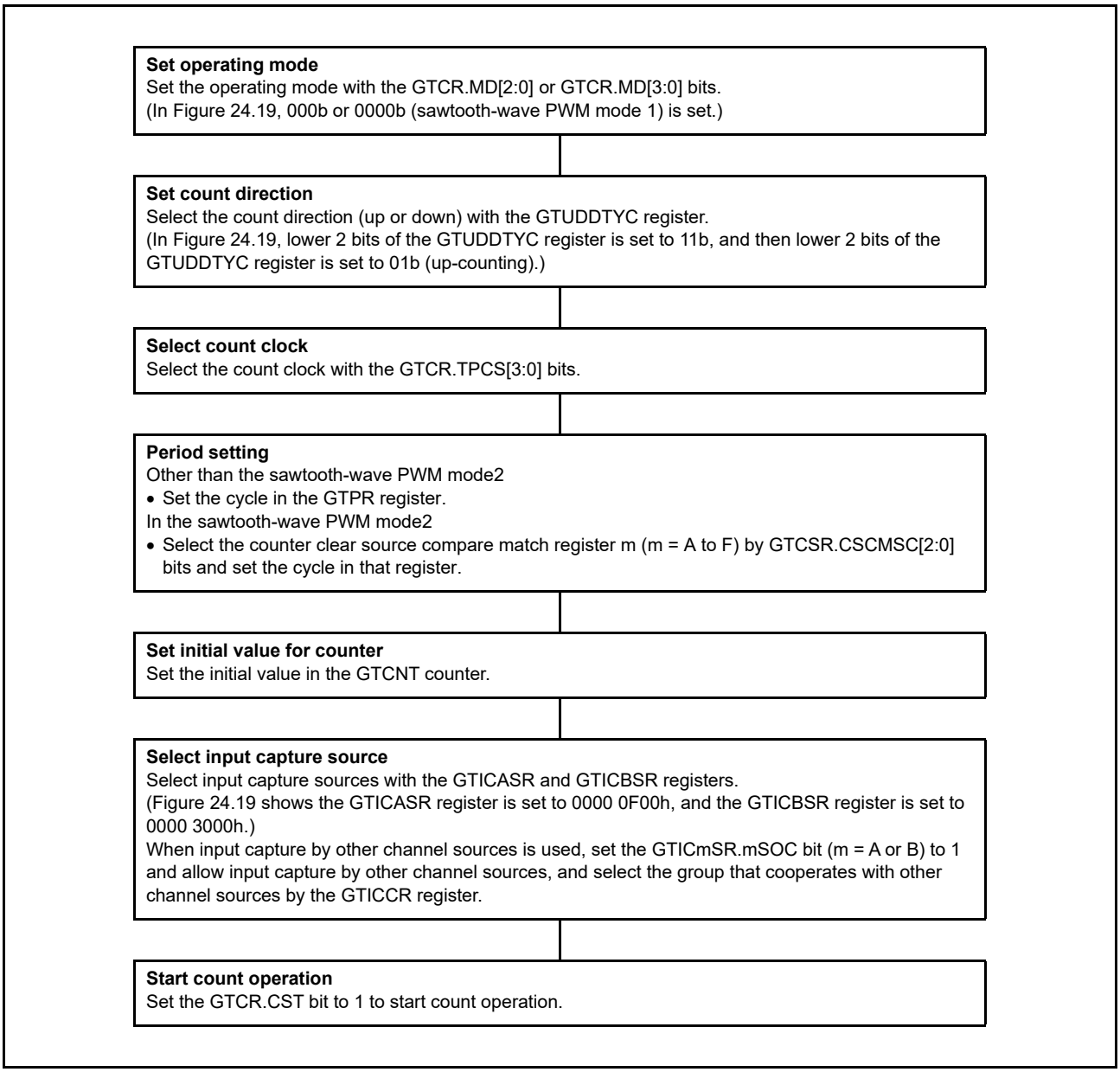
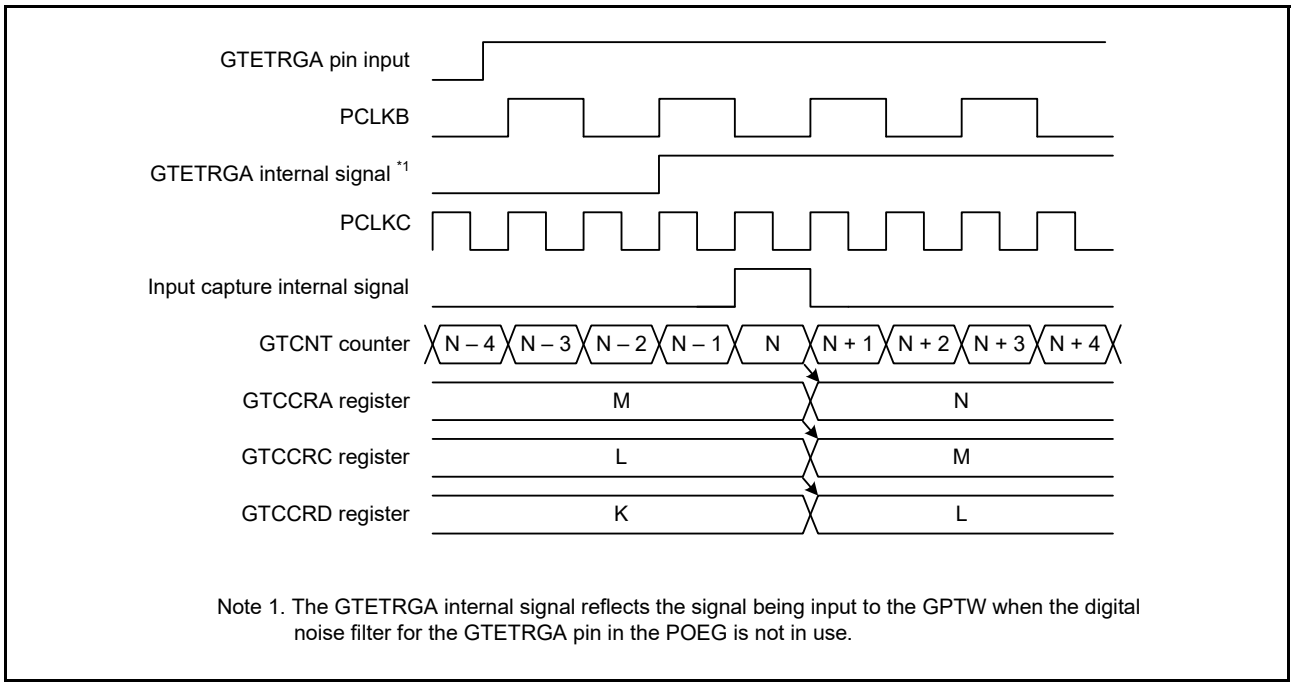


Figure 24.20 Example for Setting Input Capture Operation

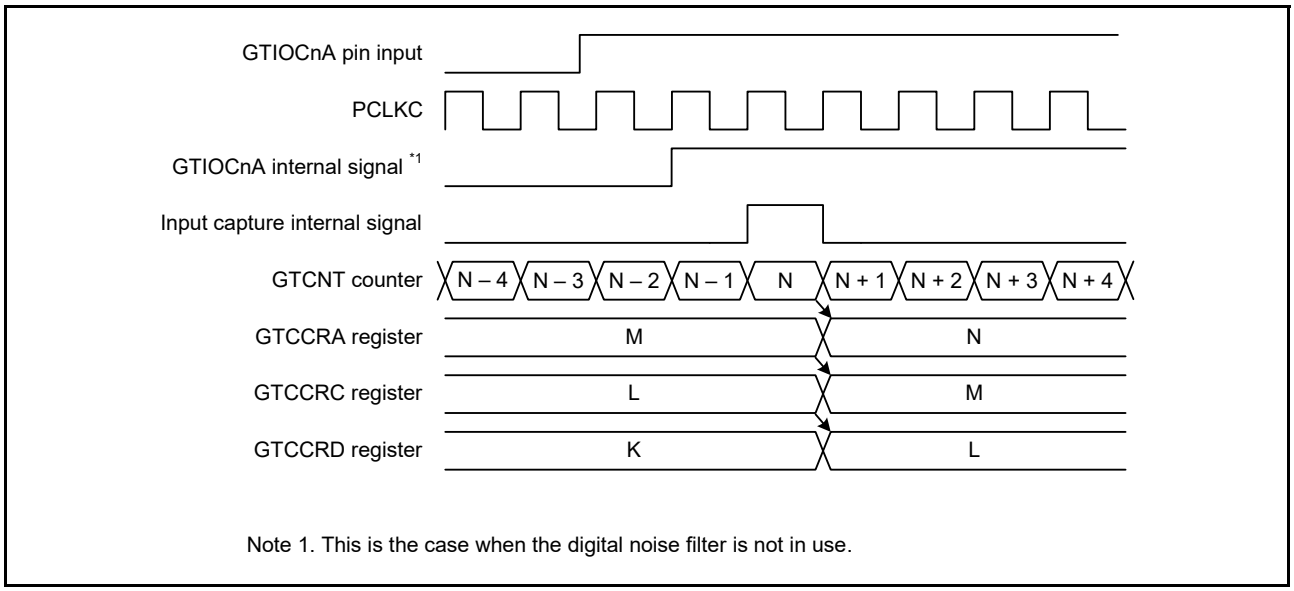
Figure 24.21 shows an example of the timing of input capture operation in response to a rising edge of the input on the GTETRGA pin.



Note 1. The GTETRGA internal signal reflects the signal being input to the GPTW when the digital noise filter for the GTETRGA pin in the POEG is not in use.

Figure 24.21 Example of the Timing of Input Capture Operation in Response to a Rising Edge of the Input on the GTETRGA Pin

Figure 24.22 shows an example of the timing of input capture operation in response to a rising edge of the input on the GTIOCnA pin.



Note 1. This is the case when the digital noise filter is not in use.

Figure 24.22 Example of the Timing of Input Capture Operation in Response to a Rising Edge of the Input on the GTIOCnA Pin

Figure 24.23 shows an example of the timing of operations for input capture in response to event input from the ELCA. This is an example of capture of the counter value by the GPTW1.GTCCRA register in response to an input signal. An event signal is output to the ELC after matches in comparison with the GPTW0.GTCCRA register. This is selected as a trigger for output to the GPTW by the ELC as event source A. The GPTW0 compare match A signal is synchronized with PCLKC. The ELC receives the signal in synchronization with PCLKB, and outputs the GPTW event source A signal after 1 cycle of PCLKB has elapsed.

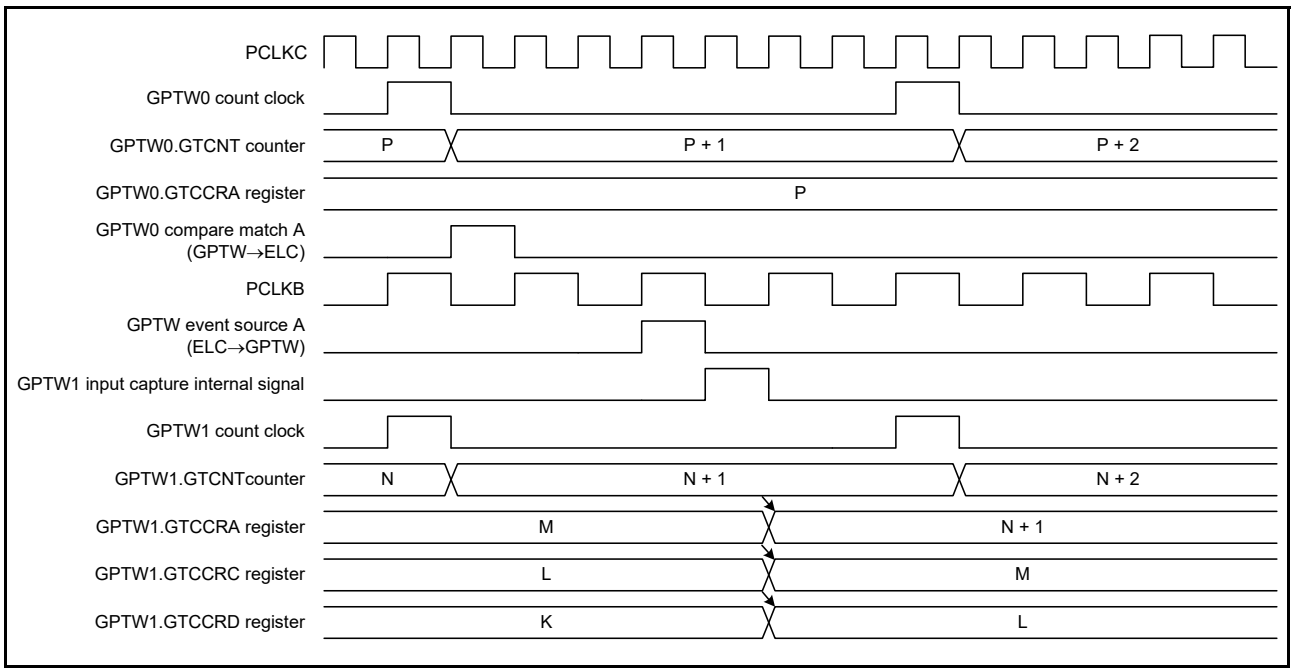


Figure 24.23 Example of the Timing of Operations for Input Capture in Response to Event Input from the ELCA

Figure 24.24 shows the input capture operation timing by the counter clock of other channels.

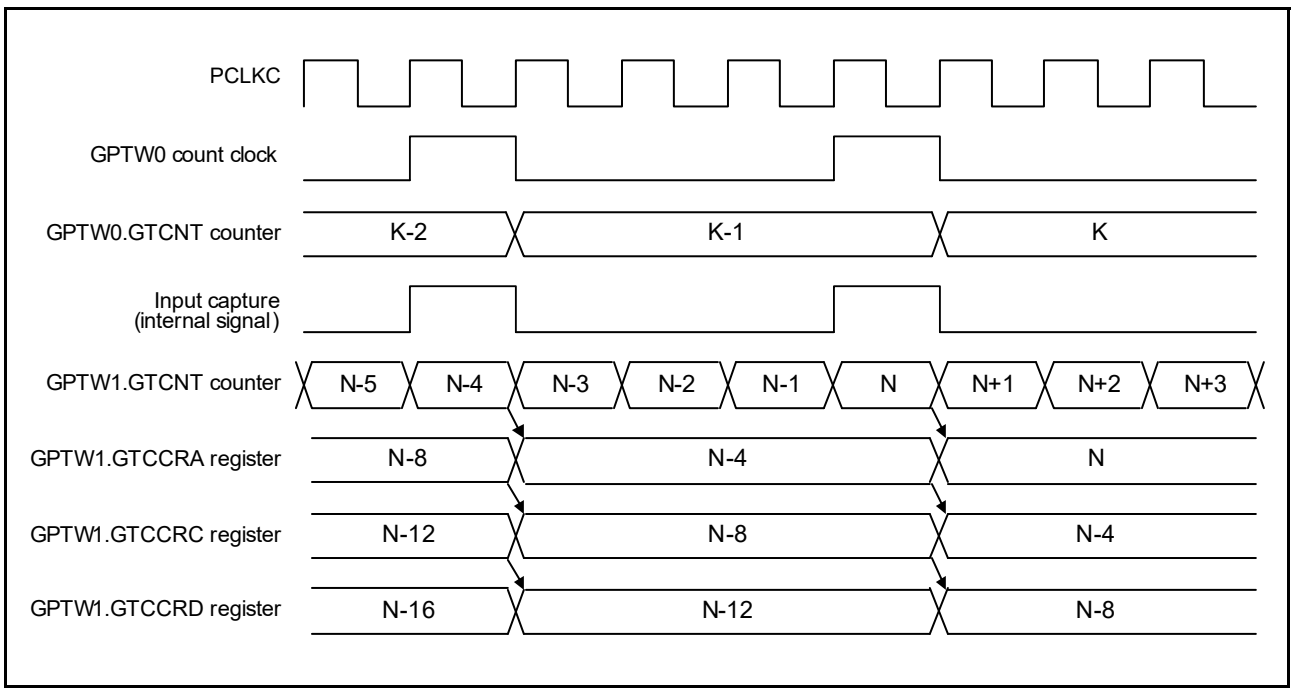


Figure 24.24 Example of the Timing of Input Capture Operation in Response to Count Clock from Other Channel

24.3.2 Buffer Operation

The following buffer operation can be set with the GTBER register.

- Buffer operation with the GTPR, GTPBR, and GTPDBR registers used together
- Buffer operation with the GTCCRA, GTCCRC, and GTCCRD registers used together
- Buffer operation with the GTCCRB, GTCCRE, and GTCCRF registers used together
- Buffer operation with the GTADTRA, GTADTBRA, and GTADTDBRA registers used together
- Buffer operation with the GTADTRB, GTADTBRB, and GTADTDBRB registers used together

The following buffer operation can be set with the GTDTCR register.

- Buffer operation with the GTDVU and GTDBU registers used together
- Buffer operation with the GTDVD and GTDBD registers used together

The following buffer operation can be set with the GTBER2 register.

- Buffer operation with the GTCCRA, GTCCRE, and GTCCRF registers (in complementary PWM mode 3, 4)
- Buffer operation with the GTOLBR.GTIOAB[4:0] bits and GTIOR.GTIOA[4:0] bits
- Buffer operation with the GTOLBR.GTIOBB[4:0] bits and GTIOR.GTIOB[4:0] bits

24.3.2.1 GTPR Register Buffer Operation

The GTPBR register can function as a buffer register for the GTPR register, and the GTPDBR register can function as a buffer register for the GTPBR register (double buffer register for the GTPR register).
In complementary PWM mode, the buffer transfer from the GTPDBR register to the temporary register P is performed only in the master channel (GPTWn). The temporary register P is transferred to each GTPBR register of master channel, slave channel 1 (GPTWn+1), and slave channel 2 (GPTWn+2). Transfer from the GTPBR register to the GTPR register is made concurrently in three channels. Therefore, the same value is stored in the same register of the three channels. The GTPR register of the master channel represents the GTCNT counter's (GTCNTn) period of the master channel. In the slave channels, periods are controlled using the GTPR register value and the GTDVU register value.
The setting is invalid in the sawtooth-wave PWM mode 2.
The buffer transfer is performed at an overflow (in up-counting) or underflow (in down-counting) in sawtooth-wave mode or in event counting, or at a trough in triangle-wave mode.
When in sawtooth-wave mode and in event counting, the buffer transfer is performed even when the counter clearing occurs during the count is in operation.

- Hardware source clearing

(Clear source selected with GTCSR.CSGTRGAR, CSGTRGAF, CSGTRGBR, CSGTRGBF, CSGTRGCR, CSGTRGCF, CSGTRGDR, CSGTRGDF, CSCARBL, CSCARBH, CSCAFBL, CSCAFBH, CSCBRAL, CSCBRAH, CSCBFAL, CSCBFAH, CSELCA, CSELCB, CSELCC, CSELCD, CSELCE, CSELCF, CSELCG, CSELCH, CSCMSC[2:0], or CP1CCE bit)

- Clearing by software

(When the GTCLR.CCLRn bit is written to 1 during the GTCSR.CCLR bit is set to 1) (n = 0 to 7)
In the case of sawtooth waves, the buffer transfer by clearing counter can by prohibited by the GTBER2.CCTPR bit.
Figure 24.10 shows the buffer transfer timing in the complementary PWM mode.

Table 24.11 The GTPR Buffer Transfer Timing in Complementary PWM Mode

Mode	Complementary PWM mode 1	Complementary PWM mode 2	Complementary PWM mode 3 Complementary PWM mode 4
GTPDBR register ↓ Temporary register P	After PCLKC 1 clock from the GTCCRD register write of slave channel 2 (GPTWn+2)	After PCLKC 1 clock from the GTCCRD register write of slave channel 2 (GPTWn+2)	After PCLKC 1 clock from the GTCCRD register write of slave channel 2 (GPTWn+2)
Temporary register P ↓ GTPBR register	When data is transferred to the temporary register P during up-counting middle • After PCLKC 1 clock from data transfer to the temporary register P When data is transferred to the temporary register P during a section other than up-counting middle • At the end of trough	When data is transferred to the temporary register P during down-counting middle • After PCLKC 1 clock from data transfer to the temporary register P When data is transferred to the temporary register P during a section other than down-counting middle • At the end of crest	When data is transferred to the temporary register P middle • After PCLKC 1 clock from data transfer to the temporary register P When data is transferred to the temporary register P during a section other than middle • At the end of crest/trough
GTPBR register ↓ GTPR register	At the end of crest • Counter clearing in an up-counting middle or crest section, including counter clearing enabled by setting the GTCSR.CP1CCE bit	At the end of trough • Counter clearing in a down-counting middle or trough section	At the end of crest At the end of trough • Counter clearing

To set the GTPR register to function as double buffer, set the GTBER.PR[1:0] bits to 10b or 11b. For single buffer operation, set 01b. Not to function as buffer, set 00b.
In complementary PWM mode, complementary PWM mode-specific buffer operation is performed regardless of the GTBER.PR[1:0] bit setting.
Figure 24.25 to Figure 24.27 show examples of the GTPR register buffer operation and Figure 24.31 shows an example for setting the GTPR register buffer operation.
For details of operation settings in complementary PWM mode, refer to section 24.3.3, PWM Output Operating Mode

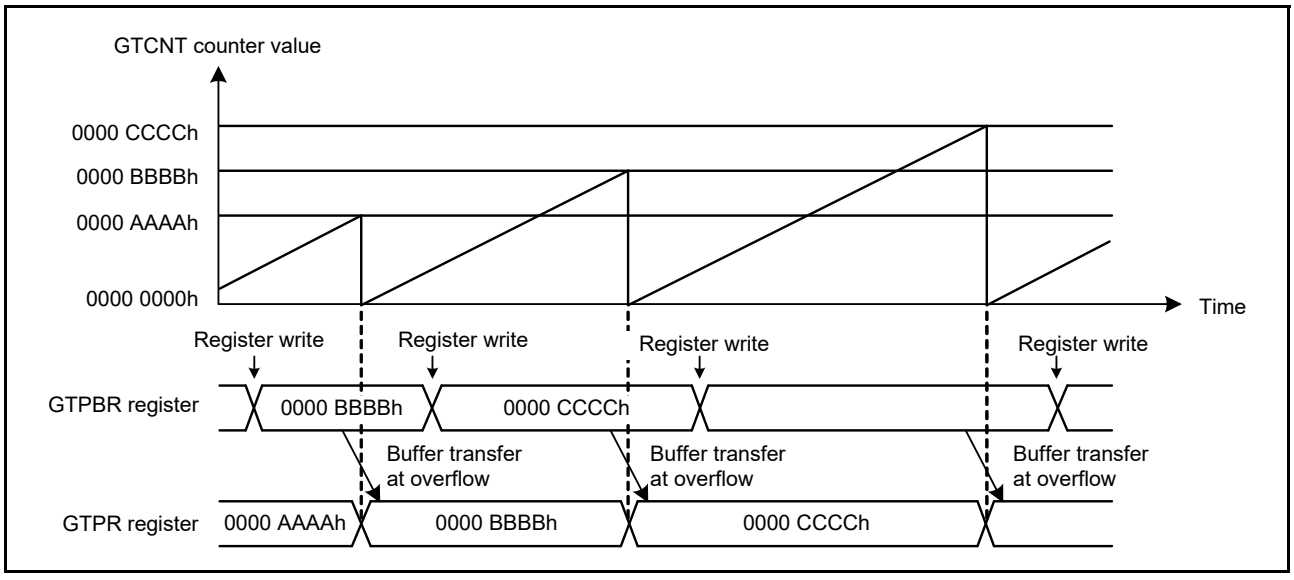


Figure 24.25 Example of the GTPR Register Buffer Operation (Sawtooth Waves in Up-Counting)

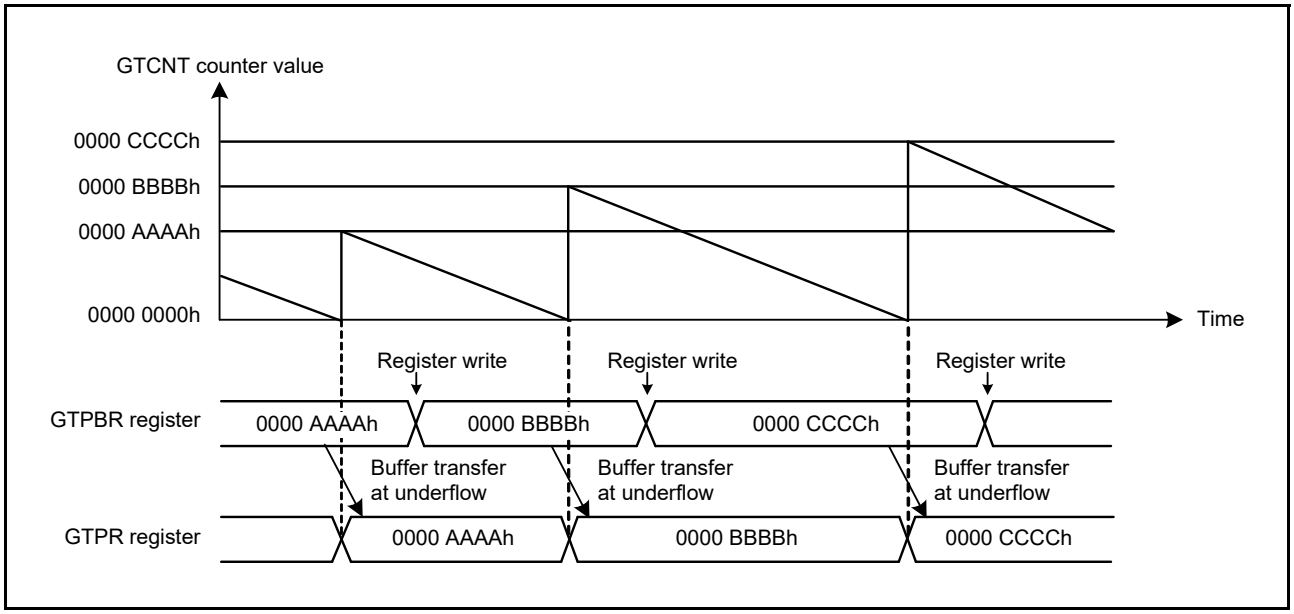


Figure 24.26 Example of the GTPR Register Buffer Operation (Sawtooth Waves in Down-Counting)

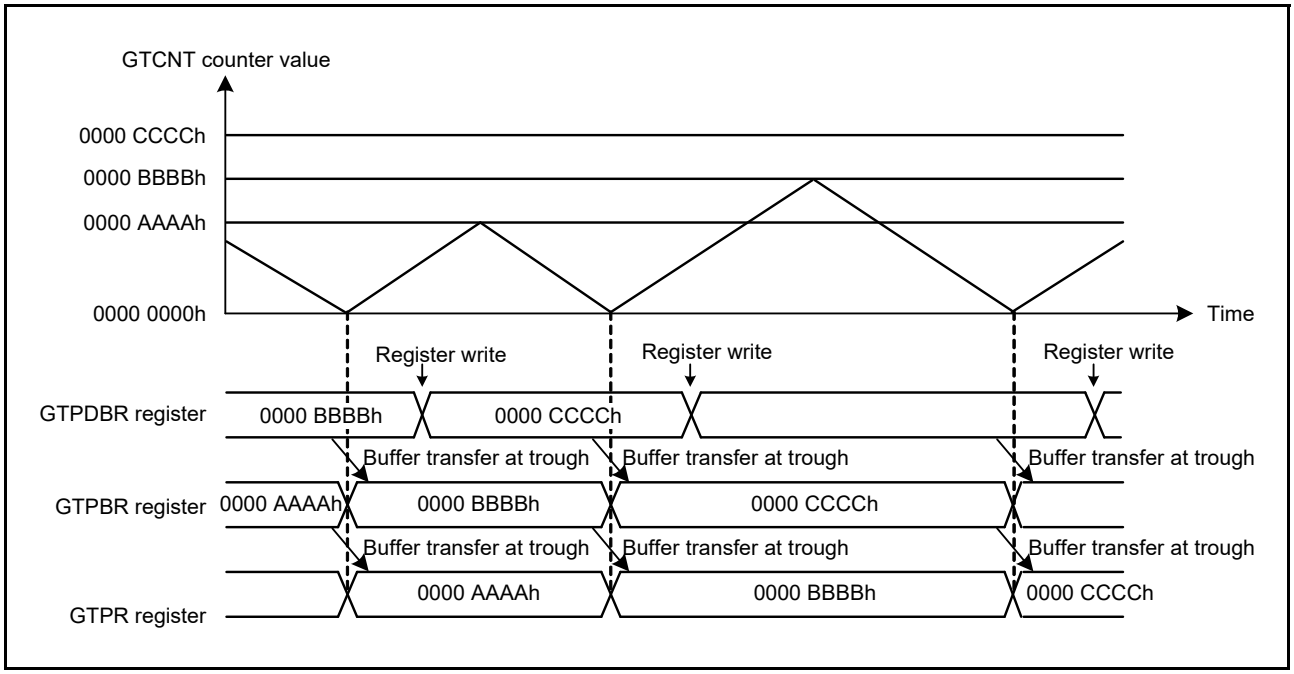


Figure 24.27 Example of the GTPR Register Double Buffer Operation (Triangle Waves)

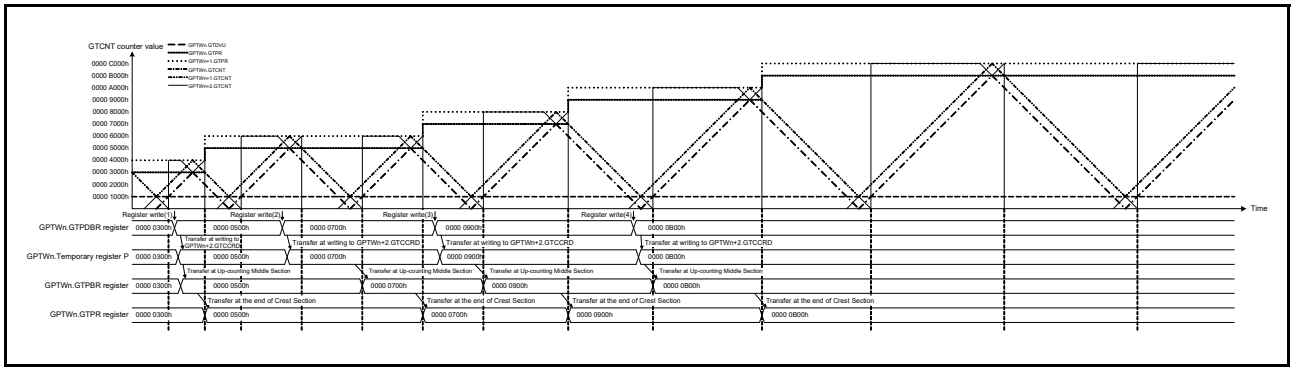


Figure 24.28 Example of GTPR Double Buffer Operation (Complementary PWM Mode 1)

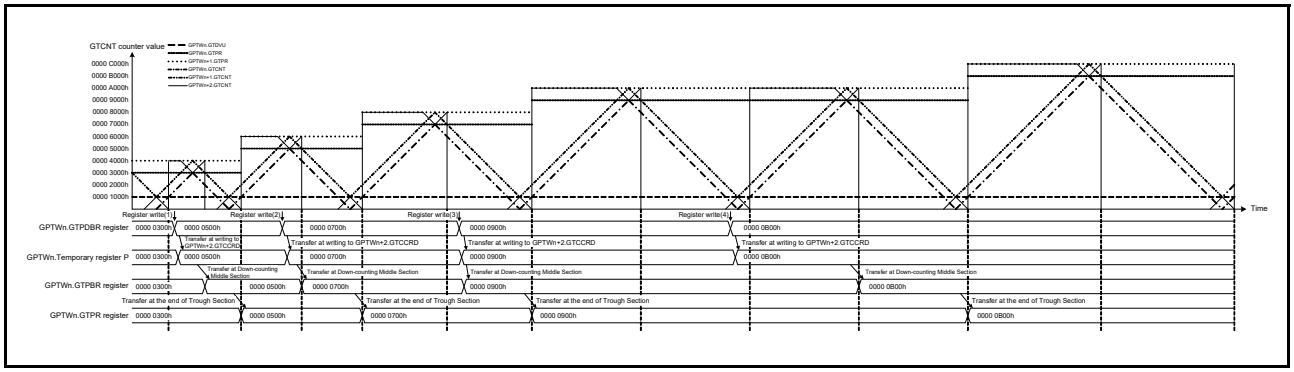


Figure 24.29 Example of GTPR Double Buffer Operation (Complementary PWM Mode 2)

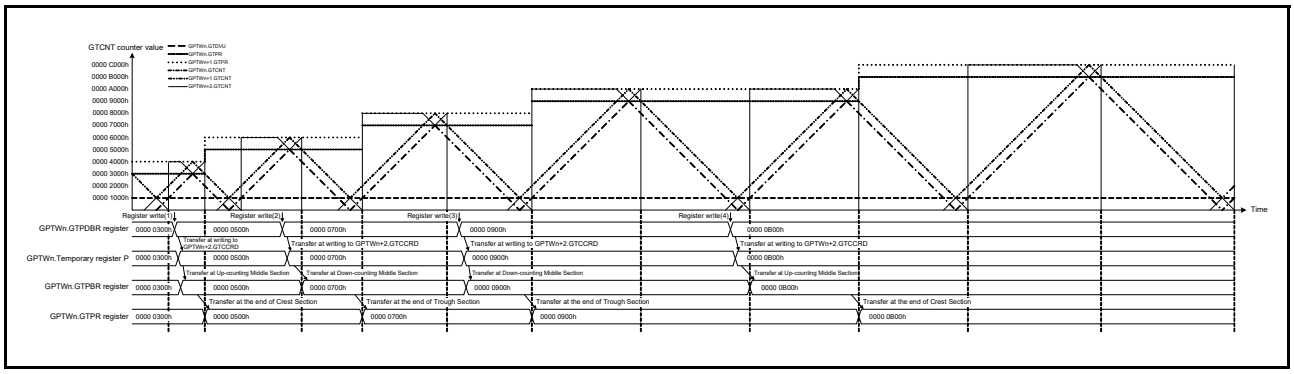


Figure 24.30 Example of GTPR Double Buffer Operation (Complementary PWM Mode 3, 4)

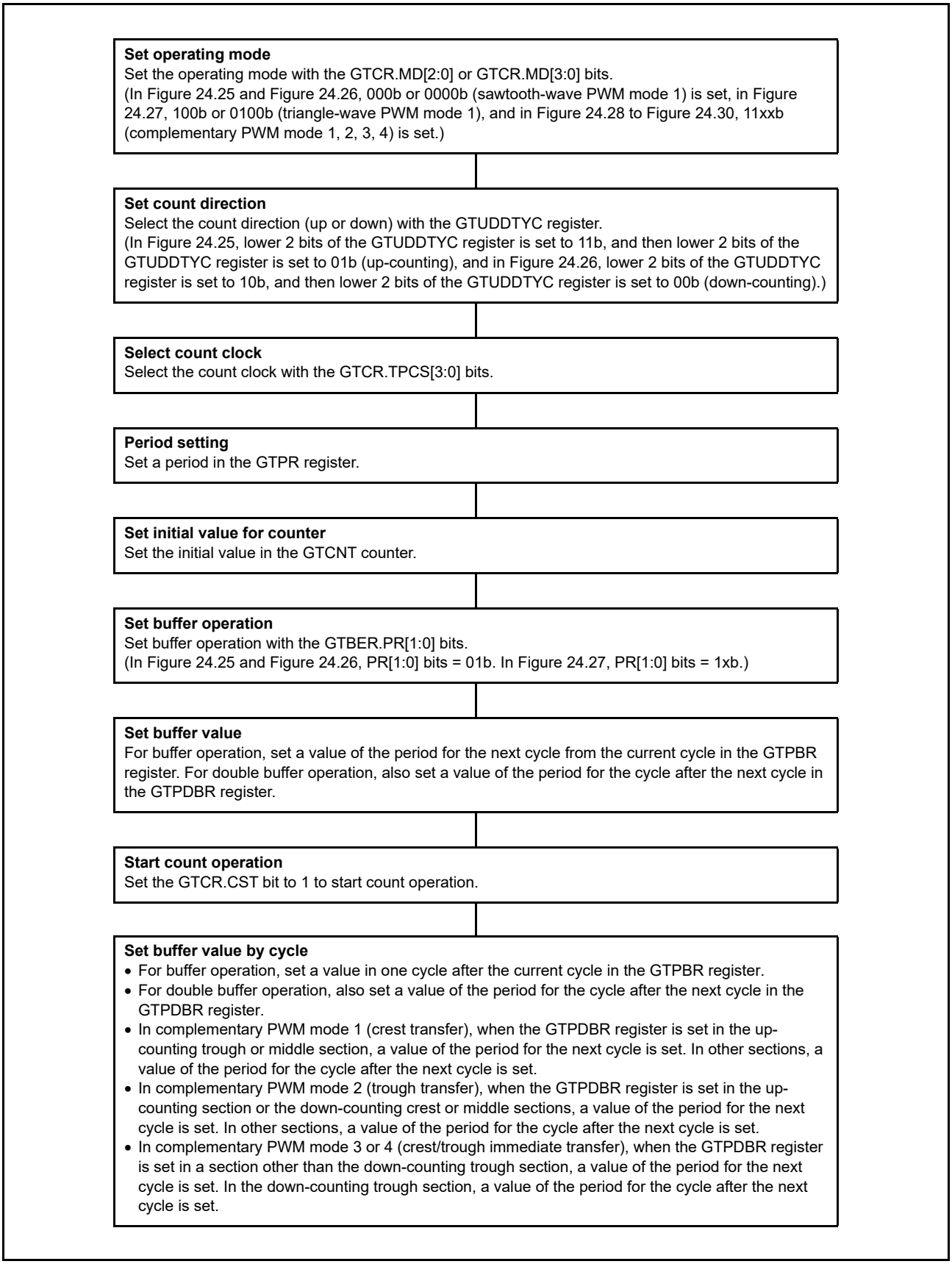


Figure 24.31 Example for Setting the GTPR Register Buffer Operation

24.3.2.2 Buffer Operation for the GTCCRA and GTCCRB Registers

The GTCCRC register can function as the GTCCRA buffer register and the GTCCRD register can function as the GTCCRC buffer register (double buffer register for the GTCCRA register). Similarly, the GTCCRE register can function as the GTCCRB buffer register and the GTCCRF register can function as the GTCCRE buffer register (double buffer register for the GTCCRB register).
To set the GTCCRA or GTCCRB register to function as a double buffer, set the GTBER.CCRA[1:0] or GTBER.CCRB[1:0] bits to 10b or 11b. For single buffer operation, set 01b. Not to function as buffer, set 00b.
The following describes buffer operation during output compare and input capture operation.
In sawtooth-wave one-shot pulse mode, triangle-wave PWM mode 3, complementary PWM mode, the buffer operations that specific each PWM output operation mode are performed regardless of the setting of GTBER.CCRA[1:0] bits and GTBER.CCRB[1:0] bits.

(1) When the GTCCRA or GTCCRB Register Functions as Output Compare Register

In sawtooth-wave one-shot pulse mode, triangle-wave PWM mode 3, complementary PWM mode, the buffer operations that specific each PWM output operation mode are performed regardless of the setting of GTBER.CCRA[1:0] bits and GTBER.CCRB[1:0] bits. For details, refer section 24.3.3, PWM Output Operating Mode. Other than above PWM output operation mode, buffer transfer is performed for the following three cases:

- Buffer transfer at an overflow (in up-counting) and underflow (in down-counting)
  Buffer transfer is performed at an overflow (in up-counting) or an underflow (in down-counting) in sawtooth-wave mode and in event counting, and at a trough (triangle-wave PWM mode 1) or at a crest/trough (triangle-wave PWM mode 2) in triangle-wave mode.
- Buffer transfer by counter clearing
  In sawtooth-wave mode and in event counting, the buffer transfer by similar counter clearing sources in counting operation and in section 24.3.2.1, GTPR Register Buffer Operation is performed in the same way at an overflow (in up-counting) or an underflow (in down-counting).
  In sawtooth-wave, the buffer transfer of GTCCRm register by counter clearing is possible to be disabled with GTBER2.CCTCm bit (m = A, B).
  In triangle-wave mode, buffer transfer by counter clearing is not performed.
- Buffer transfer by compare match
  In sawtooth-wave, the buffer transfer by the compare match of GTCCRm register enabled by GTBER2.CMTCm (m = A, B) bit is performed.
- Forcible buffer transfer
  In both sawtooth-wave mode and triangle-wave mode, buffer transfer of the GTCCRA and GTCCRB registers is forcibly performed by writing 1 to the GTBER.CCRSWT bit while the counting is stopped.
  In sawtooth-wave one-shot pulse mode and triangle-wave PWM mode 3, buffer transfer from the GTCCRD register to the temporary register A and from the GTCCRF register to the temporary register B is also performed using the forcible buffer operation.

When the setting of the GTBER.DBRTECm (m = A, B) bit is 1 in sawtooth-wave one-shot pulse mode or triangle-wave PWM mode 3, transfer from the intermediate buffers to the GTCCRm (m = A, B) registers is repeated on a cyclic basis even while buffer transfer is disabled by the setting of the GTBER.BD[0] bit or buffer transfer extended skipping function (function for repeated double-buffer operation while buffer transfer is disabled). For details, refer to section 24.8.2.2, Repeated Double-Buffered Operation When Disabling GTCCR Buffer Transfer.
Figure 24.32 to Figure 24.35 show examples of the GTCCRA and GTCCRB registers buffer operations, and Figure 24.36 shows an example for setting the GTCCRA and GTCCRB registers buffer operations.

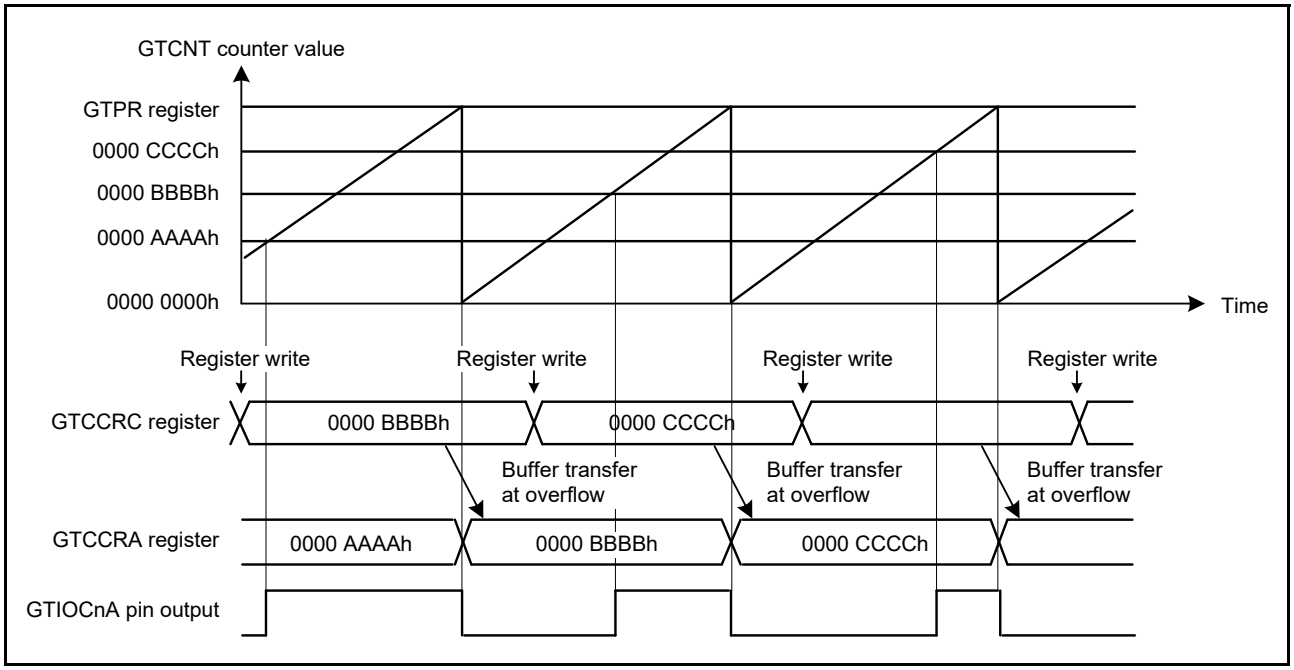


Figure 24.32 Example of GTCCRA and GTCCRB Registers Buffer Operation (Output Compare, Sawtooth-Waves, Sawtooth-Wave PWM mode 1 in Up-Counting, High Output at GTCCRA Register Compare Match, Low Output at the End of the Cycle) (n = 0 to 7)

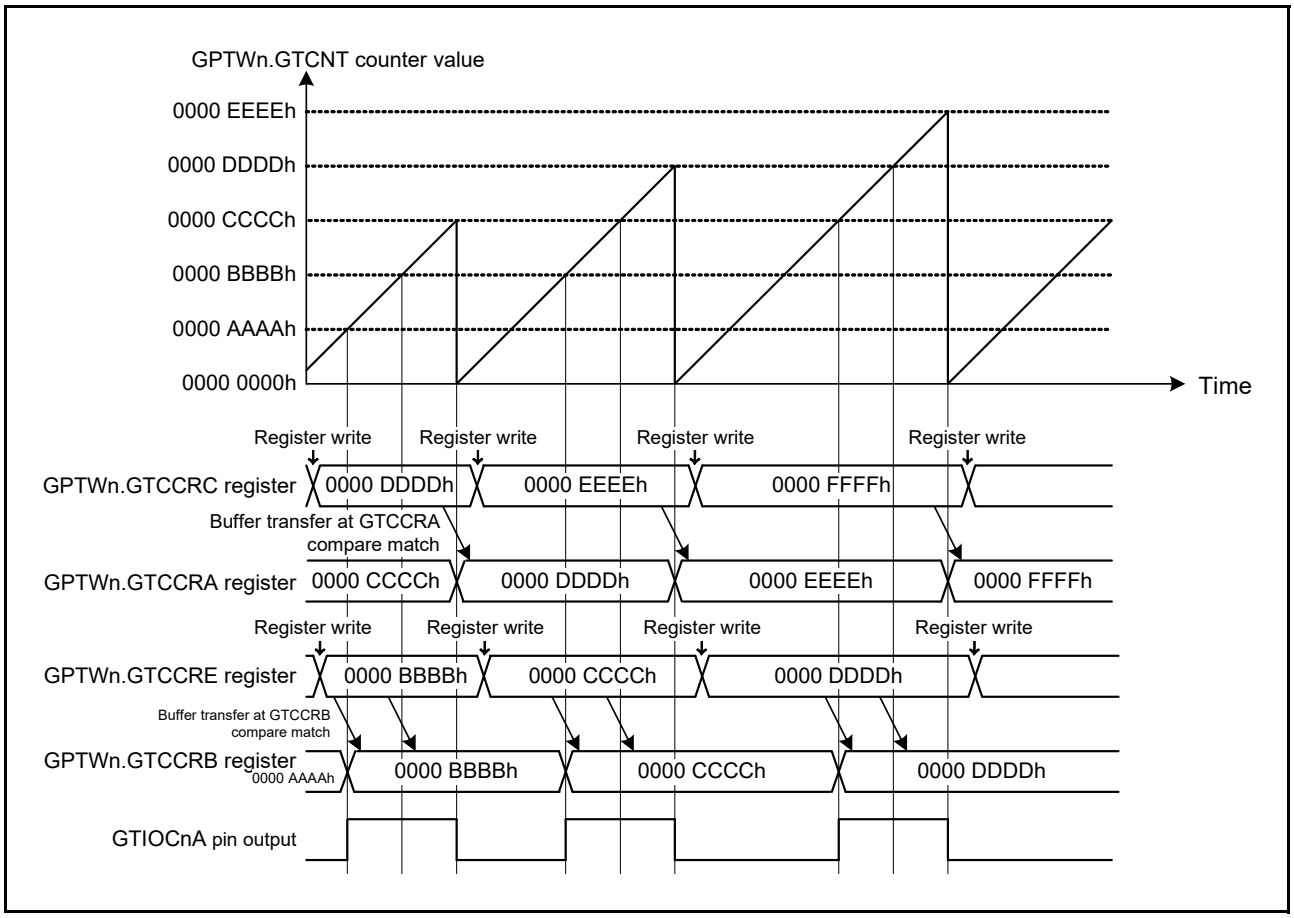


Figure 24.33 Example of GTCCRA and GTCCRB Registers Buffer Operation (Output Compare, Sawtooth-Wave PWM mode 2, Buffer Transfer at GTCCRA Register Compare Match, Counter Clearing, Low Output, Buffer Transfer at GTCCRB Register Compare Match, High Output) (n = 0 to 7)

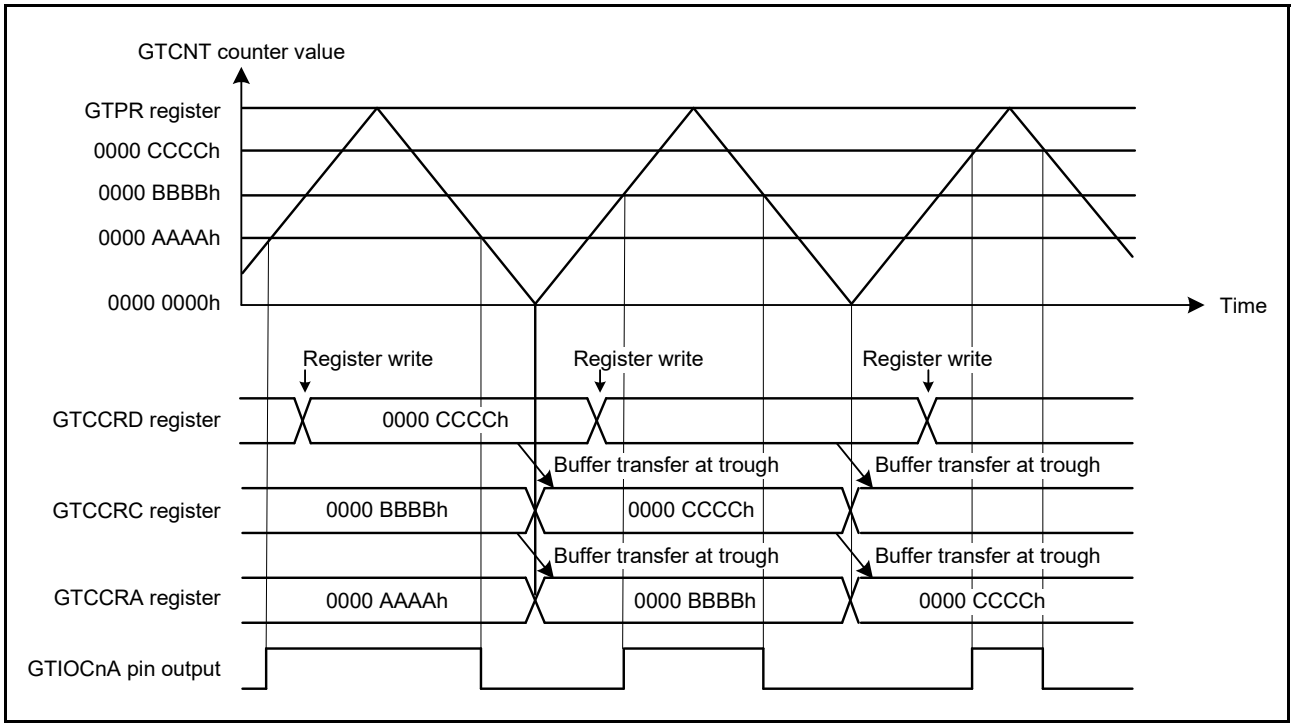


Figure 24.34 Example of GTCCRA and GTCCRB Registers Double Buffer Operation (Output Compare, Triangle Waves, Buffer Operation at Trough, Output Toggled at GTCCRA Register Compare Match, Output Retained at the End of the Cycle) (n = 0 to 7)

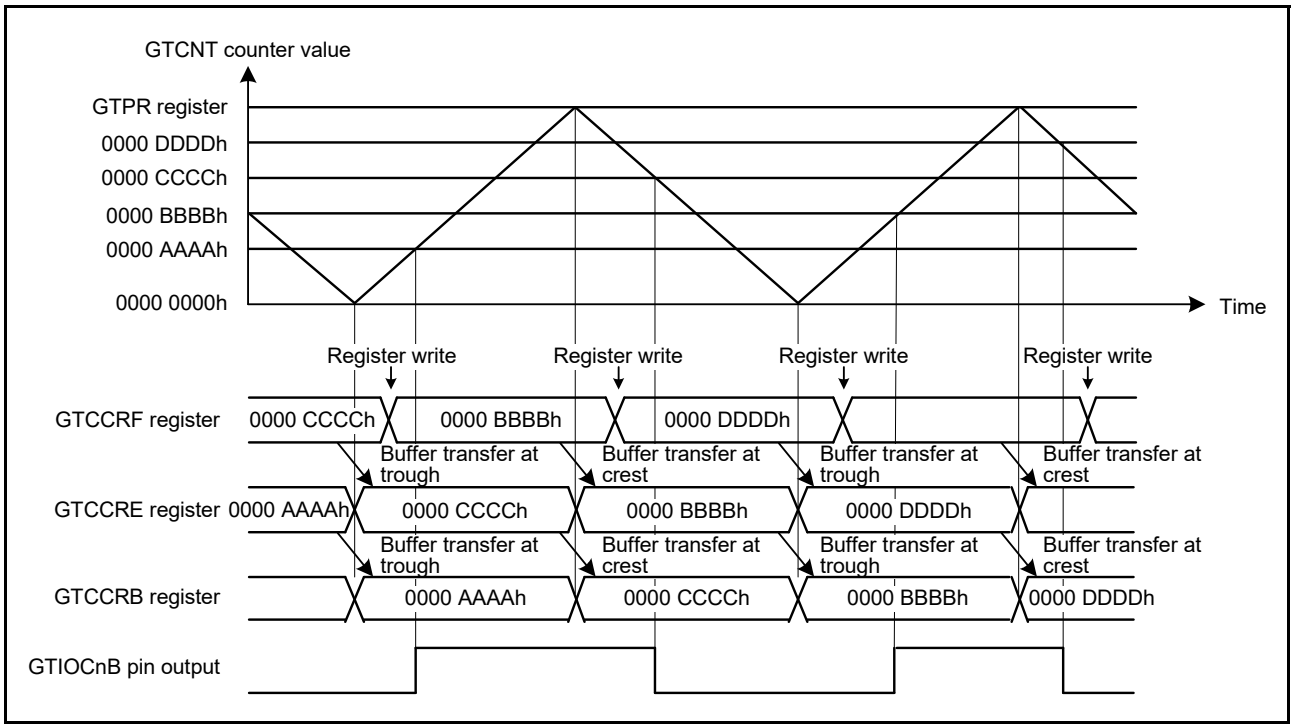
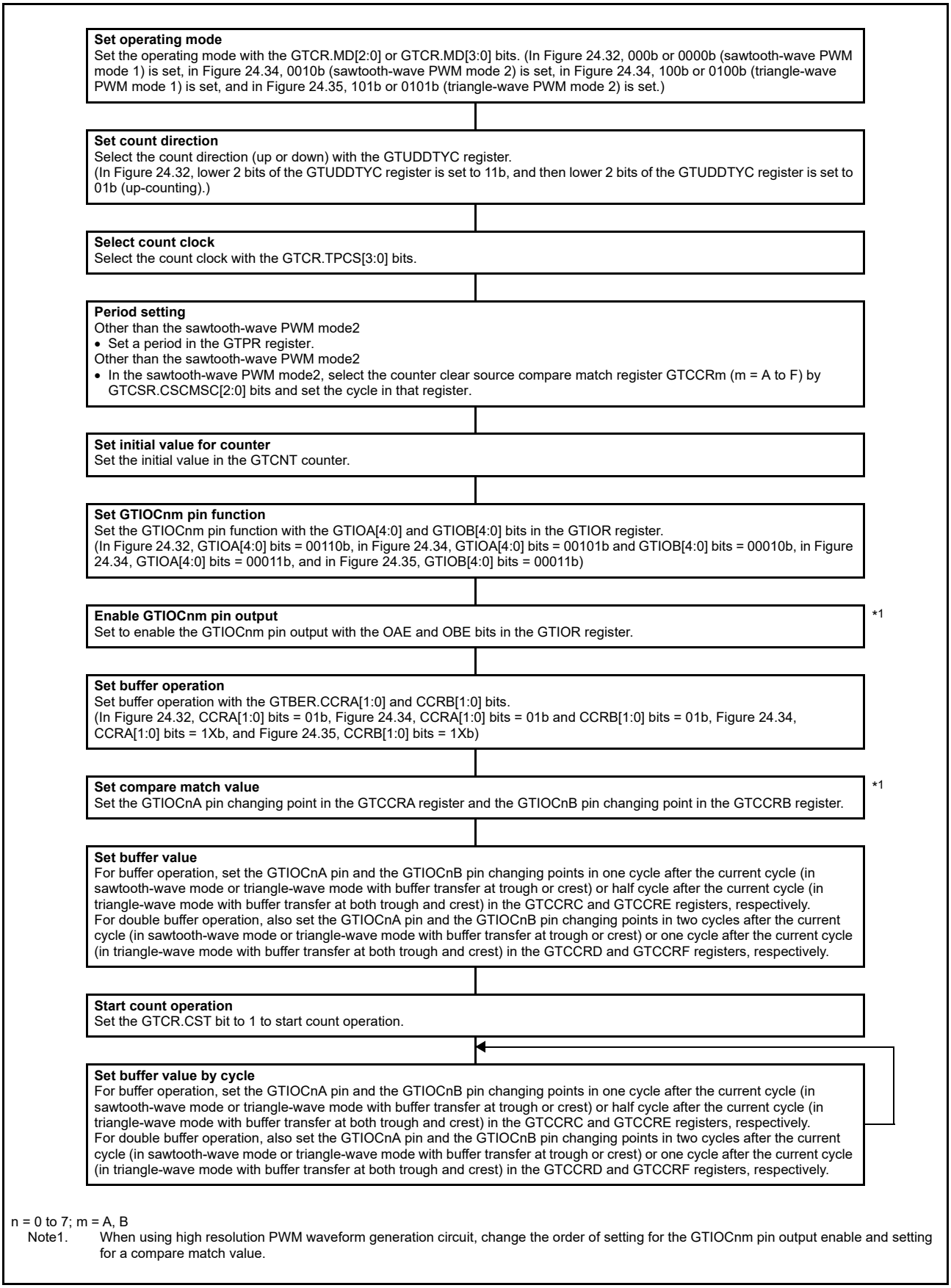


Figure 24.35 Example of GTCCRA and GTCCRB Registers Double Buffer Operation (Output Compare, Triangle Waves, Buffer Operation at Both Troughs and Crests, Output Toggled at GTCCRB Register Compare Match, Output Retained at the End of the Cycle) (n = 0 to 7)



n = 0 to 7; m = A, B

Note1. When using high resolution PWM waveform generation circuit, change the order of setting for the GTIOCnm pin output enable and setting for a compare match value.

Figure 24.36 Example for Setting Buffer Operation of the GTCCRA and GTCCRB Registers (for Output Compare)

(2) When the GTCCRA or GTCCRB Register Functions as Input Capture Register

The timing of transfer to the buffers is when an input capture is generated. When an input capture is generated, the GTCNT counter value is transferred to GTCCRA and GTCCRB registers and the stored GTCCRA and GTCCRB registers values are transferred to buffer registers. Buffer transfer by counter clearing is not performed at input capture. Figure 24.37 and Figure 24.38 show examples of buffer operation of the GTCCRA and GTCCRB registers, and Figure 24.39 shows an example for setting buffer operation of the GTCCRA and GTCCRB registers.

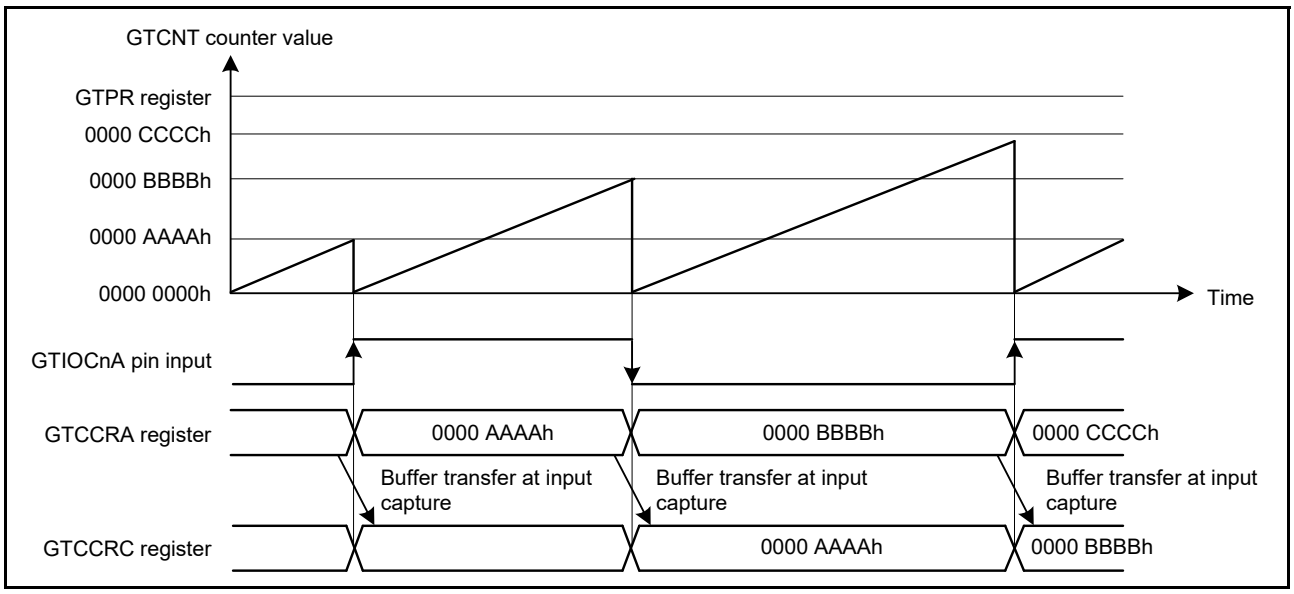


Figure 24.37 Example of Buffer Operation of the GTCCRA and GTCCRB Registers (Counting up to Produce a Sawtooth Wave, and Clearing the GTCNT Counter and Input Capture on Both Edges of the Input on the GTIOCnA Pin) (n = 0 to 7)

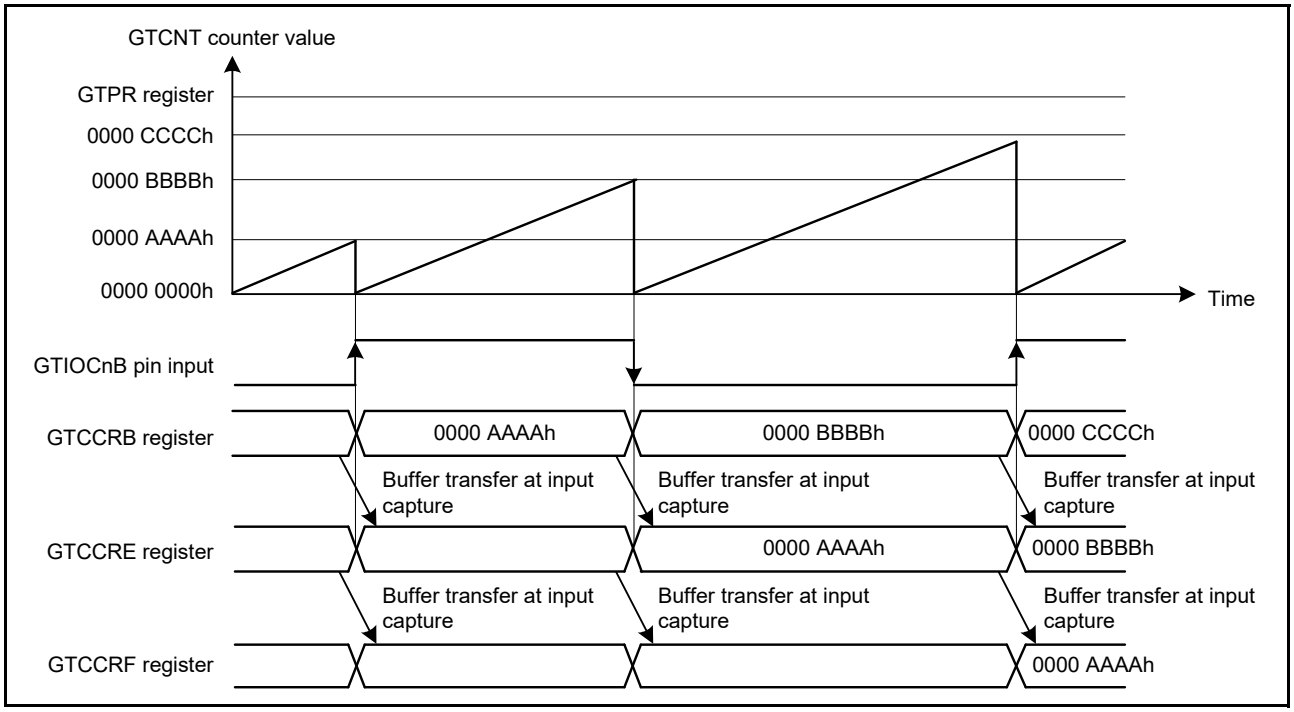


Figure 24.38 Example of Double Buffer Operation of the GTCCRA and GTCCRB Registers (Counting up to Produce a Sawtooth Wave, and Clearing the GTCNT Counter and Input Capture on Both Edges of the Input on the GTIOCnB Pin) (n = 0 to 7)

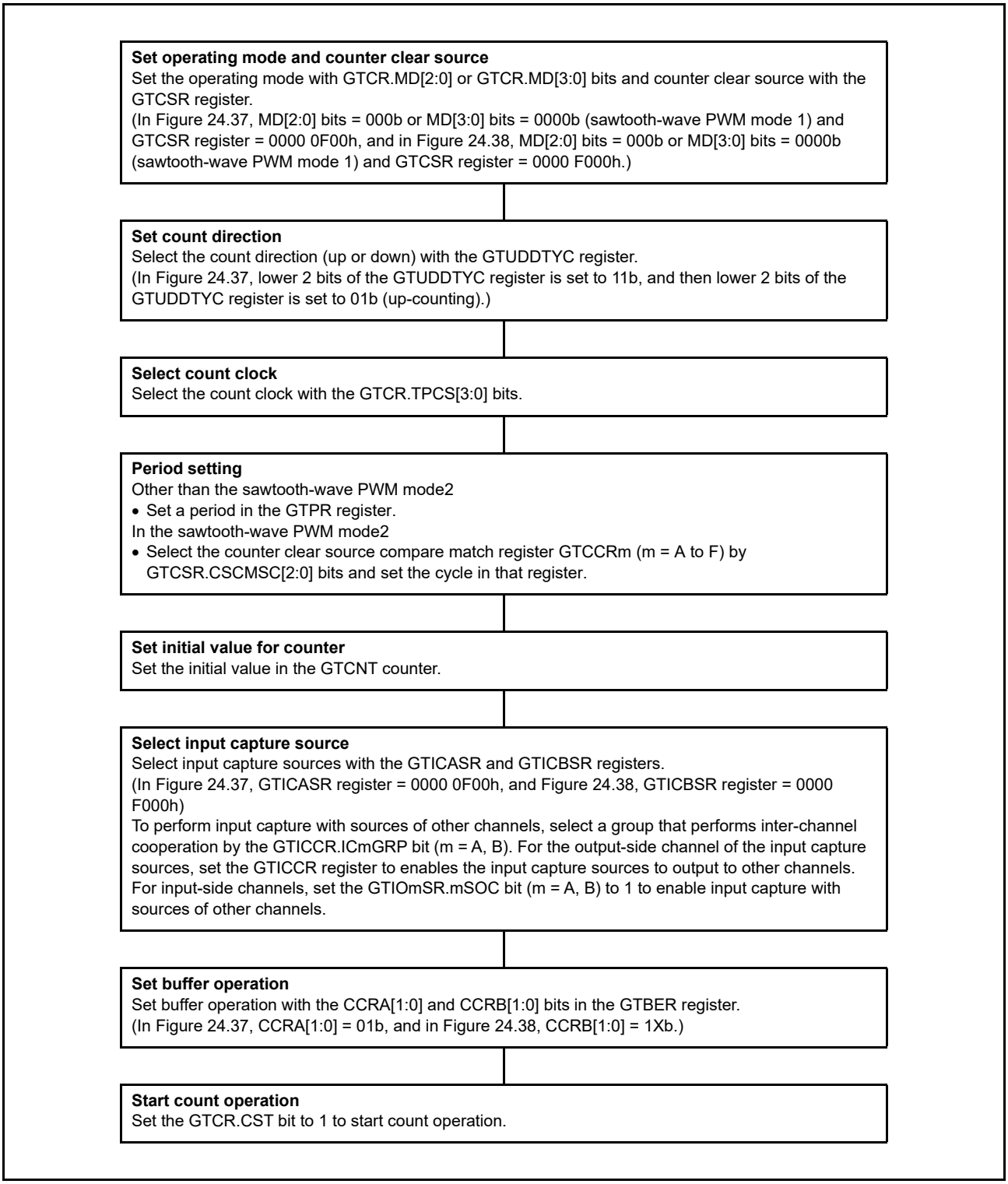


Figure 24.39 Example for Setting Buffer Operation of the GTCCRA and GTCCRB Registers (for Input Capture)

24.3.2.3 Buffer Operation for the GTADTRA and GTADTRB Registers

The GTADTBRA register can function as the GTADTRA buffer register and the GTADTDBRA register can function as the GTADTBRA buffer register (double buffer register for the GTADTRA register). Similarly, the GTADTBRB register can function as the GTADTRB buffer register and the GTADTDBRB register can function as the GTADTBRB buffer register (double buffer register for the GTADTRB register).

To set the GTADTRA or GTADTRB register to function as a double buffer, set the GTBER.ADTDA or ADTDB bit to 1. For single buffer operation, set 0. Not to function as buffer, set the GTBER.ADTTA[1:0] or ADTTB[1:0] bits to 00b.

The buffer transfer timing can be set with the ADTTA[1:0] and ADTTB[1:0] bits to an overflow (in up-counting) or an underflow (in down-counting) in sawtooth-wave mode, with ADTTA[1:0] and ADTTB[1:0] bits to 01b for a crest, to 10b for a trough, or to11b for both crest and trough in triangle-wave mode or complementary PWM mode.

In sawtooth-wave mode, when the ADTTA[1:0] and ADTTB[1:0] bits are set to value other than 00b and in count operation, the buffer transfer, by similar counter clearing sources in section 24.3.2.1, GTPR Register Buffer Operation, is performed in the same way at an overflow (in up-counting) or an underflow (in down-counting).

In complementary PWM mode, the buffer transfer is performed after one PCLK cycle from the GTCCRD register write of slave channel 2.

In sawtooth-wave mode, the buffer transfer of GTADTRm register by counter clearing can be disabled by GTBER2.CCTADm (m = A, B) bit setting.

In sawtooth-wave mode, the buffer transfer of GTADTRm register by its own compare match can be enabled by the GTBER2.CMTADm (m = A, B) bit setting.

Figure 24.40 to Figure 24.44 show examples of buffer operation of the GTADTRA and GTADTRB registers, and Figure 24.45 shows an example for setting buffer operation of the GTADTRA and GTADTRB registers.

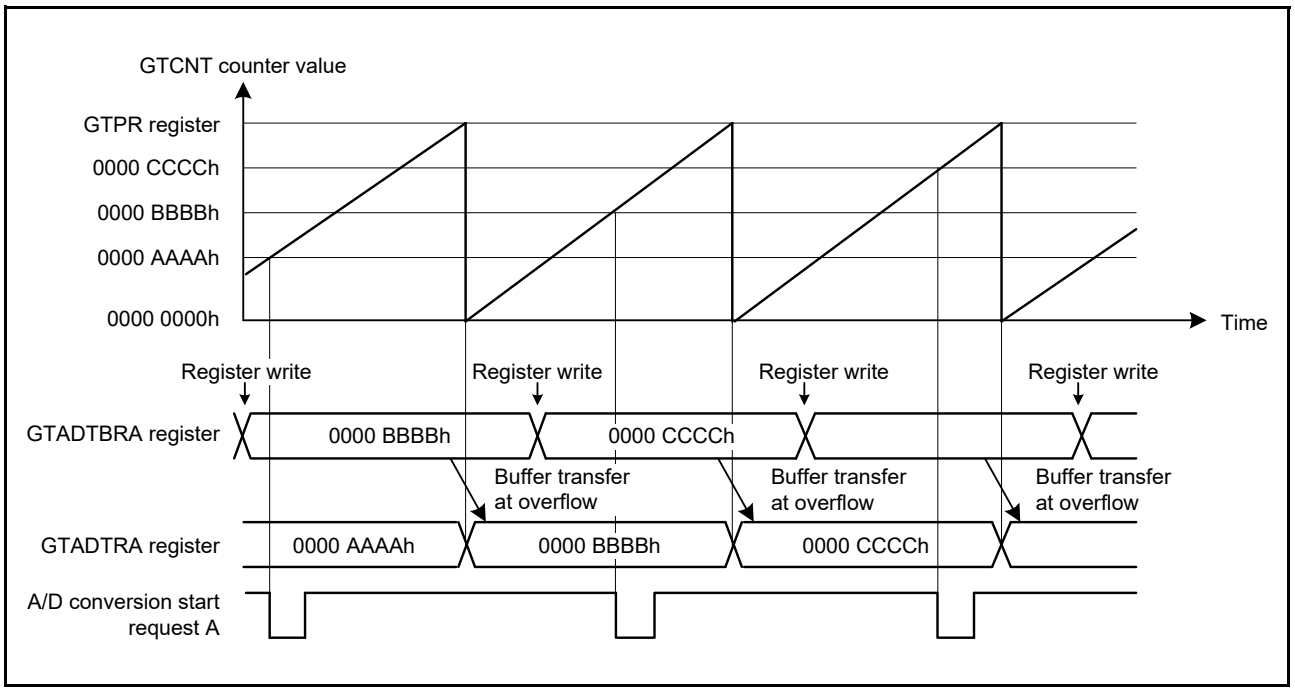


Figure 24.40 Example of Buffer Operation of the GTADTRA and GTADTRB Registers (Sawtooth Waves in Up-Counting, A/D Conversion Start Request Generated by Up-Counting)

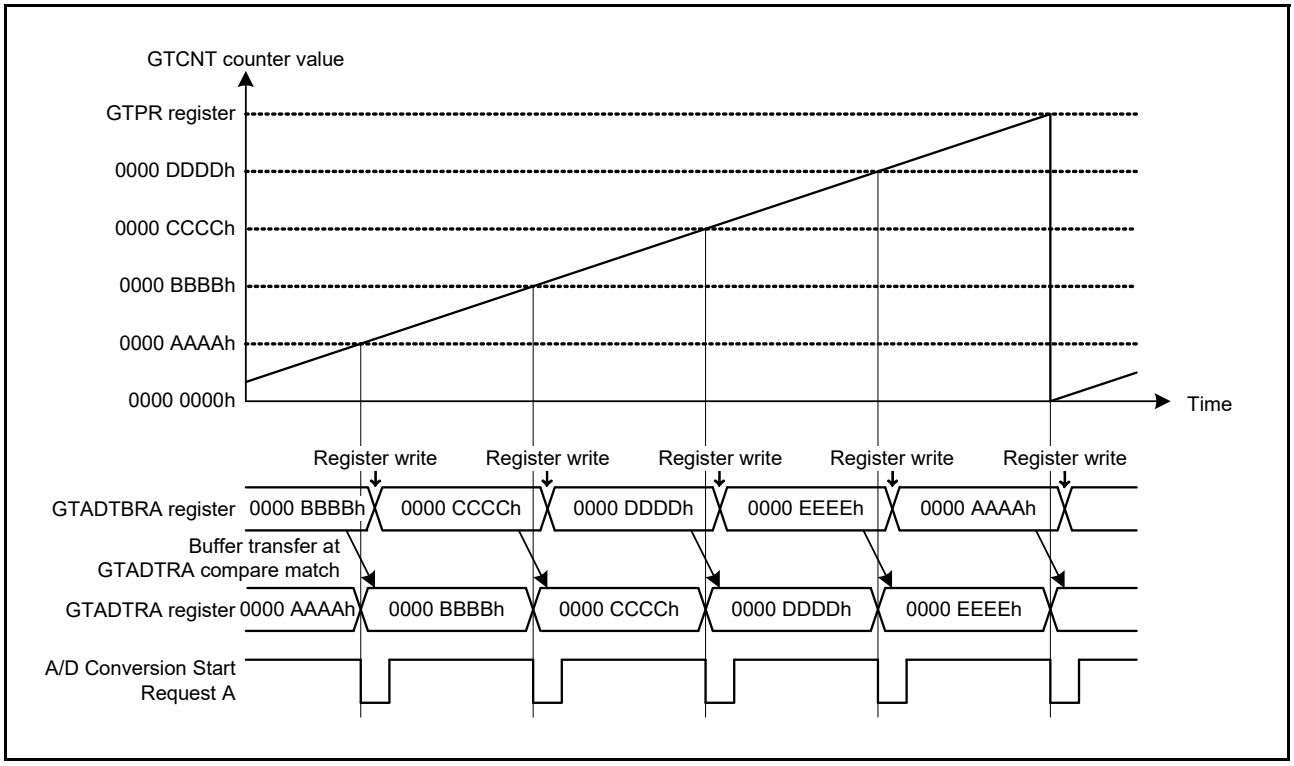


Figure 24.41 Example of Buffer Operation of the GTADTRA and GTADTRB Registers (Sawtooth Waves in Up-Counting, A/D Conversion Start Request Generated by Up-Counting)

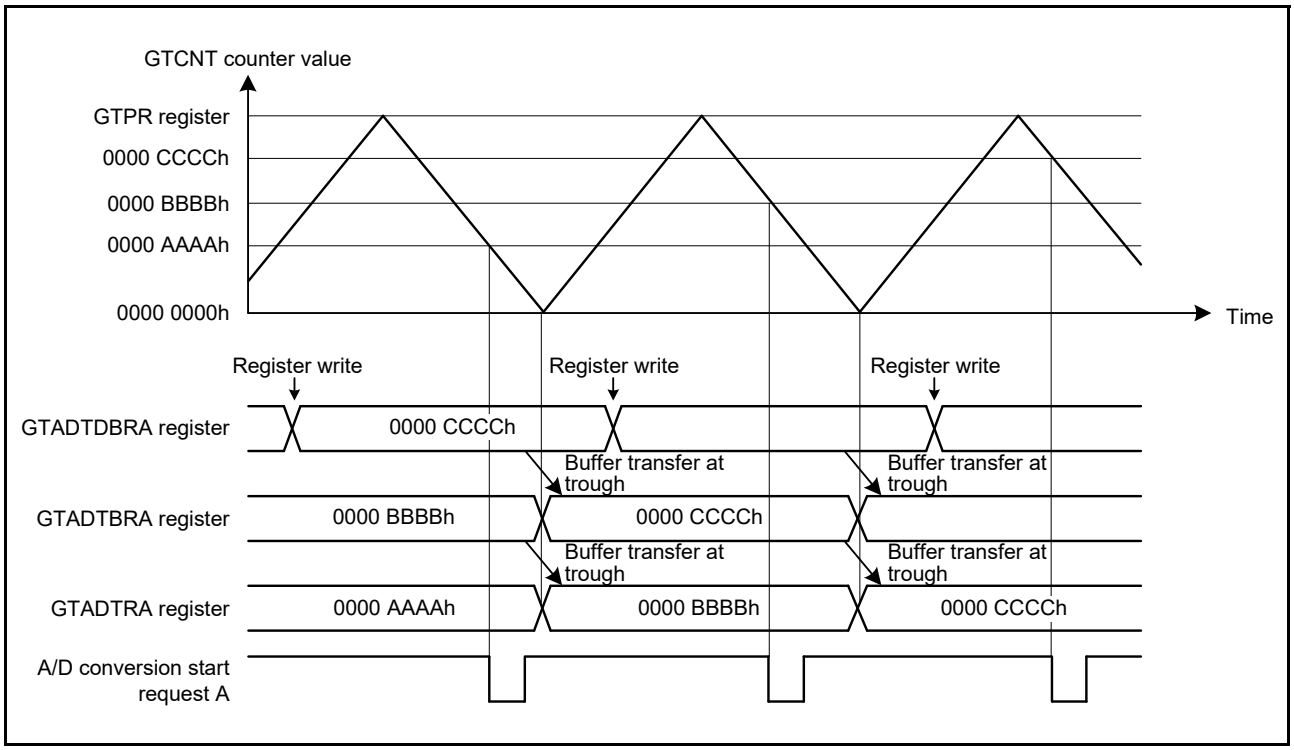


Figure 24.42 Example of Buffer Operation of the GTADTRA and GTADTRB Registers (Sawtooth Waves in Up-Counting, Buffer Transfer at GTADTRA Compare Match, A/D Conversion Start Request Generated by Up-Counting)

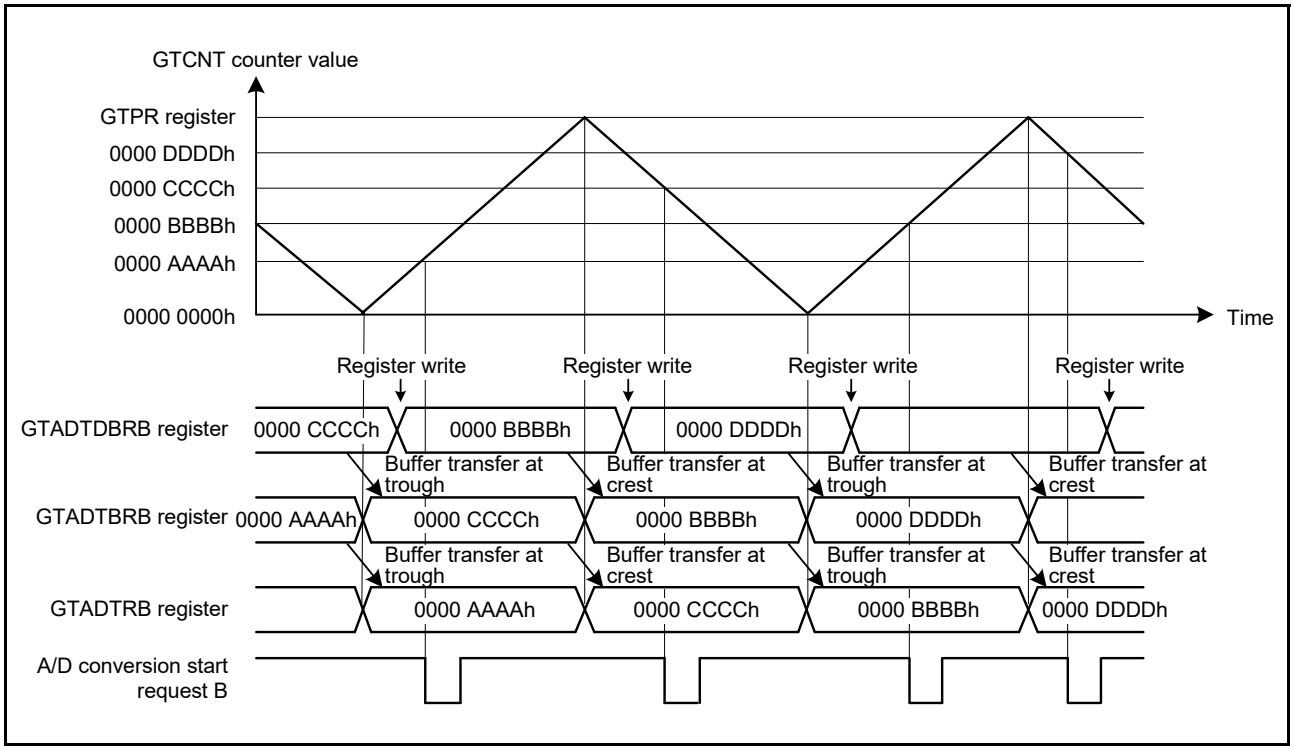


Figure 24.43 Example of Double Buffer Operation of the GTADTRA and GTADTRB Registers (Triangle Waves, Buffer Transfer at Both Troughs and Crests, A/D Conversion Start Request Generated by Both Up- and Down-Counting)

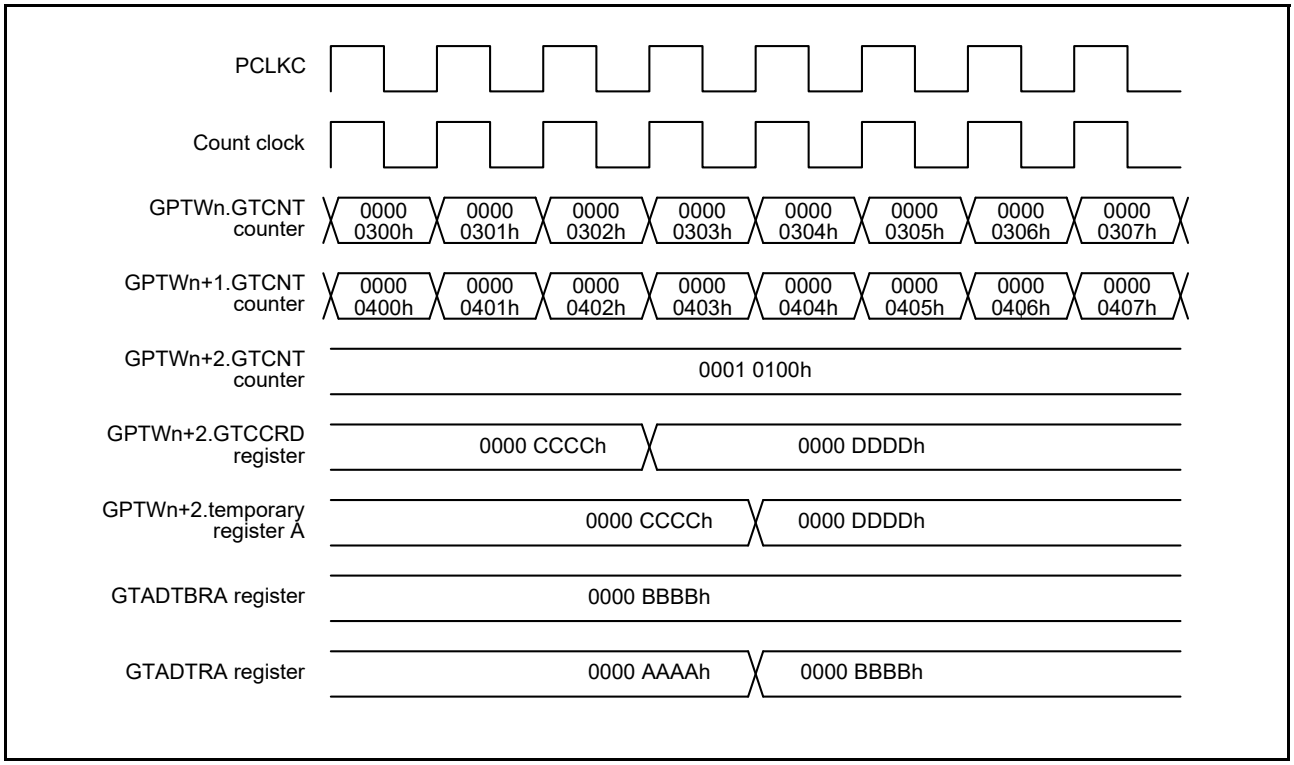


Figure 24.44 Example of Buffer Operation of the GTADTRA and GTADTRB Registers at the GTCCRD Register of Slave Channel 2 Updating in Complementary PWM Mode

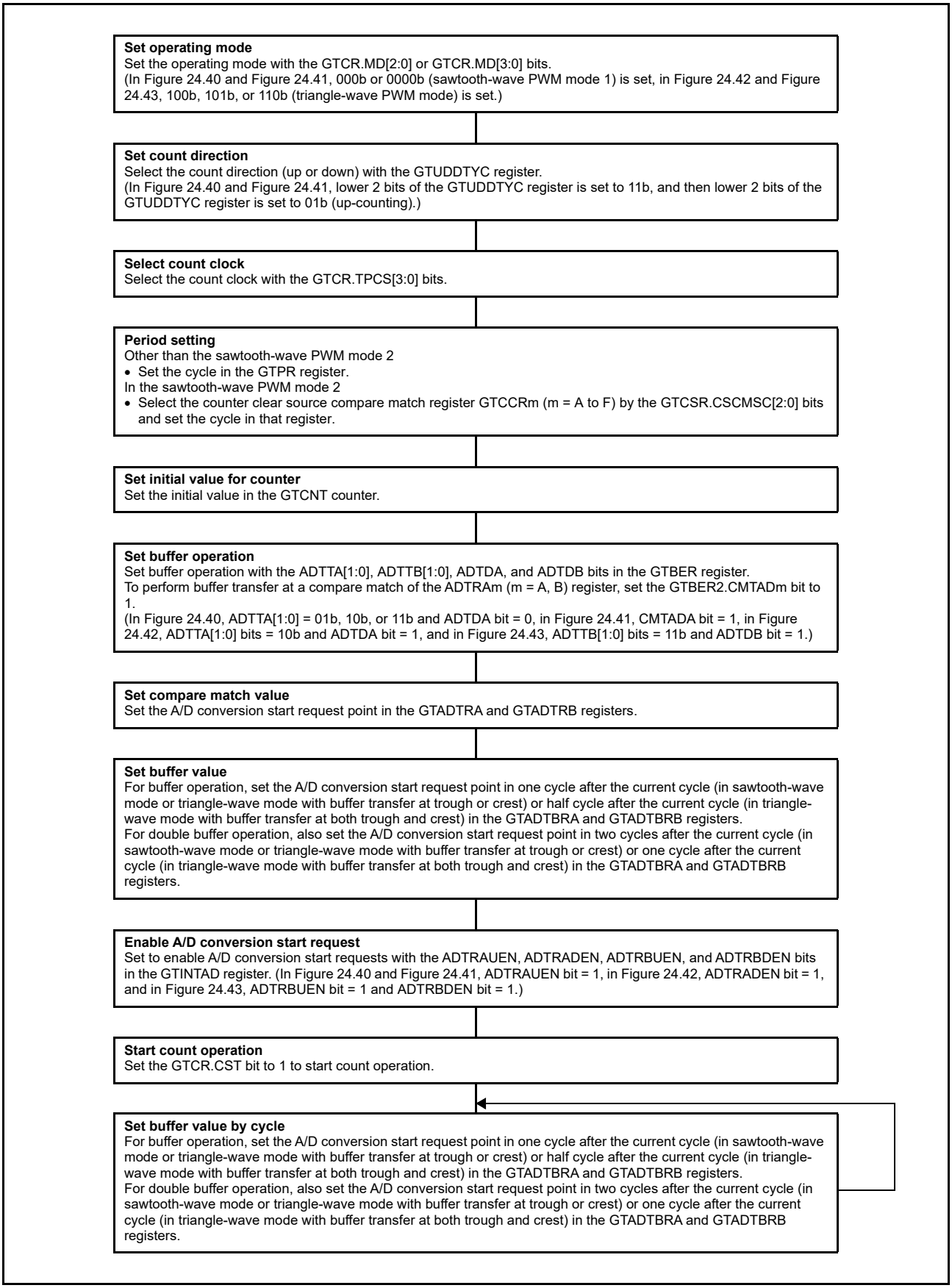
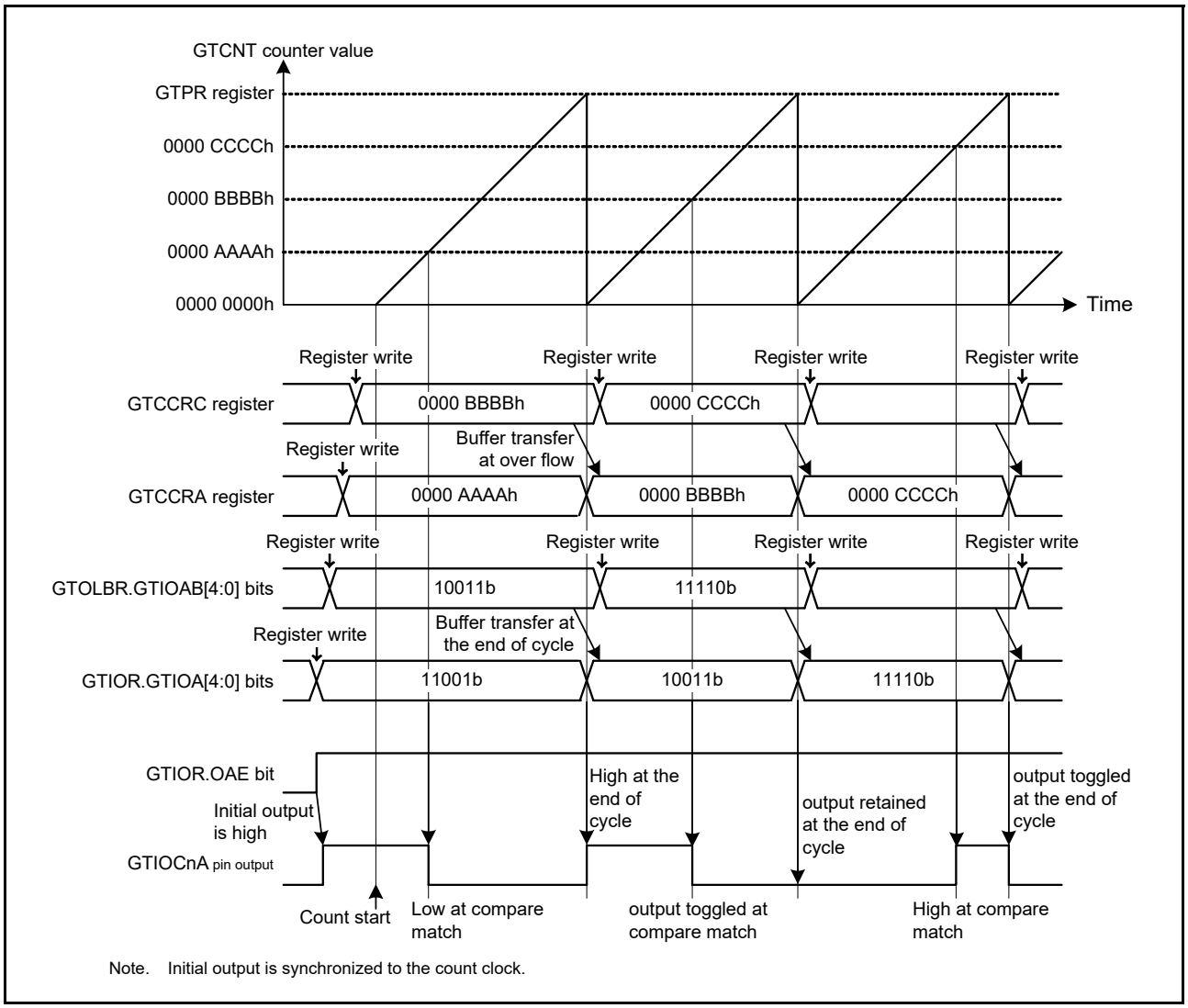


Figure 24.45 Example for Setting Buffer Operation of the GTADTRA and GTADTRB Registers

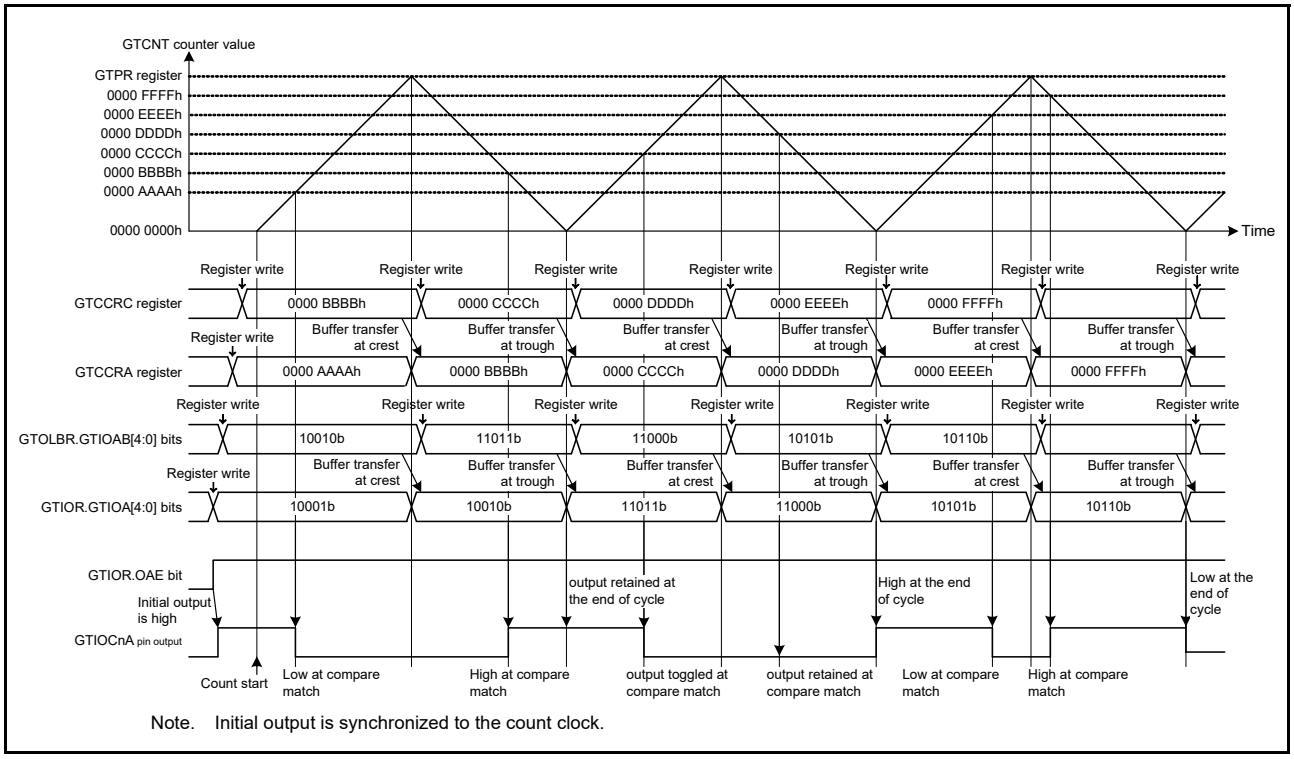
24.3.2.4 Buffer Operation for the GTIOA[4:0] and GTIOB[4:0] Bits

The GTOLBR.GTIOAB[4:0] bits can function as the buffer register of the GTIOR.GTIOA[4:0] bits and the GTOLBR.GTIOBB[4:0] bits can function as the buffer register of the GTIOR.GTIOB[4:0] bits.
Buffer transfer timing can be set with the GTBER2.OLTTm[1:0] bits (m = A, B). It can be selected from the end of cycle or GTCCR register compare match (in sawtooth-wave mode), from crest, trough, or both crest and trough (in triangle-wave mode and complementary PWM mode). In the case that the GTBER2.OLTTm[1:0] bits are 00b, buffer transfer is not performed.
In complementary PWM mode, it is prohibited to set the buffer transfer timing to overlap with the dead time. Therefore, when the buffer transfer timing is crest, set the GTCCRm (m = A, C, E) register to satisfy GTCCRm < GTPR. And when the buffer transfer timing is trough, set the GTCCRm register to satisfy GTDVU < GTCCRm.
Figure 24.46 to Figure 24.48 show examples of buffer operation of the GTIOR.GTIOA[4:0] and GTIOR.GTIOB[4:0] bits. Figure 24.49 shows an example for setting buffer operation of the GTIOR.GTIOA[4:0] and GTIOR.GTIOB[4:0] bits.



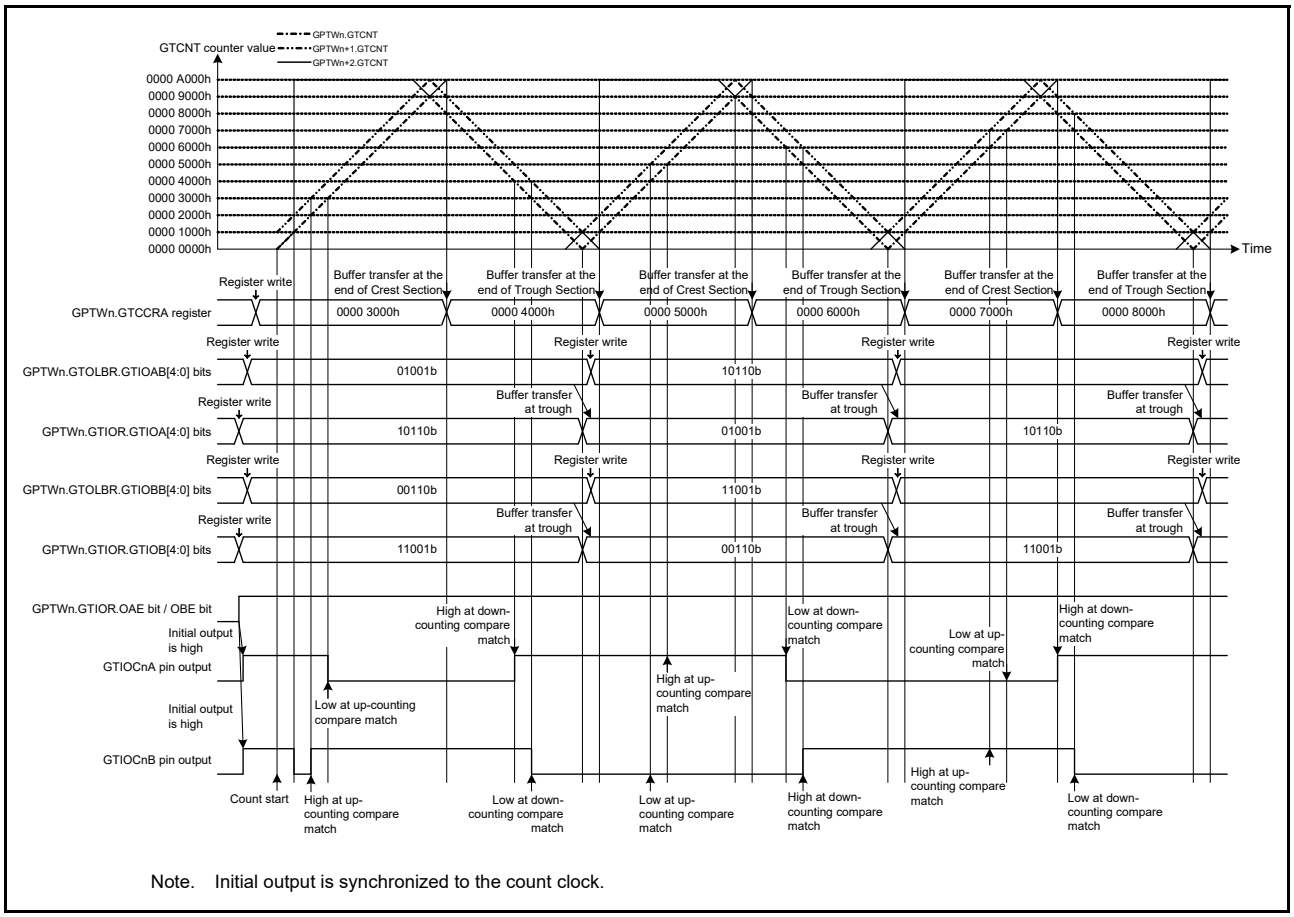
Note. Initial output is synchronized to the count clock.

Figure 24.46 Example of Buffer Operation of the GTIOA[4:0] and GTIOB[4:0] Bits (Up-Counting in Sawtooth-Wave PWM Mode 1, Buffer Transfer at the End of Period)



Note. Initial output is synchronized to the count clock.

Figure 24.47 Example of Buffer Operation of the GTIOA[4:0] and GTIOB[4:0] Bits (Triangle-Waves PWM Mode 2, Buffer Transfer at Both Troughs and Crests)



Note. Initial output is synchronized to the count clock.

Figure 24.48 Example of Buffer Operation of the GTIOA[4:0] and GTIOB[4:0] Bits (Complementary PWM Mode 3, Buffer Transfer at Troughs)

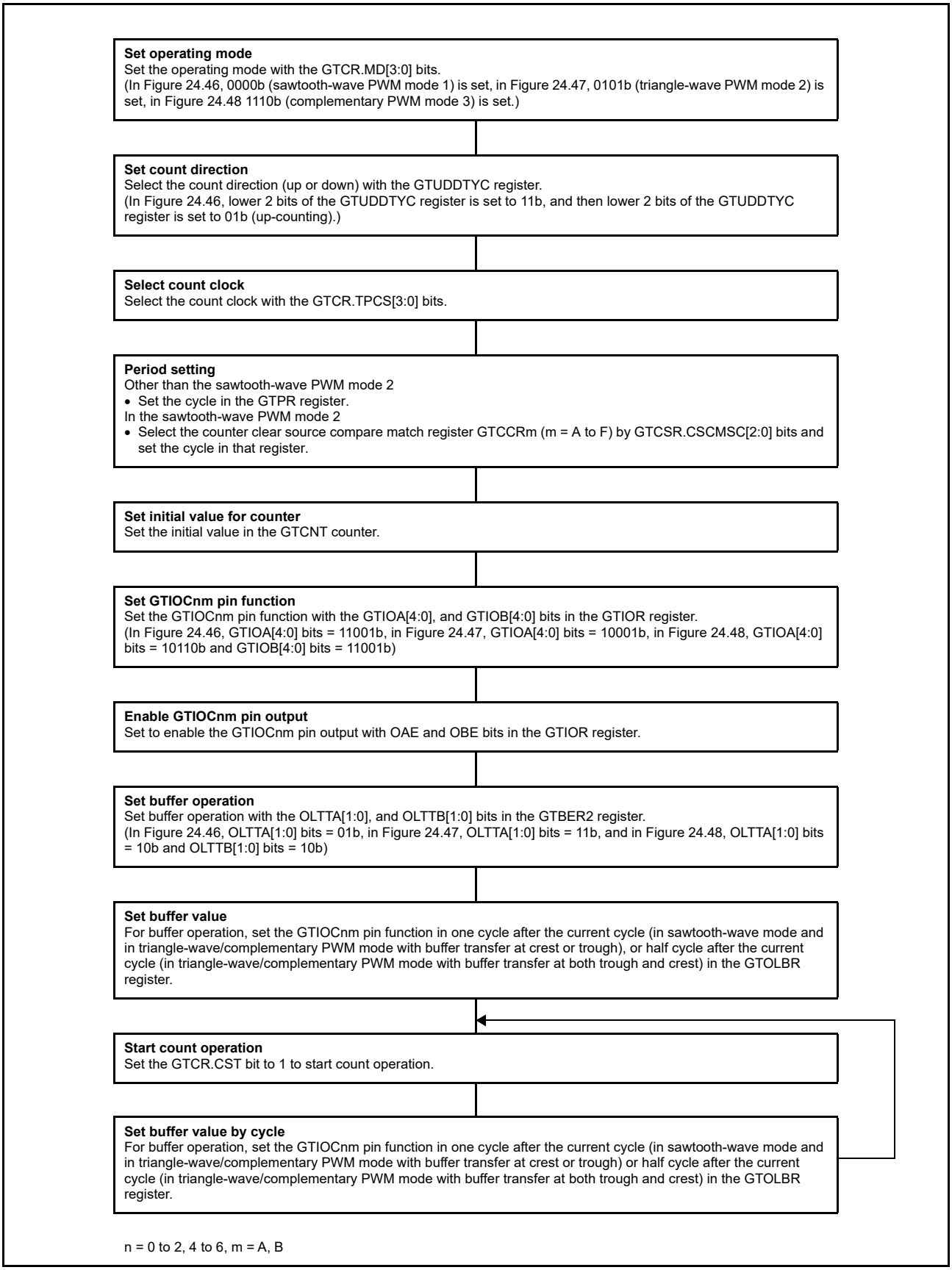


Figure 24.49 Example for Setting Buffer Operation of the GTIOA[4:0] and GTIOB[4:0] Bits

24.3.3 PWM Output Operating Mode

The GPTW can output PWM waveforms to the GTIOCnA pin or GTIOCnB pin (n = 0 to 7) by a compare match between the GTCNT counter and the GTCCRA or GTCCRB register.
By setting the GTDTCR, GTDVU, and GTDVD registers, the compare match value for a negative-phase waveform with dead time can automatically be set to the GTCCRB register.
In complementary PWM mode, it is possible to output PWM waveforms (positive phase, negative phase) with dead time that guarantees the linearity of the PWM output pulse width near 0% and 100% duty.
In sawtooth-wave mode other than sawtooth-wave PWM mode 2, or triangle-wave mode, or the master channel of complementary PWM mode, the signal synchronized with the PWM cycle can be output from the GTCPPOn output terminal by setting the GTIOR.PSYE bit to 1. The GTCPPOn output is toggled at the end of cycle in sawtooth-wave mode or at the timing of crest/trough/GTCNT counter clearing in triangle-wave mode or complementary PWM mode. The initial output of GTCPPOn is low, and it becomes high when the count starts.

(1) Sawtooth-Wave PWM Mode 1

In sawtooth-wave PWM mode 1, the GTCNT counter performs sawtooth-wave (half-wave) operation by setting the period in the GTPR register and a PWM waveform is output to the GTIOCnA or GTIOCnB pin when a GTCCRA or GTCCRB register compare match occurs. The pin output value can be selected from low output, high output, or toggle output separately for a compare match and for the end of the cycle according to the GTIOR register setting.
When 0 is set in the GTIOR.OmEOCD (m = A, B) bit and the timing of the end of cycle and the timing of the GTCCRm register compare match are the same time, the output pin performs along the PWM output setting for the end of cycle set by the GTIOR.GTIOm[3:2] bits.
When 1 is set in the GTIOR.OmEOCD bit, GTIOCnm output is retained.
Figure 24.50 shows an example of sawtooth-wave PWM mode 1 operation, and Figure 24.51 shows an example for setting sawtooth-wave PWM mode 1.

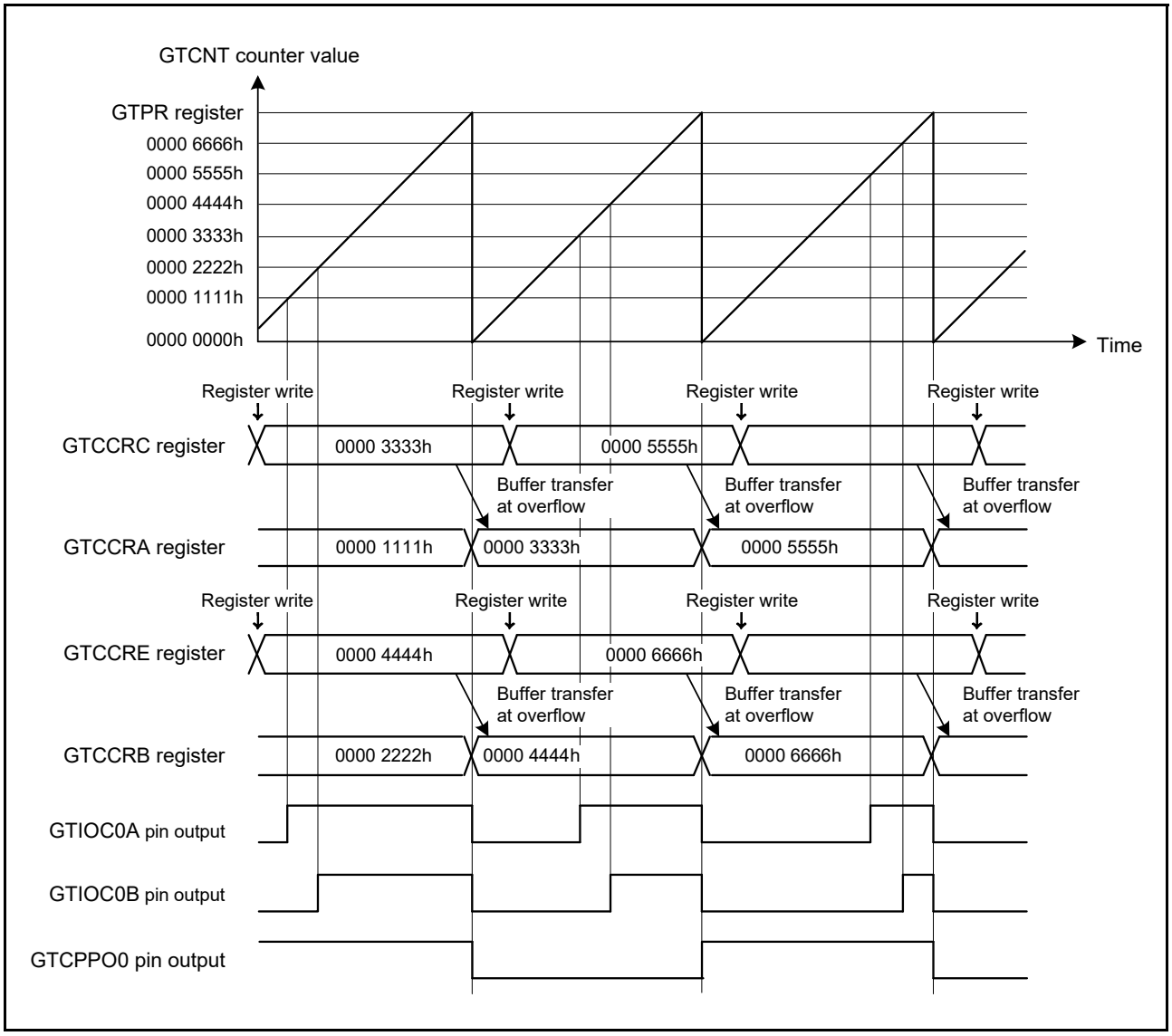
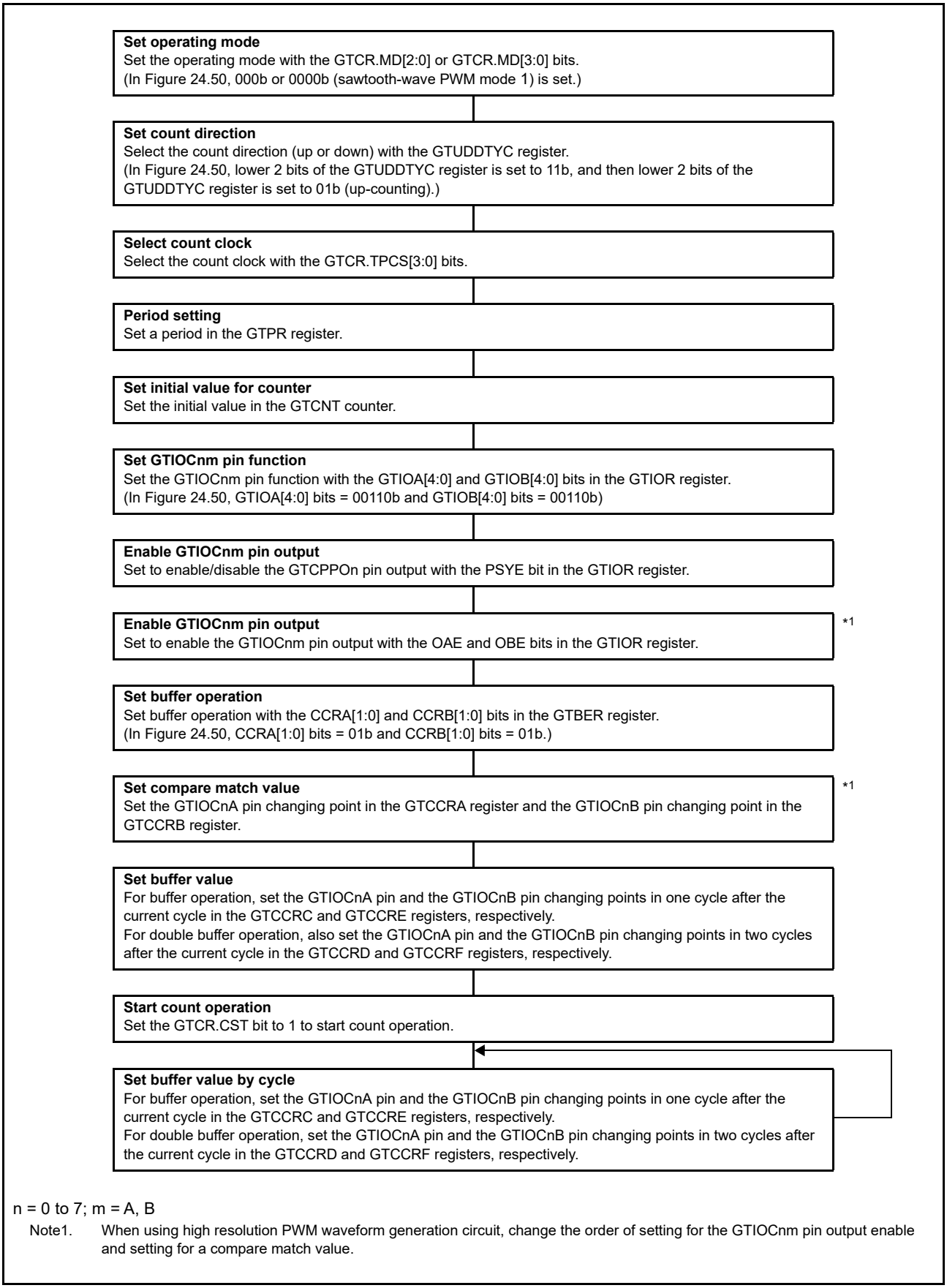


Figure 24.50 Example of Sawtooth-Wave PWM Mode Operation (Up-Counting, Buffer Operation, High Output at GTCCRA/GTCCRB Register Compare Match, Low Output at the End of the Cycle, GTIOR.PSYE = 1) (n = 0 to 7)



n = 0 to 7; m = A, B

Note1. When using high resolution PWM waveform generation circuit, change the order of setting for the GTIOCnm pin output enable and setting for a compare match value.

Figure 24.51 Example for Setting Sawtooth-Wave PWM Mode 1

(2) Sawtooth-Wave PWM Mode 2

The sawtooth-wave PWM mode 2 is a mode in which the GTCNT counter is operated as a sawtooth wave by up-counting without using the GTPR register, and the PWM waveform is output by the compare match of the GTCCRA and GTCCRB registers.

The pin output level can be selected from low output, high output, or toggle output separately for a compare match according to the GTIOR register setting.

The GTIOCnA (n = 0 to 2, 4 to 6) pin is used as an output pin. Use the GTIOR.GTIOB[1:0] bits for setting the GTIOCnA pin output at a compare match of the GTCCRB register.

When a counter clear occurs due to the GTCNT counter clearing source selected in the GTCSR register, this is handled at the end of cycle and PWM output operation at the end of the cycle selected with the GTIOR.GTIOA[3:2] bits is performed.

If a counter clear (at the end of cycle) conflicts with a PWM output change due to a GTCCRm (m = A, B) register compare match, PWM output operation is performed at the end of cycle (in the case of the GTIOR.OmEOCD bit = 0) or the PWM output is retained (in the case of GTIOR.OmEOCD bit = 1).

Figure 24.52 to Figure 24.54 show examples of sawtooth-wave PWM mode 2 operation. Figure 24.55 shows an example for setting sawtooth-wave PWM mode 2.

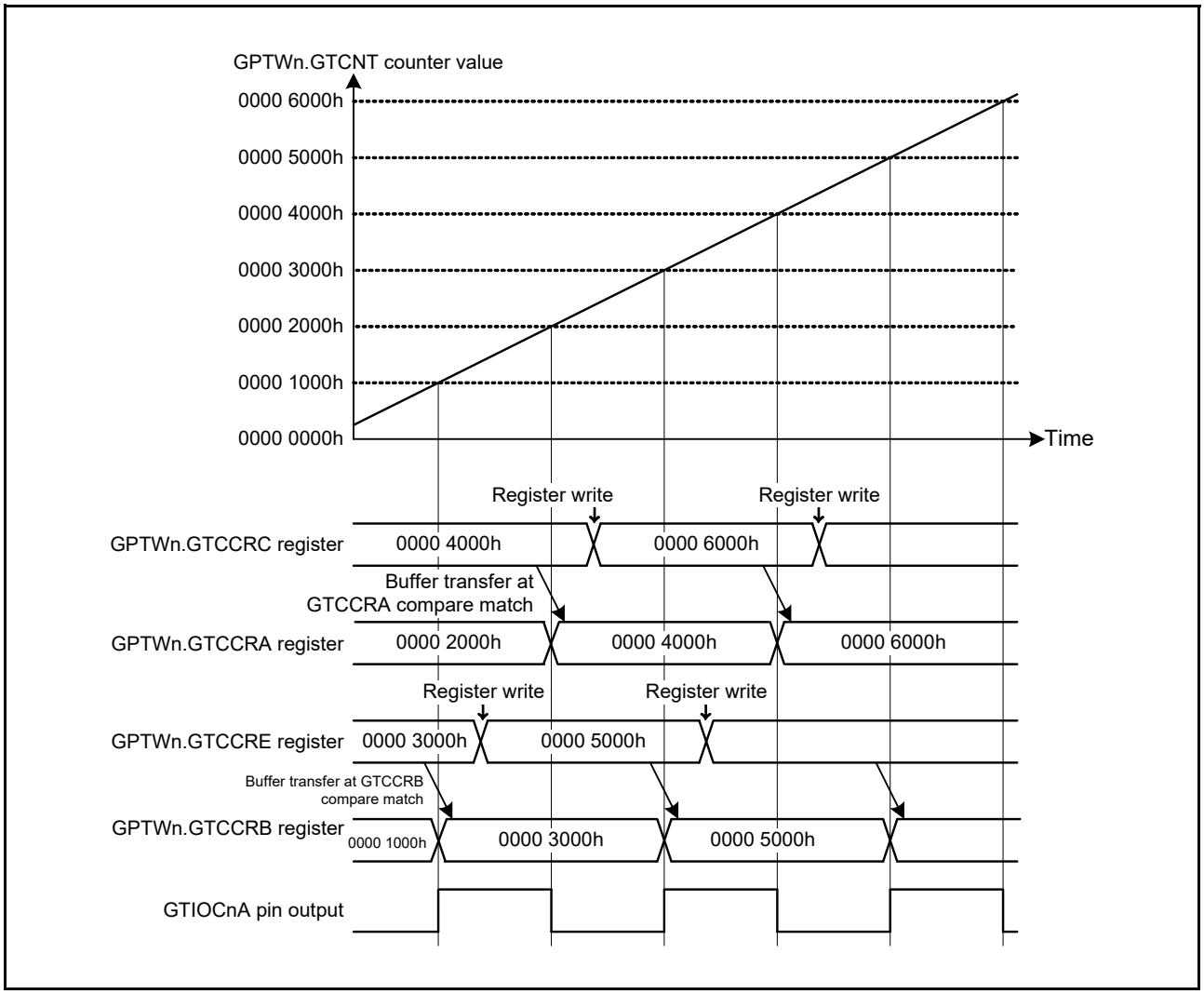


Figure 24.52 Example of Sawtooth-Wave PWM Mode 2 Operation (Low Output GTCCRA Register Compare Match, High Output at the GTCCRB Register Compare Match, Single Buffer Operation, and No Clear Setting) (n = 0 to 2, 4 to 6)

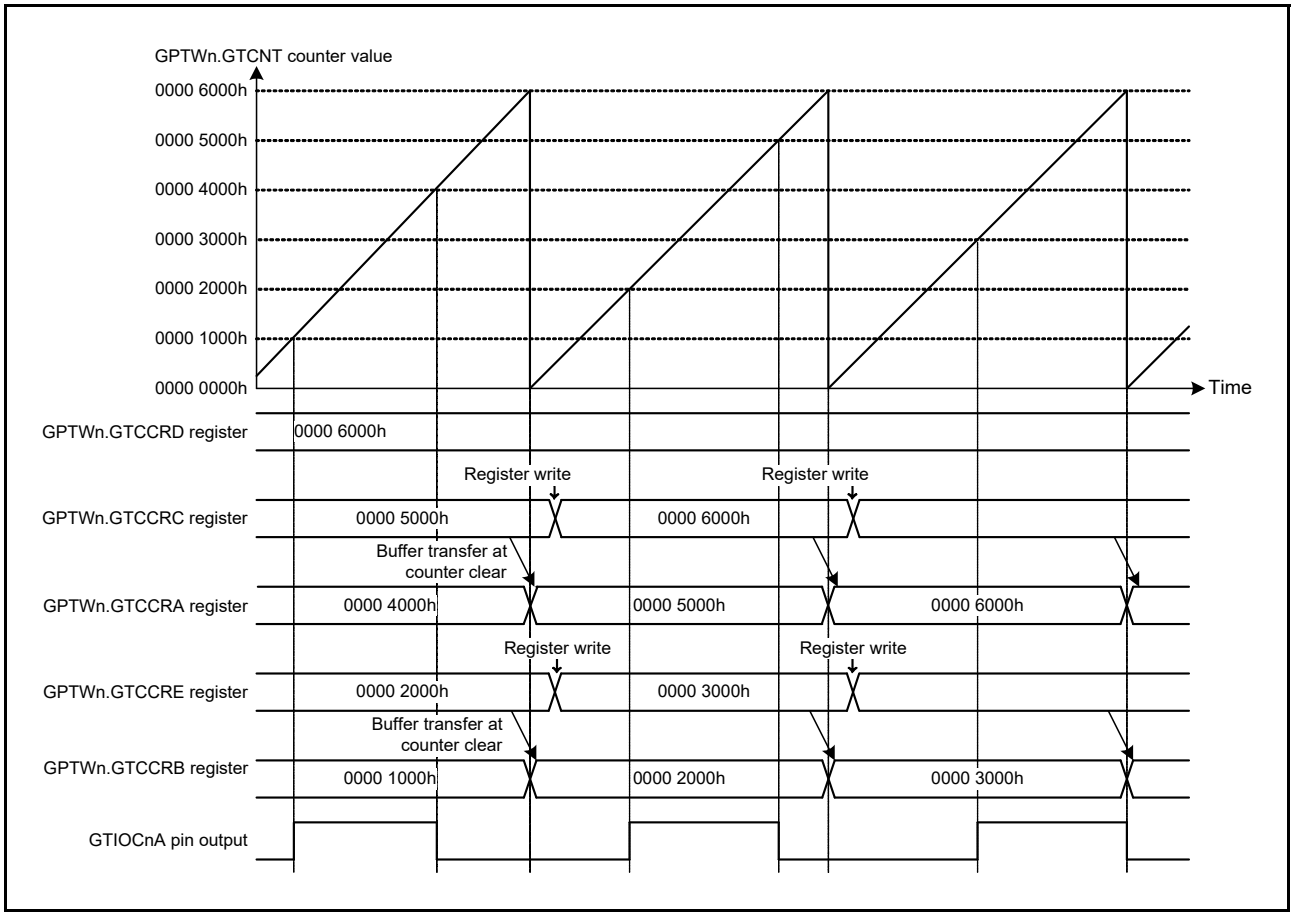


Figure 24.53 Example of Sawtooth-Wave PWM Mode 2 Operation (Low Output at the GTCCRA Register Compare Match, High Output at the GTCCRB Register Compare Match, Low Output at the End of the Cycle, Single Buffer Operation, Cleared at the GTCCRD Register Compare Match, and the GTIOR.OAEOCD Bit = 0) (n = 0 to 2, 4 to 6)

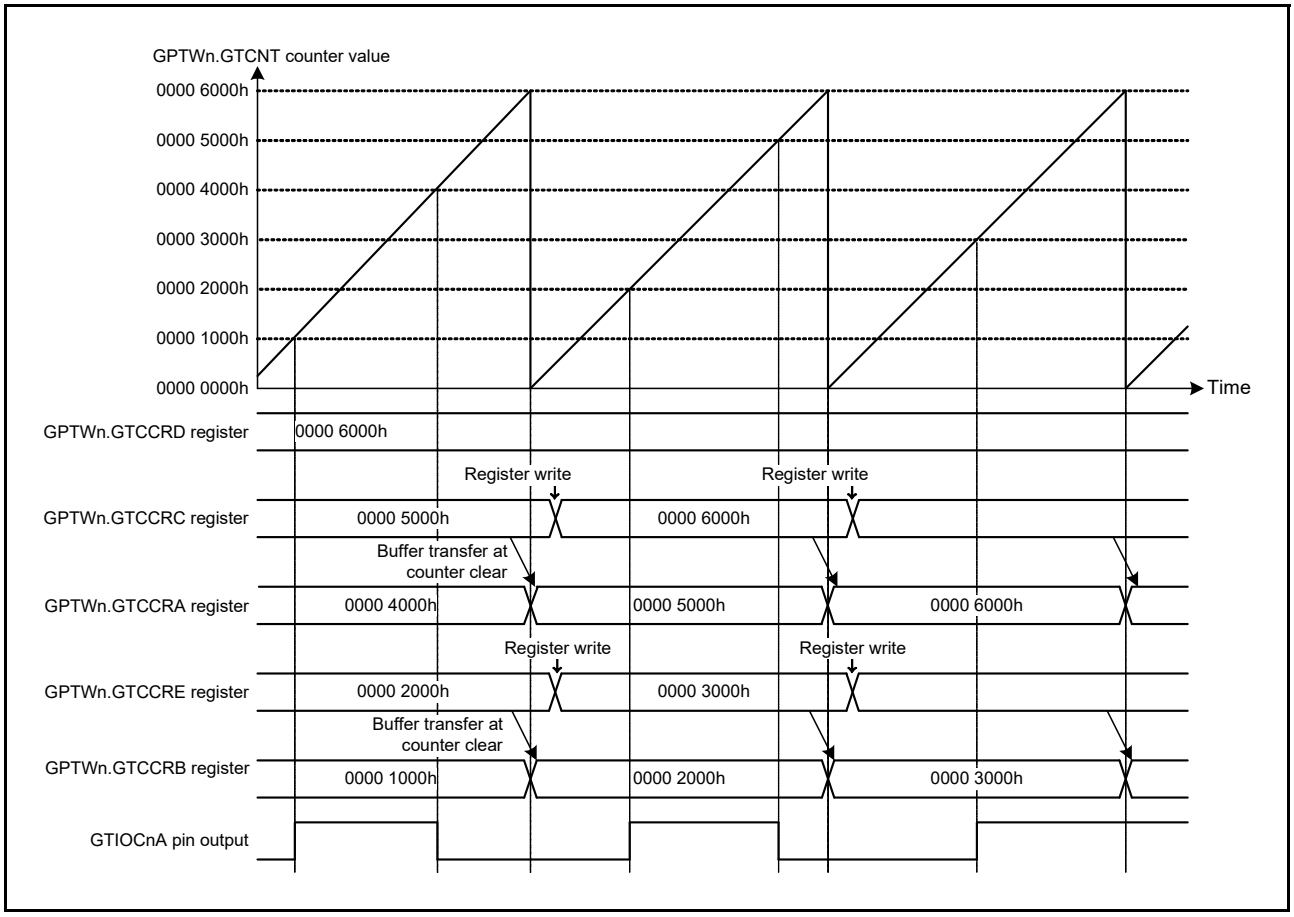
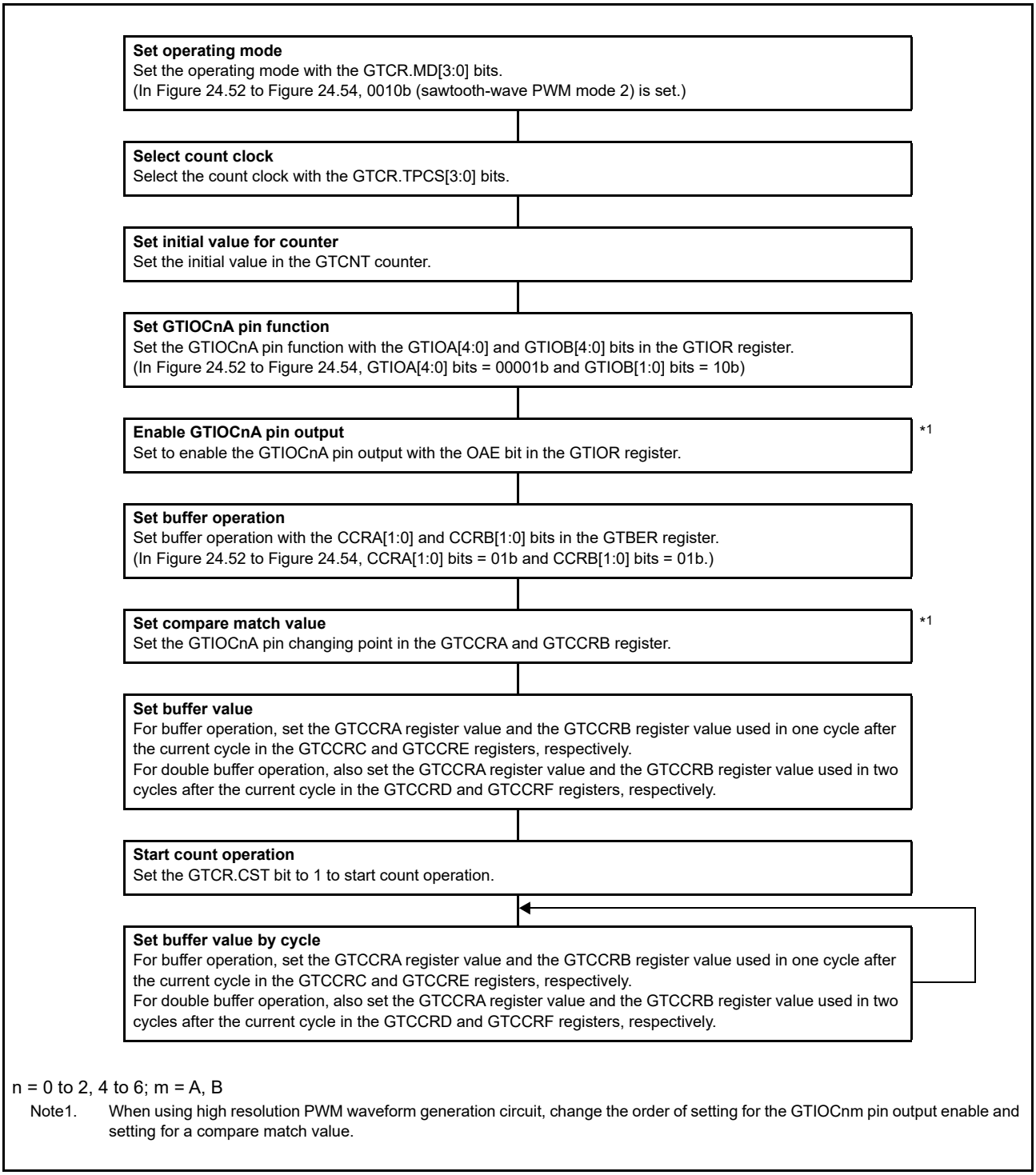


Figure 24.54 Example of Sawtooth-Wave PWM Mode 2 Operation (Low Output at the GTCCRA Register Compare Match, High Output at the GTCCRB Register Compare Match, Low Output at the End of Cycle, Single Buffer Operation, Cleared at the GTCCRD Register Compare Match, and GTIOR.OAEOCD Bit = 1) (n = 0 to 2, 4 to 6)



n = 0 to 2, 4 to 6; m = A, B

Note1. When using high resolution PWM waveform generation circuit, change the order of setting for the GTIOCnm pin output enable and setting for a compare match value.

Figure 24.55 Example for Setting Sawtooth-Wave PWM Mode 2

(3) Sawtooth-Wave One-Shot Pulse Mode

The sawtooth-wave one-shot pulse mode is a mode in which the period is set in the GTPR register, the GTCNT counter performs sawtooth-wave (half-wave) operation and a PWM waveform is output to the GTIOCnA or GTIOCnB pin at a compare match of the GTCCRA or GTCCRB register with buffer operation fixed (n = 0 to 7).

Buffer operation in sawtooth-wave one-shot pulse mode is different from the usual buffer operation. Buffer transfer is performed from the GTCCRC register to the GTCCRA register, from the GTCCRE register to the GTCCRB register, from the GTCCRD register to the temporary register A, and from the GTCCRF register to the temporary register B at the end of the cycle, and from the temporary register A to the GTCCRA register at a GTCCRA register compare match and from the temporary register B to the GTCCRB register at a GTCCRB register compare match. The pin output value can be selected from low output, high output, or toggle output separately for a compare match and the end of the cycle according to the GTIOR register setting.

The temporary register A and the temporary register B can perform forcible buffer transfers from the GTCCRD register to the temporary register A and from the GTCCRF register to the temporary register B respectively by writing 1 to the GTBER.CCRSWT bit while the count is stopped.

By setting the GTDTCR, GTDVU, and GTDVD registers, a compare match value for a negative-phase waveform with dead time can automatically be set to the GTCCRB register.

When the GTBER.DBRTECm (m = A, B) bit is set to 1, transfer from an intermediate buffer to the GTCCRm (m = A, B) register is repeated on a cyclic basis with using the temporary register x (x = C, E) and temporary register m (m = A, B) which operate as intermediate buffers for the GTCCRx (x = C, E) and GTCCRm (m = A, B) registers, respectively, even while buffer transfer is disabled (repeated double buffer operation function during disabling of buffer transfer). For details, refer to section 24.8.2.2, Repeated Double-Buffered Operation When Disabling GTCCR Buffer Transfer.

Figure 24.56 shows an example of sawtooth-wave one-shot pulse mode operation, and Figure 24.57 shows an example for setting sawtooth-wave one-shot pulse mode.

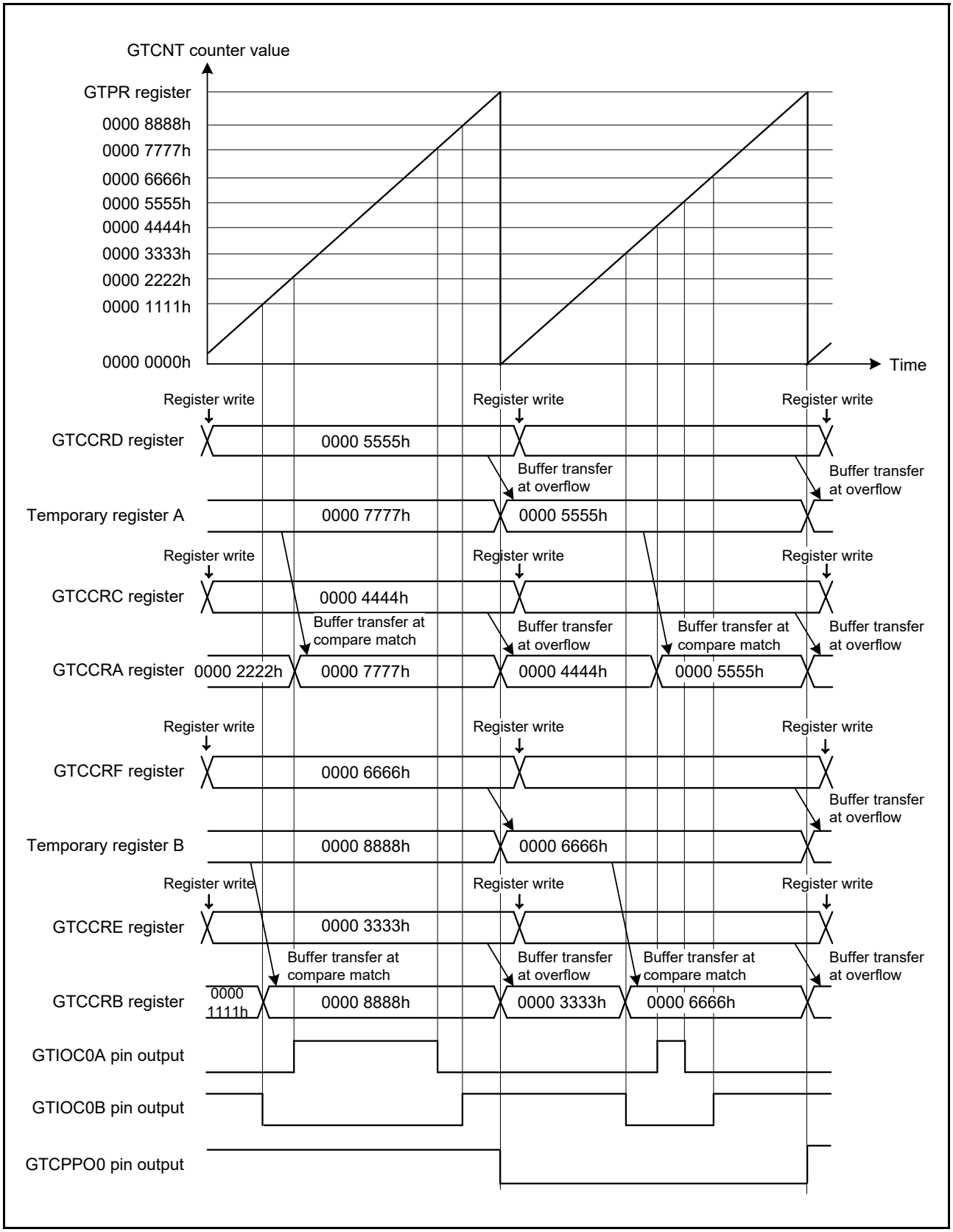
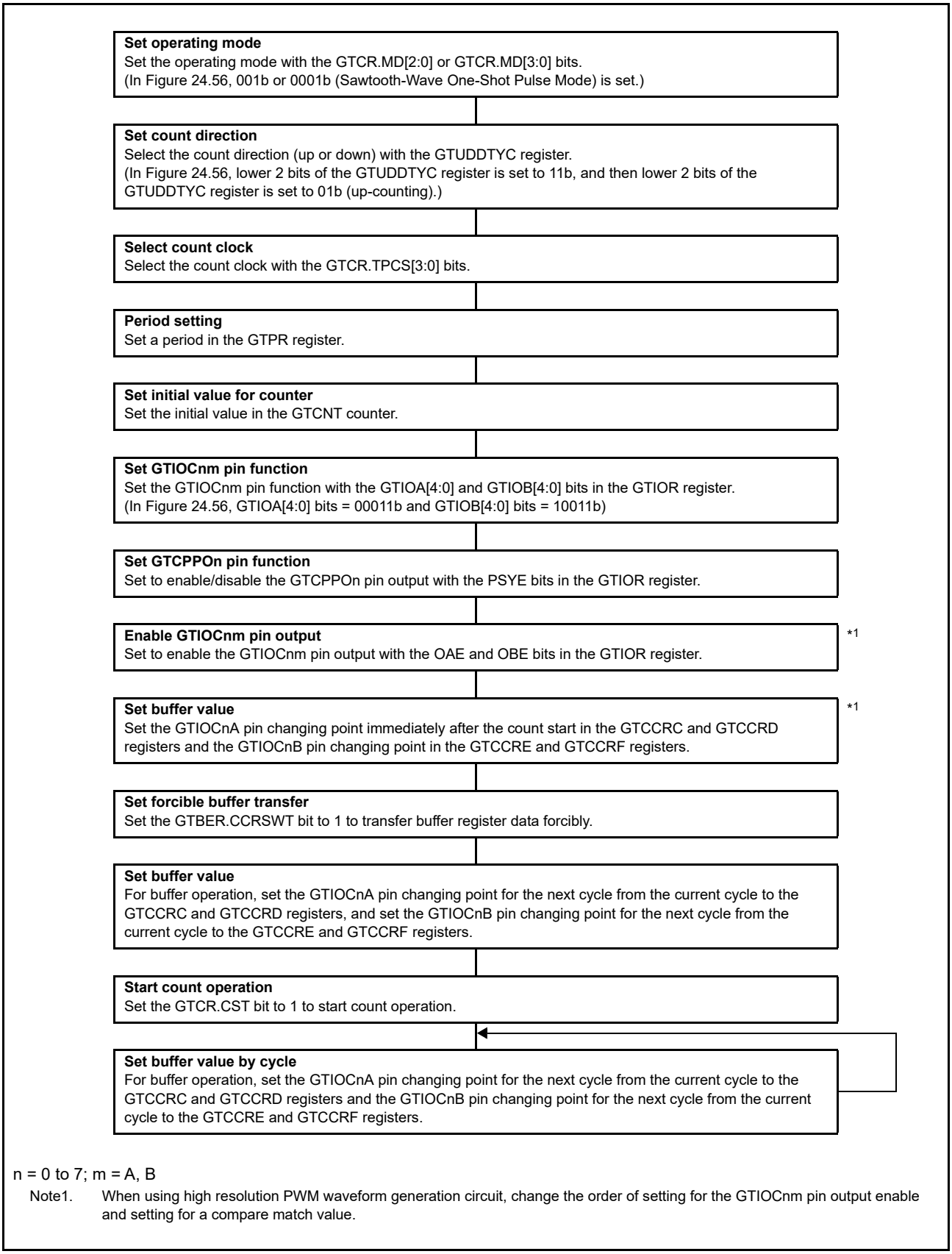


Figure 24.56 Example of Sawtooth-Wave One-Shot Pulse Mode Operation (Up-Counting, Low Output from the GTIOCnA Pin and High Output from the GTIOCnB Pin at Initial Output, Output Toggled at GTCCRA/GTCCRB Register Compare Match, Output Retained at the End of the Cycle, GTIOR.PSYE = 1) (n = 0 to 7)



n = 0 to 7; m = A, B

Note1. When using high resolution PWM waveform generation circuit, change the order of setting for the GTIOCnm pin output enable and setting for a compare match value.

Figure 24.57 Example for Setting Sawtooth-Wave One-Shot Pulse Mode

(4) Triangle-Wave PWM Mode 1 (32-Bit Transfer at Trough)

The triangle-wave PWM mode 1 is a mode in which the period is set in the GTPR register, the GTCNT counter performs triangle-wave (full-wave) operation, and a PWM waveform is output to the GTIOCnA or GTIOCnB pin when a GTCCRA or GTCCRB register compare match occurs (n = 0 to 7).

Buffer transfer is performed at the trough. The pin output value can be selected from low output, high output, or toggle output separately for a compare match and for the end of the cycle according to the GTIOR register setting.

By setting the GTDTCR, GTDVU, and GTDVD registers, a compare match value for a negative-phase waveform with dead time can automatically be set to the GTCCRB register.

Figure 24.58 shows an example of triangle-wave PWM mode 1 operation, and Figure 24.59 shows an example for setting triangle-wave PWM mode 1.

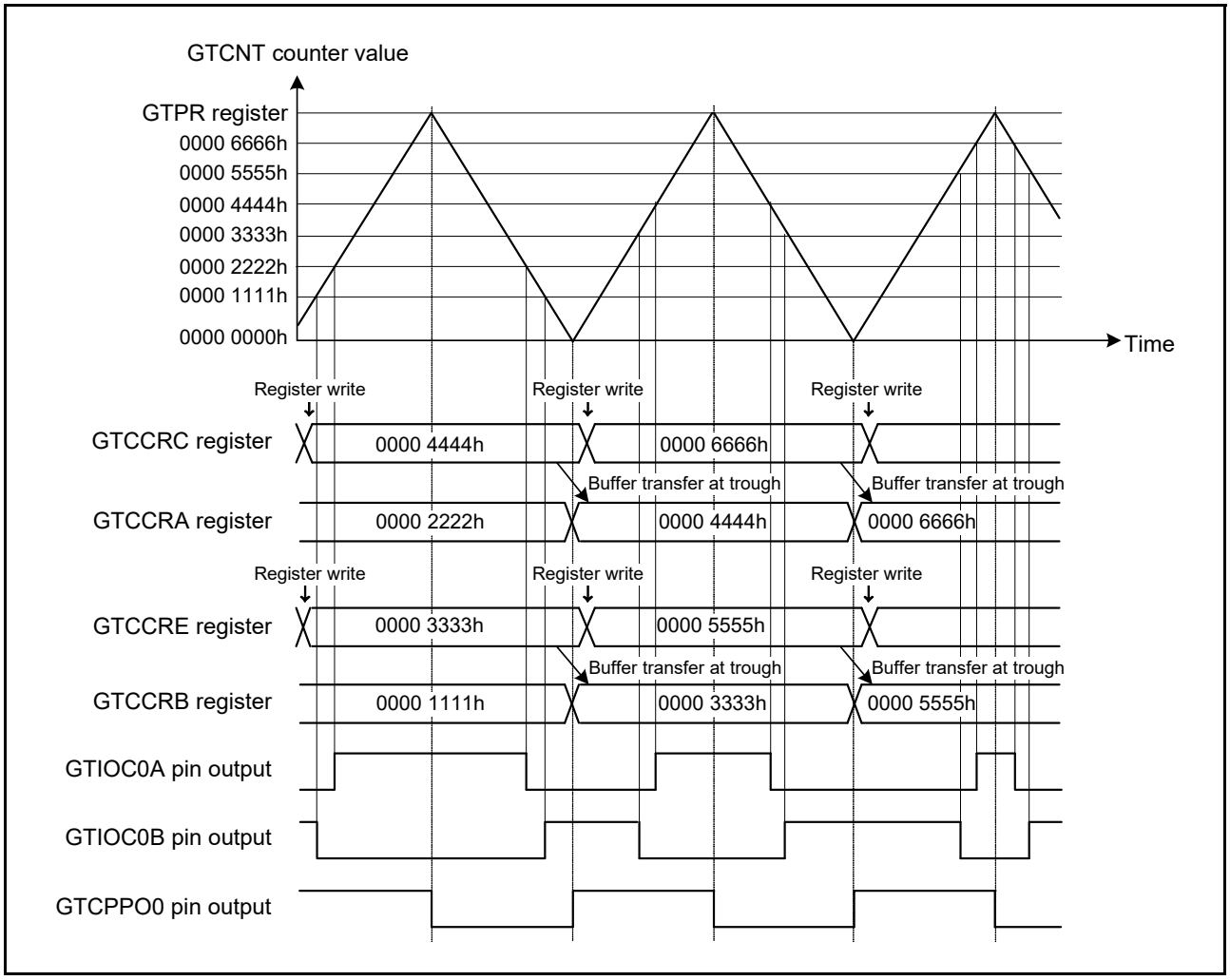
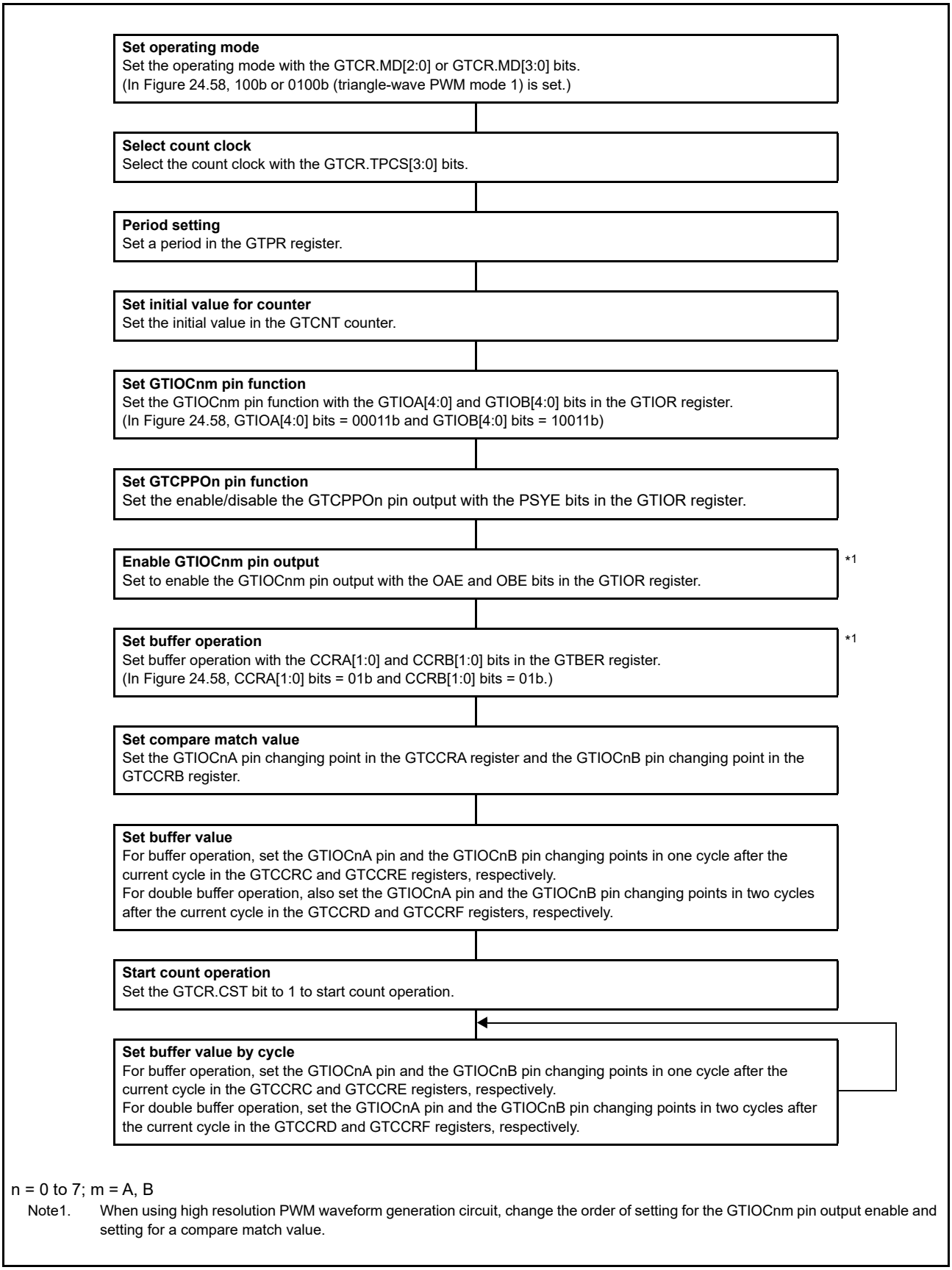


Figure 24.58 Example of Triangle-Wave PWM Mode 1 Operation (Buffer Operation, Low Output from the GTIOCnA Pin and High Output from the GTIOCnB Pin at Initial Output, Output Toggled at GTCCRA/GTCCRB Register Compare Match, Output Retained at the End of the Cycle, and GTIOR.PSYE = 1) (n = 0 to 7)



n = 0 to 7; m = A, B

Note1. When using high resolution PWM waveform generation circuit, change the order of setting for the GTIOCnm pin output enable and setting for a compare match value.

Figure 24.59 Example for Setting Triangle-Wave PWM Mode 1

(5) Triangle-Wave PWM Mode 2 (32-Bit Transfer at Crest and Trough)

Similarly to triangle-wave PWM mode 1, in triangle-wave PWM mode 2 the period is set in the GTPR register, the GTCNT counter performs triangle-wave (full-wave) operation, and a PWM waveform is output to the GTIOCnA or GTIOCnB pin when a GTCCRA or GTCCRB register compare match occurs (n = 0 to 7). The buffer transfer is performed at both crests and troughs. The pin output value can be selected from low output, high output, or toggle output separately for a compare match and for the end of the cycle according to the GTIOR register setting.
By setting the GTDTCR, GTDVU, and GTDVD registers, a compare match value for a negative-phase waveform with dead time can automatically be set to the GTCCRB register.
Figure 24.60 shows an example of triangle-wave PWM mode 2 operation, and Figure 24.61 shows an example for setting triangle-wave PWM mode 2.

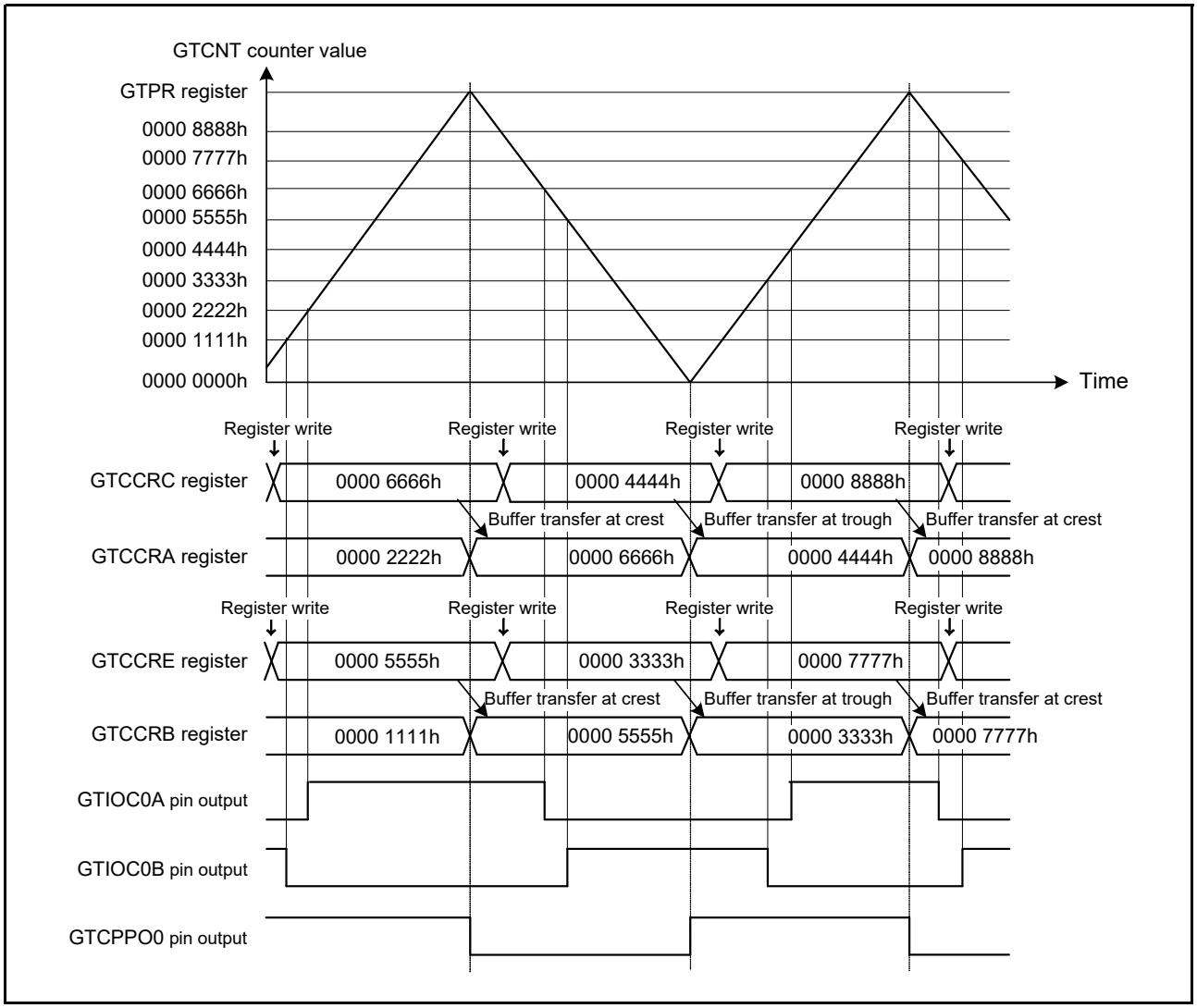
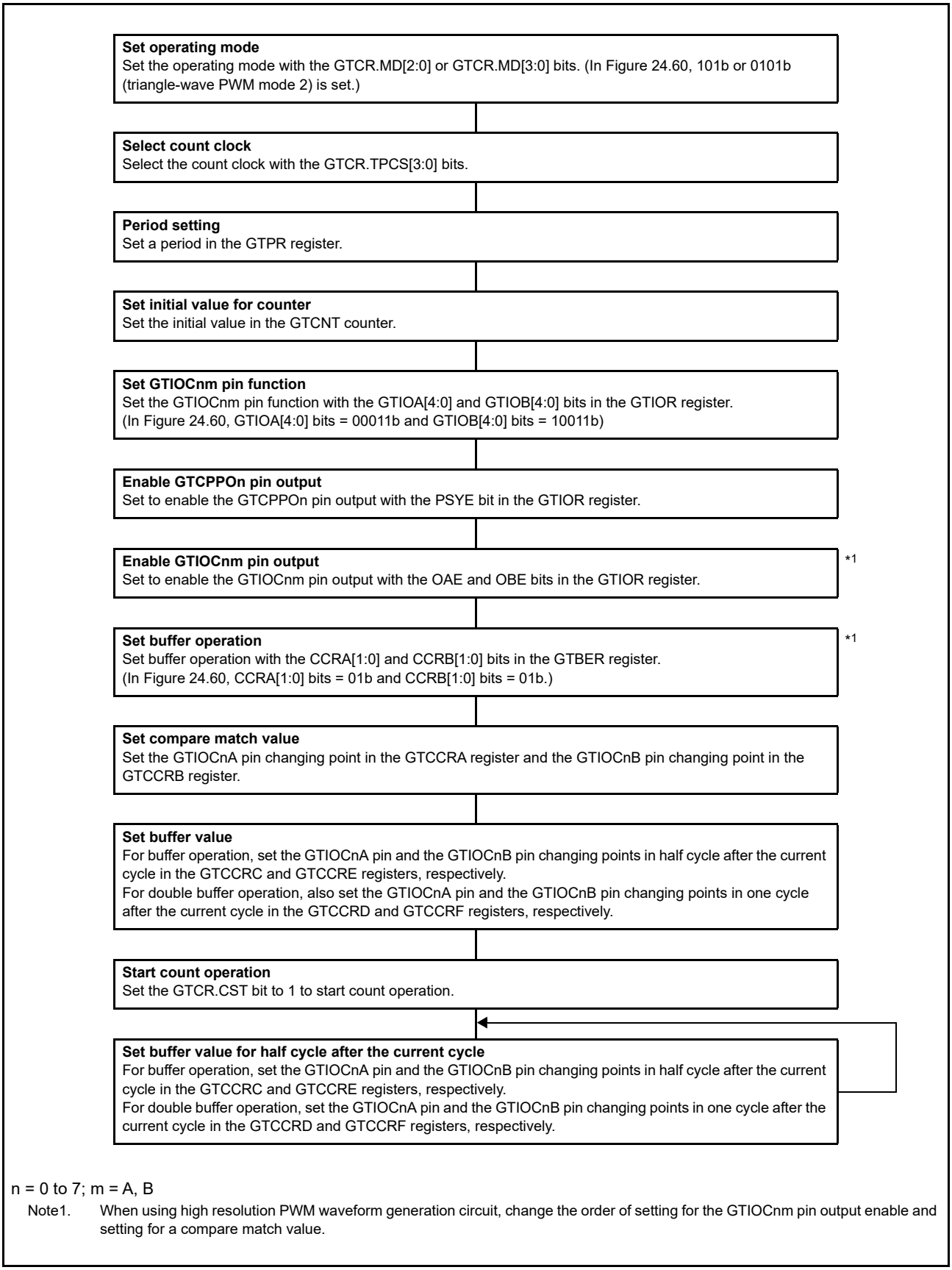


Figure 24.60 Example of Triangle-Wave PWM Mode 2 Operation (Buffer Operation, Low Output from the GTIOCnA Pin and High Output from the GTIOCnB Pin at Initial Output, Output Toggled at GTCCRA/GTCCRB Register Compare Match, Output Retained at the End of the Cycle, GTIOR.PSYE = 1) (n = 0 to 7)



n = 0 to 7; m = A, B

Note1. When using high resolution PWM waveform generation circuit, change the order of setting for the GTIOCnm pin output enable and setting for a compare match value.

Figure 24.61 Example for Setting Triangle-Wave PWM Mode 2

(6) Triangle-Wave PWM Mode 3 (64-Bit Transfer at Trough)

The triangle-wave PWM mode 3 is a mode in which the period is set in the GTPR register, the GTCNT counter performs triangle-wave (full-wave) operation and a PWM waveform is output to the GTIOCnA or GTIOCnB pin at a compare match of the GTCCRA or GTCCRB register with buffer operation fixed (n = 0 to 7). Buffer operation in triangle-wave PWM mode 3 is different from the usual buffer operation. Buffer transfer is performed from the GTCCRC register to the GTCCRA register, from the GTCCRE register to the GTCCRB register, from the GTCCRD register to the temporary register A, and from the GTCCRF register to the temporary register B at the trough, and from the temporary register A to the GTCCRA register and from the temporary register B to the GTCCRB register at the crest. The pin output value can be selected from low output, high output, or toggle output separately for a compare match and for the end of the cycle according to the GTIOR register setting.

By setting the GTDTCR, GTDVU, and GTDVD registers, a compare match value for a negative-phase waveform with dead time can automatically be set to the GTCCRB register.

When the GTBER.DBRTECm (m = A, B) bit is set to 1, transfer from an intermediate buffer to the GTCCRm (m = A, B) register is repeated on a cyclic basis with using the temporary register x (x = C, E) and temporary register m (m = A, B) which operate as intermediate buffers for the GTCCRx (x = C, E) and GTCCRm (m = A, B) registers, respectively, even while buffer transfer is disabled (repeated double buffer operation function during disabling of buffer transfer). For details, refer to section 24.8.2.2, Repeated Double-Buffered Operation When Disabling GTCCR Buffer Transfer.

Figure 24.62 shows an example of triangle-wave PWM mode 3 operation, and Figure 24.63 shows an example for setting triangle-wave PWM mode 3.

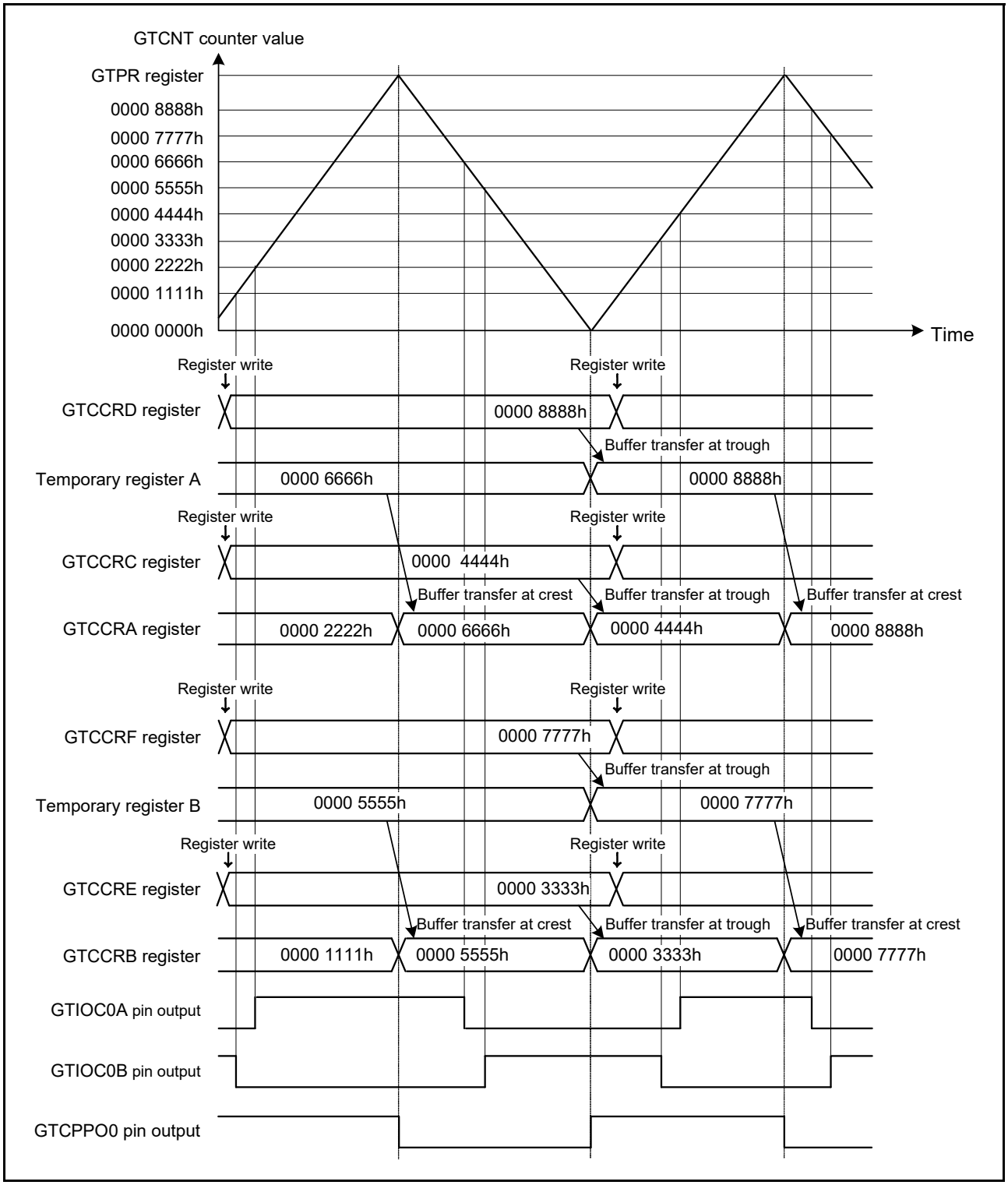
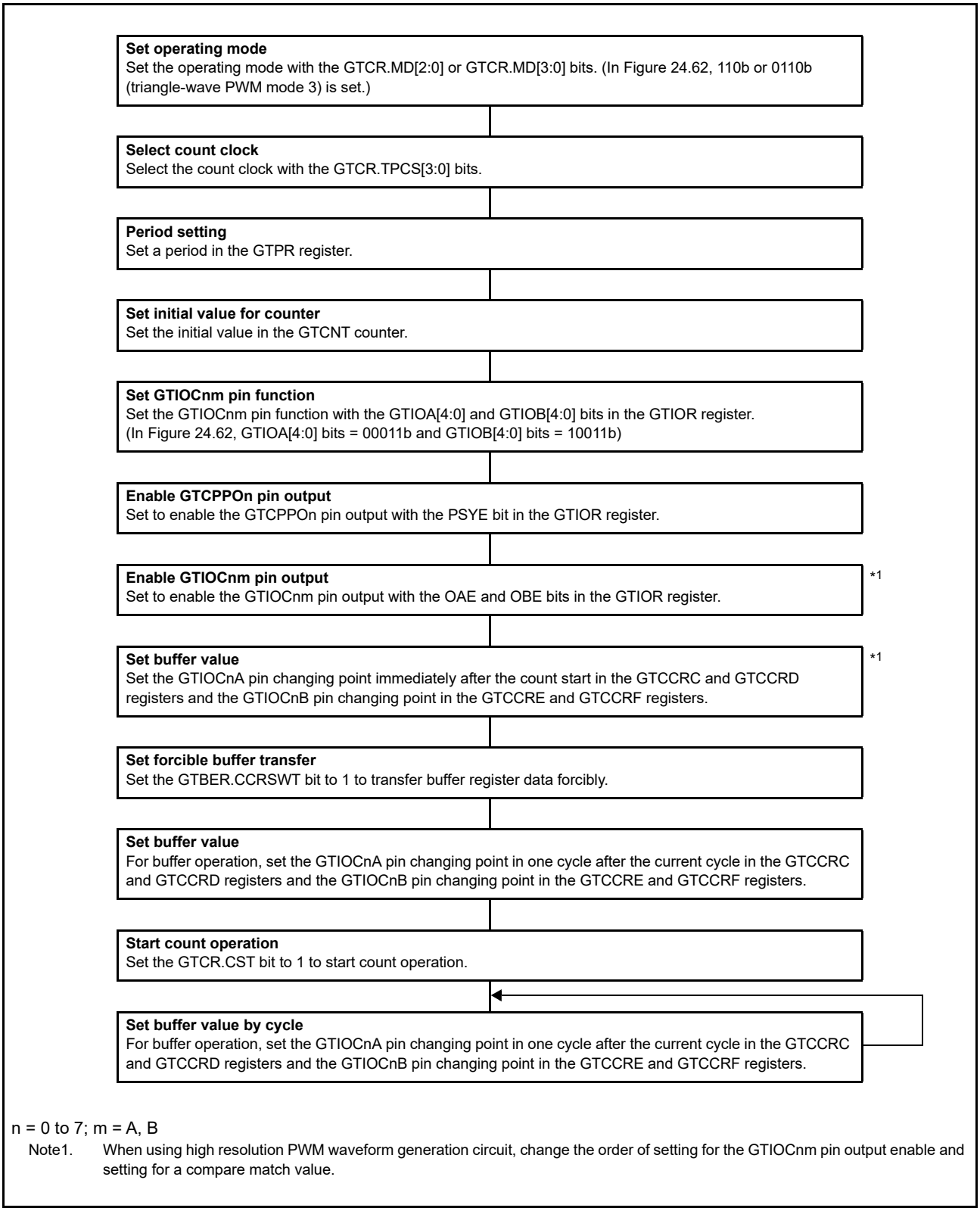


Figure 24.62 Example of Triangle-Wave PWM Mode 3 Operation (Low Output from the GTIOCnA Pin and High Output from the GTIOCnB Pin at Initial Output, Output Toggled at GTCCRA/GTCCRB Register Compare Match, Output Retained at the End of the Cycle, GTIOR.PSYE = 1) (n = 0 to 7)



n = 0 to 7; m = A, B

Note1. When using high resolution PWM waveform generation circuit, change the order of setting for the GTIOCnm pin output enable and setting for a compare match value.

Figure 24.63 Example for Setting Triangle-Wave PWM Mode 3

(7) Complementary PWM mode 1, 2, 3

In complementary PWM mode, a three-phase PWM waveform with dead time that ensures the linearity in the vicinity of duty 0% and 100% can be output using the GTCNT counter of consecutive three channels. There are four modes depending on differences in buffer operation: (1) complementary PWM mode 1 (transfer at crests), (2) complementary PWM mode 2 (transfer at troughs), (3) complementary PWM mode 3 (transfer at crests and troughs), and (4) complementary PWM mode 4 (immediate transfer).

Figure 24.64 shows the block diagram in complementary PWM modes 1 to 3.

Among consecutive three channels, the lowest channel is referred to as master channel, and the adjacent upper two channels are referred to as slave channel 1 (lower) and slave channel 2 (upper).

The GTCNT counter of each channel performs individual count operation under the cycle operation by the master channel.

In each channel, compare match with the GTCCRA register is performed selecting one of the three GTCNT counters in each operation section, and a positive-phase waveform and a negative-phase waveform are output from the GTIOCn+iA pin (n = 0, 4; i = 0, 1, 2) and the GTIOCn+iB pin respectively with a non-overlapping section of the dead time value set in the GTDVU register of the master channel.

The GTCCRA register performs buffer operation by the GTCCRC register, temporary register A, and GTCCRD register. In complementary PWM mode 3, setting the GTBER2.CP3DB bit to 1 also enables buffer operation of the GTCCRA register by the GTCCRE register, temporary register B, and GTCCRF register, allowing double buffer operation.

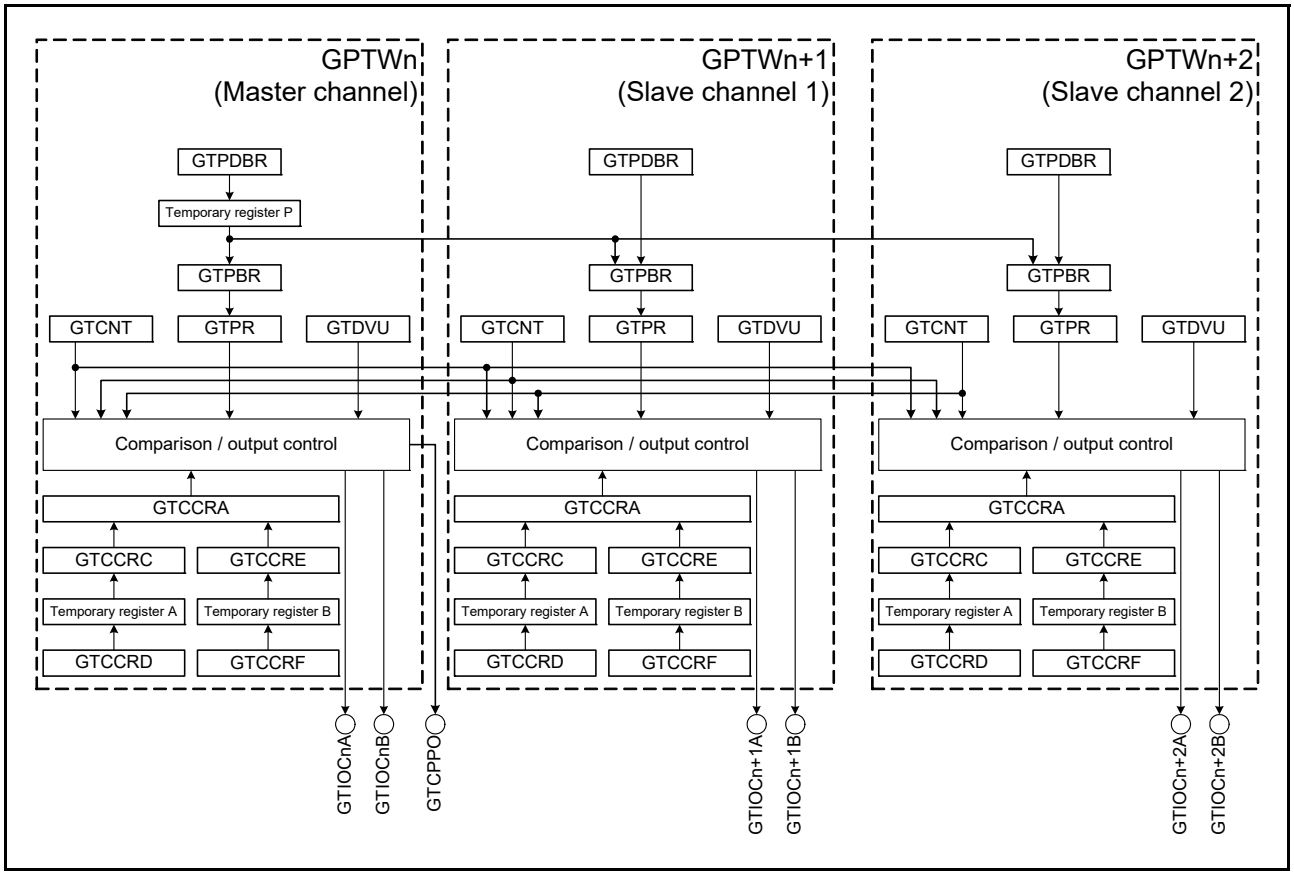


Figure 24.64 Block Diagram in Complementary PWM Mode 1, 2, 3 (n = 0, 4)

The GPTWn.GTCNT counter performs count operation for triangle waves using the GPTWn.GTPR register as a cycle register. A section where the GPTWn.GTCNT counter value is not larger than the dead time value is referred to as trough section.

The GPTWn+1.GTCNT counter performs count operation with the value (GPTWn.GTCNT counter value + dead time value set in the GPTWn.GTDVU register). A section where the GPTWn+1.GTCNT counter value is larger than the

GPTWn.GTPR register value is referred to as crest section.
Crest section and trough section are classified into up-counting crest section, down-counting crest section, up-counting trough section, and down-counting trough section according to counting direction. A section between trough section and crest section is referred to as up-counting middle section or down-counting middle section according to counting direction.
A section equivalent to the up-counting trough section after starting count operation is referred to as initial output section where operation differs partially from other up-counting trough sections.
The GPTWn+2.GTCNT counter functions as a counter to ensure the linearity in the vicinity of duty 0% and 100%. In a crest section, this counter performs count operation for a triangle wave (up-counting after down-counting) with the value (GPTWn.GTPR register value + dead time value) as an initial value and the GPTWn.GTPR register value as a trough. This counter is cleared to 0 at the end of the crest section, and then stops counting until the next trough section. In a trough section, this counter performs count operation for a triangle wave with an initial value of 0 and dead time value as a crest.
This counter becomes the value (GPTWn.GTPR register value + dead time value) at the end of the trough section, and then stops counting until the next crest section. In the initial output section, however, this counter counts up with an initial value of 0 until the dead time value, and then becomes the value (GPTWn.GTPR register value + dead time value).
If the counter stops and then restarts in complementary PWM mode, the counter of each channel returns to the initial value after starting count operation, and then starts counting from the initial output section.
Table 24.12 and Table 24.13 show count operation (counting direction/counting range) in each section. In these tables, registers with no channel identification indicate that the same value is stored in them of the master channel, slave channel 1, and slave channel 2.

Table 24.12 Count Operation in Complementary PWM Mode (1)

Counter	Initial Value	Initial Output Section (After Start)	Up-Counting Middle Section	Up-Counting Crest Section	Down-Counting Crest Section
GPTWn.GTCNT	0	Up-Counting 0 to GTDVU	Up-Counting GTDVU + 1 to GTPR – GTDVU	Up-Counting GTPR – GTDVU + 1 to GTPR	Down-Counting GTPR – 1 to GTPR – GTDVU
GPTWn+1.GTCNT	GTDVU	Up-Counting GTDVU to GTDVU × 2	Up-Counting GTDVU × 2 – 1 to GTPR	Up-Counting GTPR + 1 to GTPR + GTDVU	Down-Counting GTPR + GTDVU – 1 to GTPR
GPTWn+2.GTCNT	0	Up-Counting 0 to GTDVU	Stop GTPR + GTDVU	Down-Counting GTPR + GTDVU – 1 to GTPR	Up-Counting GTPR + 1 to GTPR + GPDVU

Table 24.13 Count Operation in Complementary PWM Mode (2)

Counter	Down-Counting Middle Section	Down-Counting Trough Section	Up-Counting Trough Section
GPTWn.GTCNT	Down-Counting GTPR – GTDVU – 1 to GTDVU	Down-Counting GTDVU – 1 to 0	Up-Counting 1 to GTDVU
GPTWn+1.GTCNT	Down-Counting GTPR – 1 to GTDVU × 2	Down-Counting GTDVU × 2 – 1 to GTDVU	Down-Counting GTDVU + 1 to GTDVU × 2
GPTWn+2.GTCNT	Stop 0	Up-Counting 1 to GTDVU	Down-Counting GTDVU – 1 to 0

In complementary PWM mode, the GTCCRA register buffer operation is different from normal buffer operation.
Data transfers from the GTCCRD register to the temporary register A and from the GTCCRF register to the temporary register B are performed at the same time in three channels by writing a value to the GTCCRD register of the GPTWn+2 channel.
Data transfers from the temporary register A and temporary register B to the GTCCRC and GTCCRE registers vary depending on the transfer timing to the temporary register A and temporary register B. Data transfers from the GTCCRC and GTCCRE registers to the GTCCRA register are performed according to each complementary PWM mode name (crest transfer, trough transfer, and crest/trough transfer).
Buffer operation of the GTPR register in complementary PWM mode is described in section 24.3.2.1, GTPR Register Buffer Operation. Do not perform buffer operation for the GTDVU register in complementary PWM mode.

Table 24.14 shows buffer transfer timing during single buffer operation in complementary PWM modes 1 to 3. Table 24.15 shows buffer transfer timing during double buffer operation in complementary PWM mode 3.

Table 24.14 Single Buffer Transfer Timing in Complementary PWM Mode 1, 2, 3

Buffer Transfer	Complementary PWM Mode 1	Complementary PWM Mode 2	Complementary PWM Mode 3
GTCCRD ↓ Temporary register A	After one PCLKC cycle from the GTCCRD register write of slave channel 2 (GPTWn+2)	After one PCLKC cycle from the GTCCRD register write of slave channel 2 (GPTWn+2)	After one PCLKC cycle from the GTCCRD register write of slave channel 2 (GPTWn+2)
Temporary register A ↓ GTCCRC	(1) When transferring to the temporary register A at the middle section of the up-count • After 1 PCLKC after for transferred to the temporary register A (2) When transferring to the temporary register A at other than the middle section of the up-count • The end of the trough	(1) When transferring to the temporary register A at the middle section of the down-count • After 1 PCLKC after for transferred to the temporary register A (2) When transferring to the temporary register A at other than the middle section of the down-count • The end of the crest	(1) When transferring to the temporary register A at the middle section • After 1 PCLKC after for transferred to the temporary register A (2) When transferring to the temporary register A at other than the middle section • The end of the crest/trough
GTCCRC ↓ GTCCRA	• The end of the trough • Counter clear in the up-count intermediate section and crest section	• The end of the trough section excluding the initial output section • Counter clear in the down-count intermediate section and trough section	• The end of the trough • The end of the trough section excluding the initial output section • Counter clear

Table 24.15 Double Buffer Transfer Timing in Complementary PWM Mode 3

Buffer Transfer	Transfer Timing
GTCCRD ↓ Temporary register A	After 1 PCLKC after slave channel 2 (GPTWn+2) writes to the GTCCRD register
Temporary register A ↓ GTCCRC	(1) When transferring data to the temporary register A at the middle section • After 1 PCLKC after data transfer to the temporary register A (2) When transferring data to the temporary register A at other than the middle section • At the end of crest and trough sections
GTCCRC ↓ GTCCRA	• The end of the crest • Counter clear

Buffer Transfer	Transfer Timing
GTCCRF ↓ Temporary register B	After 1 PCLKC after slave channel 2 (GPTWn+2) writes to the GTCCRD register
Temporary register B ↓ GTCCRE	(1) When transferring data to the temporary register B at the middle section • After 1 PCLKC after data transfer to the temporary register A (2) When transferring data to the temporary register B at other than the middle section • At the end of crest and trough sections
GTCCRE ↓ GTCCRA	• The end of the trough section excluding the initial output section

The change in the output level of the positive phase waveform output from the GTIOCn+mA pin (m = 0, 1, 2) and the negative phase waveform output from the GTIOCn+mB pin (m = 0, 1, 2) is determined by the compare match by the combination of the counter and register in each operation section. In the middle section, the compare match between the GPTWn.GTCNT counter and the GTCCRA register changes the positive phase waveform output level, and the compare match between the GPTWn+1.GTCNT counter and the GTCCRA register changes the negative phase waveform output level. The crest and trough sections use the GPTWn+2.GTCNT counter, GTCCRC register, and GTCCRE register to perform a compare match operation to ensure linearity near 0% and 100% duty.

When the comparison match value is greater than or equal to the GPTWn.GTPR register value, the duty is 0% (positive phase waveform OFF, negative phase waveform ON). When the comparison match value is 0, the duty is 100% (positive phase waveform ON, negative phase waveform OFF).

Table 24.16 shows the counter and register combinations used in the comparative match operation to generate a

positive-phase waveform and a negative-phase waveform in each operating section.

Table 24.16 Combinations of Counters and Registers for Compare Match Operation in Complementary PWM Mode

	Up-Counting Middle Section	Up-Counting Crest Section	Up-Counting Crest Section	Up-Counting Middle Section	Up-Counting Trough Section	Up-Counting Trough Section
Negative-Phase OFF	GPTWn+1.GTCNT	GPTWn+1.GTCNT	—	GPTWn+2.GTCNT*1	GPTWn+2.GTCNT	GPTWn+1.GTCNT
	GTCCRA	GTCCRA	—	GTCCRC (GTCCRE for double buffer operation)	GTCCRC (GTCCRE for double buffer operation)	GTCCRC (GTCCRE for double buffer operation)
Positive-Phase ON	GPTWn.GTCNT	GPTWn.GTCNT	GPTWn+2.GTCNT	—	GPTWn+2.GTCNT*1	GPTWn+2.GTCNT
	GTCCRA	GTCCRA	GTCCRA	—	GTCCRC (GTCCRE for double buffer operation)	GTCCRC (GTCCRE for double buffer operation)
Negative-Phase OFF	GPTWn+2.GTCNT*1	GPTWn+2.GTCNT	GPTWn.GTCNT	GPTWn.GTCNT	GPTWn.GTCNT	—
	GTCCRC	GTCCRC	GTCCRC	GTCCRA	GTCCRA	—
Positive-Phase ON	—	GPTWn+1.GTCNT*1	GPTWn+1.GTCNT	GPTWn+1.GTCNT	GPTWn+1.GTCNT	GPTWn+2.GTCNT
	—	GTCCRC	GTCCRC	GTCCRA	GTCCRA	GTCCRA

Note 1. Compare match is performed only at the time of final count in the target section, but is not performed at count values other than the final count.

In the case of normal complementary PWM mode waveform, a PWM waveform change occurs in the order of negative-phase OFF → positive-phase ON → positive-phase OFF → negative-phase ON. However, this order may vary depending on operation section and register values. In this case, OFF takes precedence in trough sections and ON takes precedence in crest sections (for negative-phase waveforms), and ON takes precedence in trough sections and OFF takes precedence in crest sections (for positive-phase waveforms). A lower-priority compare match that occurs at the same time or after a higher-priority compare match is ignored.

In the initial output section, the initial output set in the GTIOR register is retained. In the case that the GTCCRA register value is larger than the GTDVU register value at the end of the initial output section, negative phase is enabled. In the case that the GTCCRA register value is not larger than the GTDVU register value, positive phase is enabled.

As operation examples of normal complementary PWM mode waveform where compare match operation occurs in the middle section, Figure 24.65 and Figure 24.66 show complementary PWM mode 1, Figure 24.67 and Figure 24.68 show complementary PWM mode 2, Figure 24.69 and Figure 24.70 show single buffer complementary PWM mode 3, and Figure 24.71 and Figure 24.72 show double buffer complementary PWM mode 3.

Figure 24.73 to Figure 24.84 show complementary PWM mode waveforms where compare match operation occurs in crest sections and trough sections and differences due to compare match occurrence order.

Figure 24.85 and Figure 24.86 show examples of initial output operation according to the GTCCRA register value.

Figure 24.87 shows an example for setting complementary PWM modes 1 to 3.

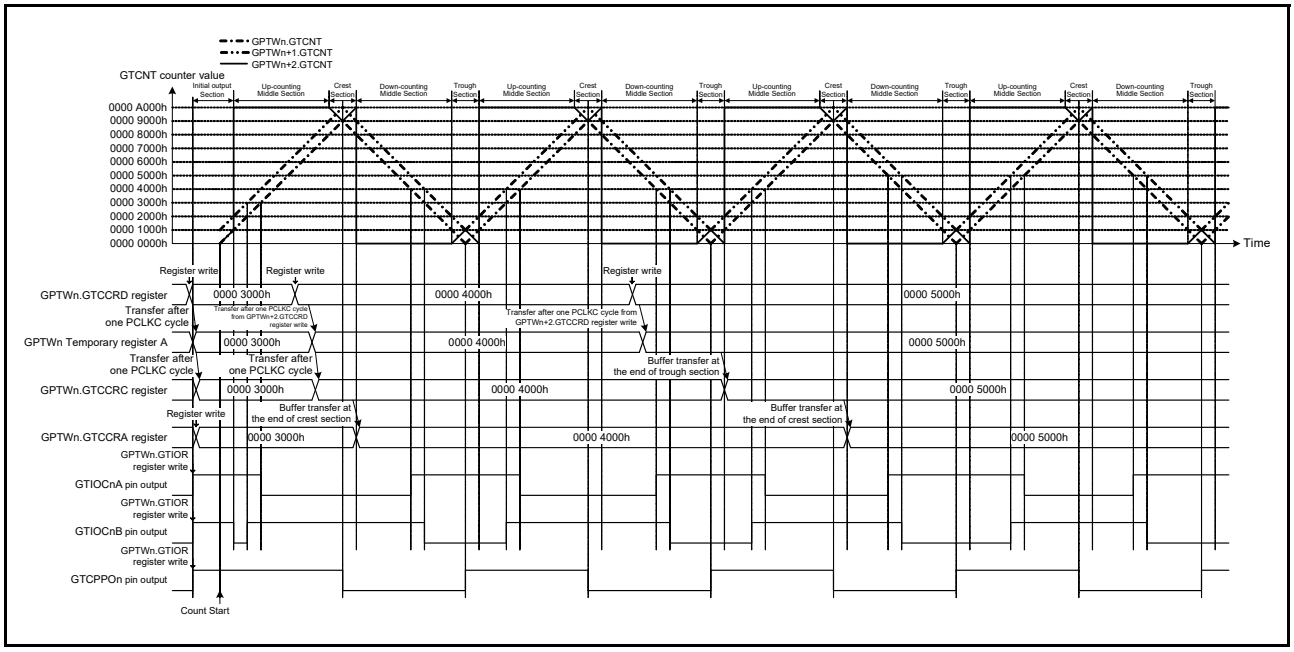


Figure 24.65 Example of Complementary PWM Mode 1 Operation (1)
(GTIOCnA pin = High/GTIOCnB pin = High as initial output,
GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting,
GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting,
the dead time value is 0000 1000h and updating GTCCRD register in middle section) (n = 0, 4)

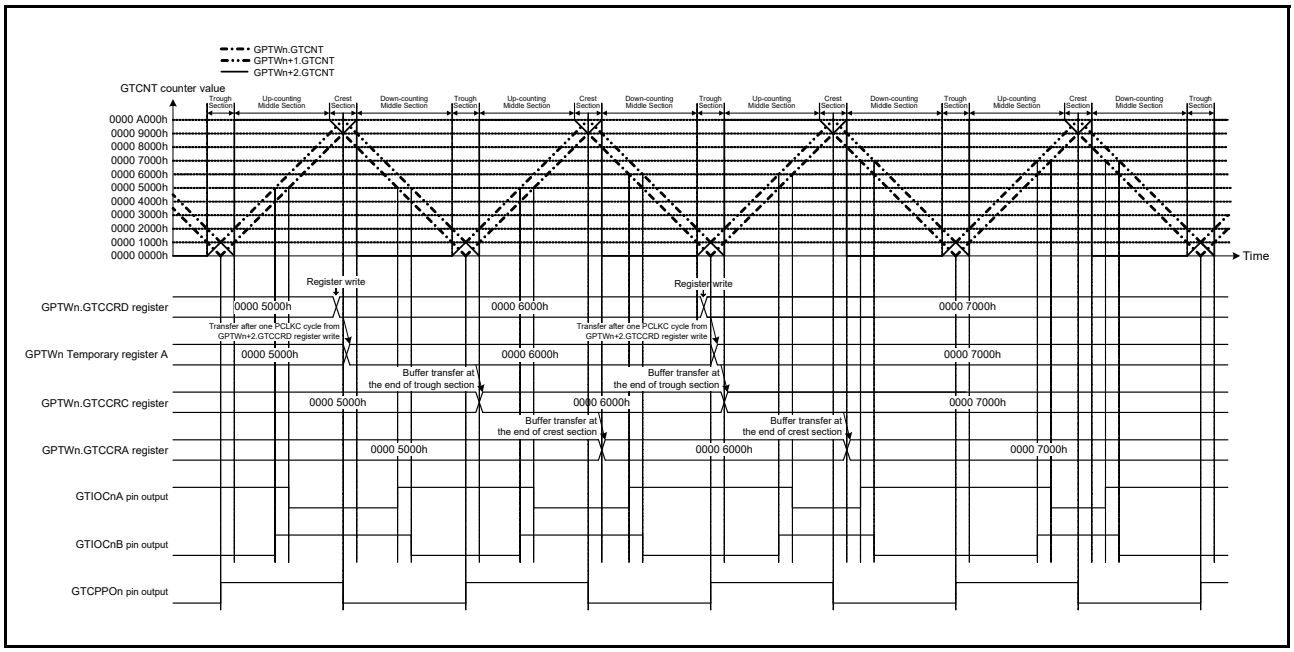


Figure 24.66 Example of Complementary PWM Mode 1 Operation (2)
(GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting,
GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting,
the dead time value is 0000 1000h and updating GTCCRD register in crest and trough sections) (n = 0, 4)

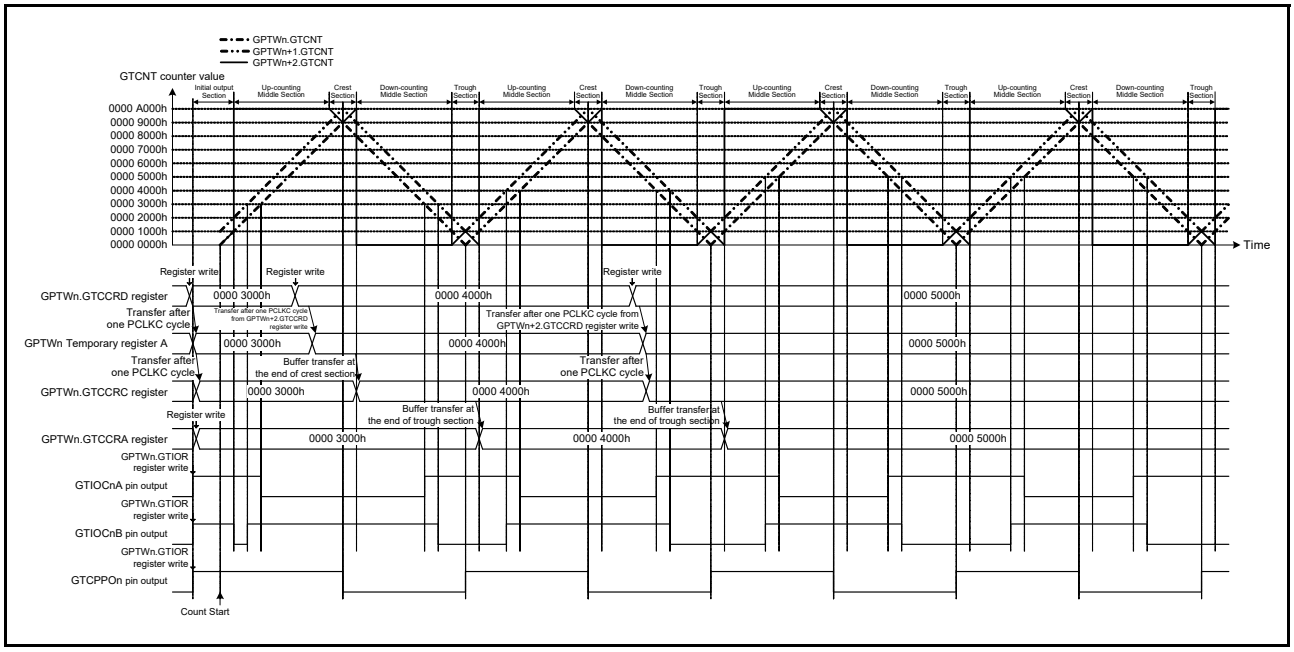


Figure 24.67 Example of Complementary PWM Mode 2 Operation (1)
(GTIOCnA pin = High/GTIOCnB pin = High as initial output,
GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting,
GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting,
the dead time value is 0000 1000h and updating GTCCRD register in middle section) (n = 0, 4)

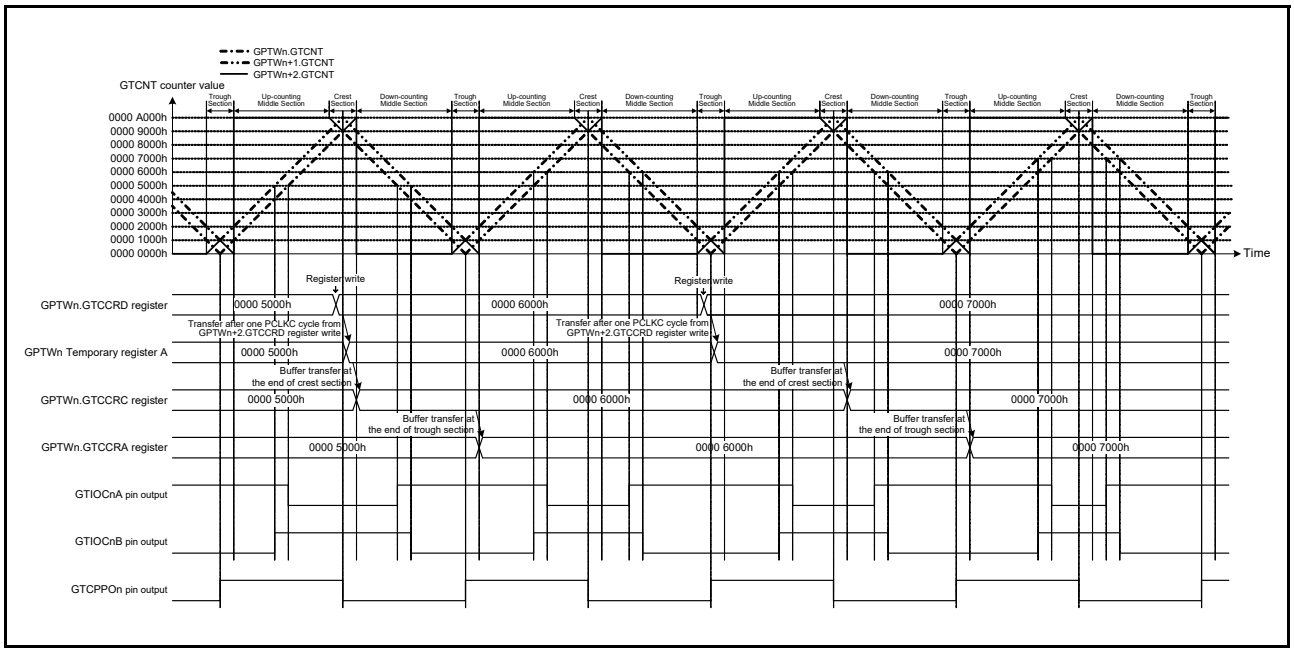


Figure 24.68 Example of Complementary PWM Mode 2 Operation (2)
(GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting,
GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting,
the dead time value is 0000 1000h and updating GTCCRD register in crest and trough sections) (n = 0, 4)

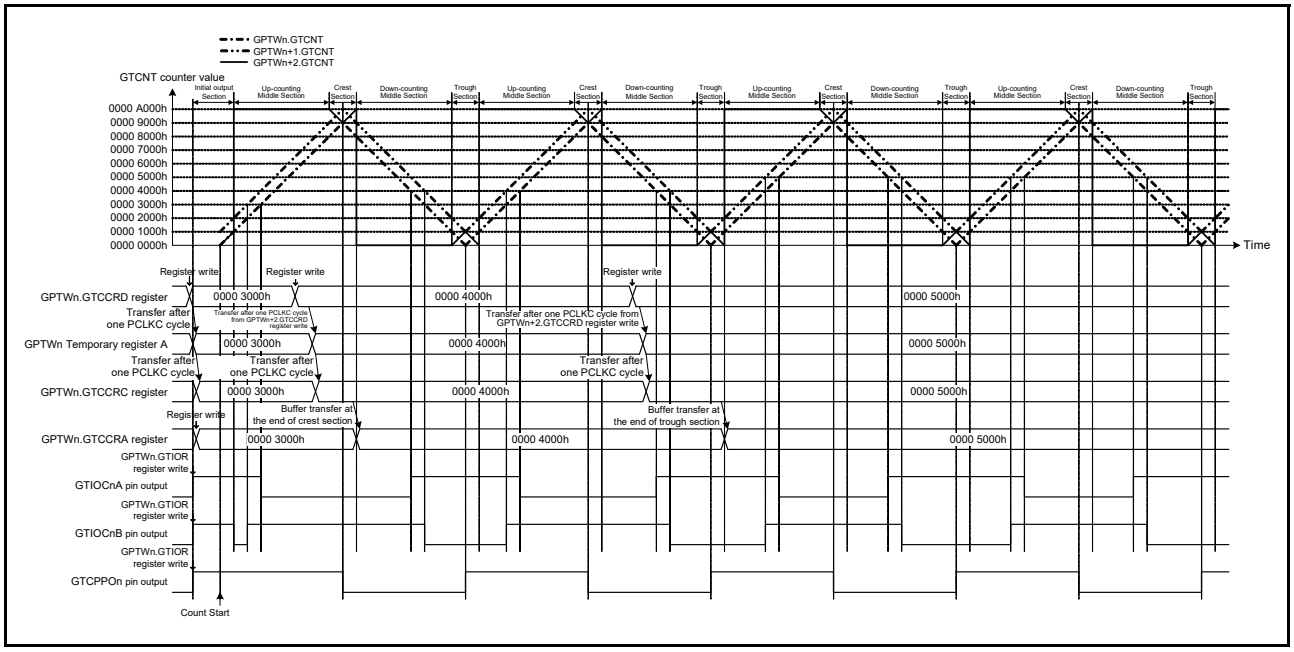


Figure 24.69 Example of Complementary PWM Mode 3 Operation (1)
(Single buffer operation, GTIOCnA pin = High/GTIOCnB pin = High as initial output, GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting,
the dead time value is 0000 1000h and updating GTCCRD register in middle section) (n = 0, 4)

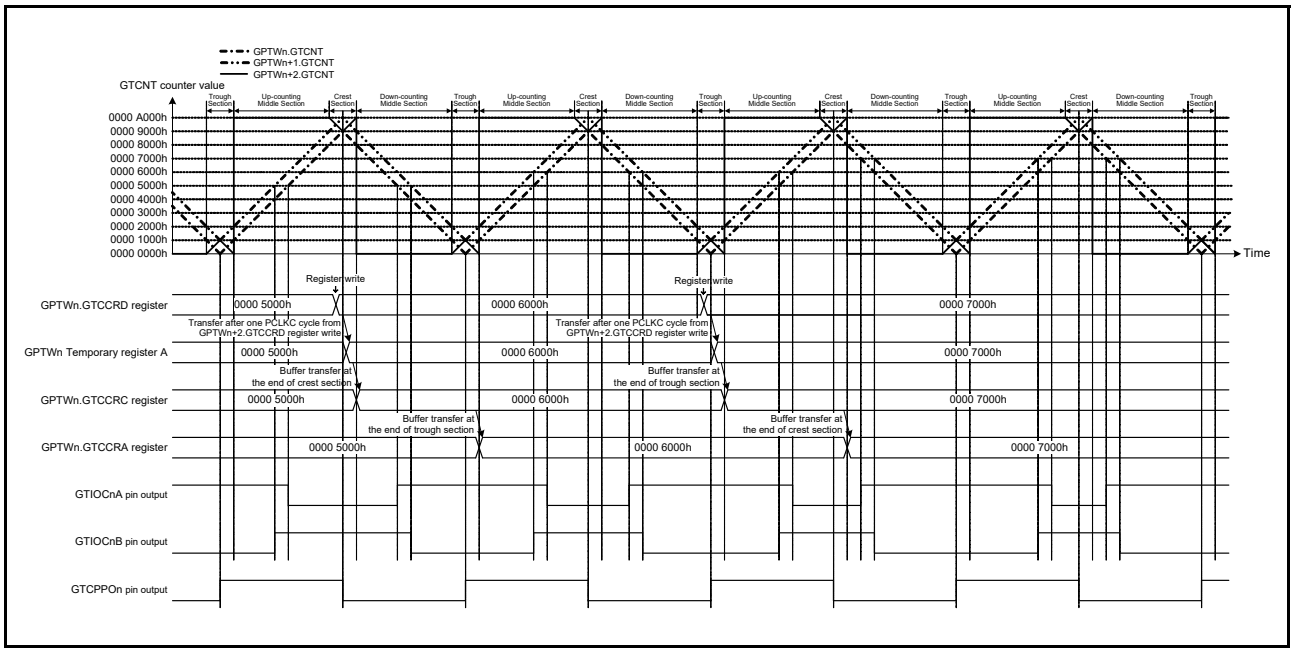


Figure 24.70 Example of Complementary PWM Mode 3 Operation (2)
(Single buffer operation,
GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting,
the dead time value is 0000 1000h and updating GTCCRD register in crest and trough sections) (n = 0, 4)

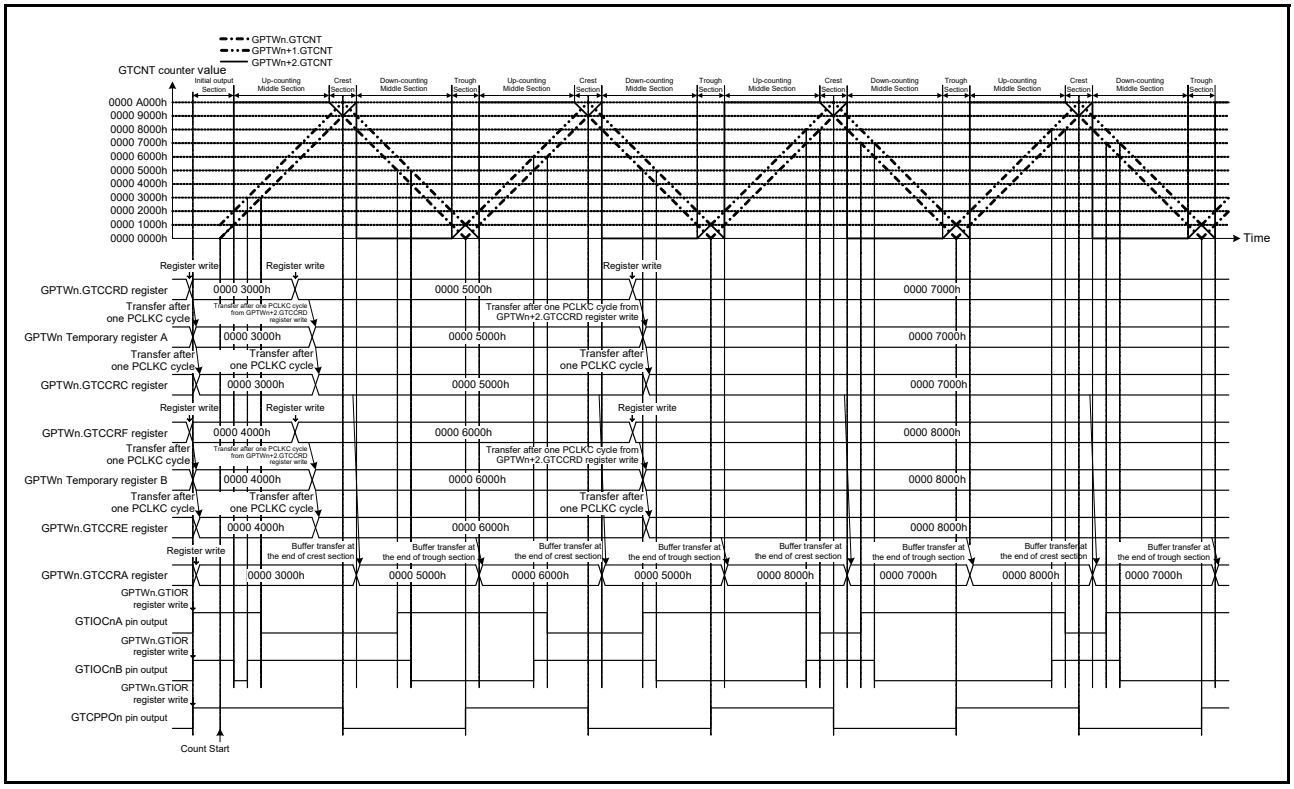


Figure 24.71 Example of Complementary PWM Mode 3 Operation (3)
(Double buffer operation,
GTIOCnA pin = High/GTIOCnB pin = High as initial output,
GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting,
GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting,
the dead time value is 0000 1000h and updating GTCCRD register in middle section) (n = 0, 4)

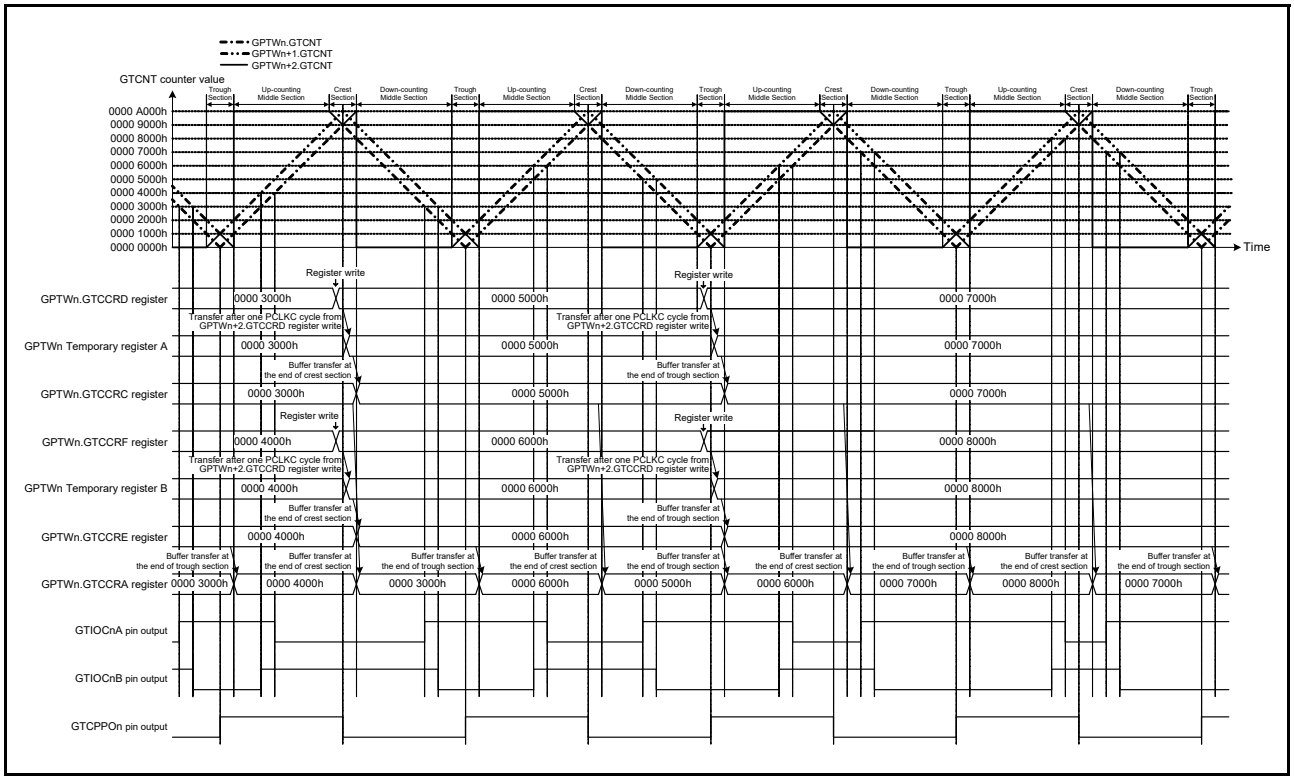


Figure 24.72 Example of Complementary PWM Mode 3 Operation (4)
(Double buffer operation,
GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting,
GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting,
the dead time value is 0000 1000h and updating GTCCRD register in crest and trough sections)
(n = 0, 4)

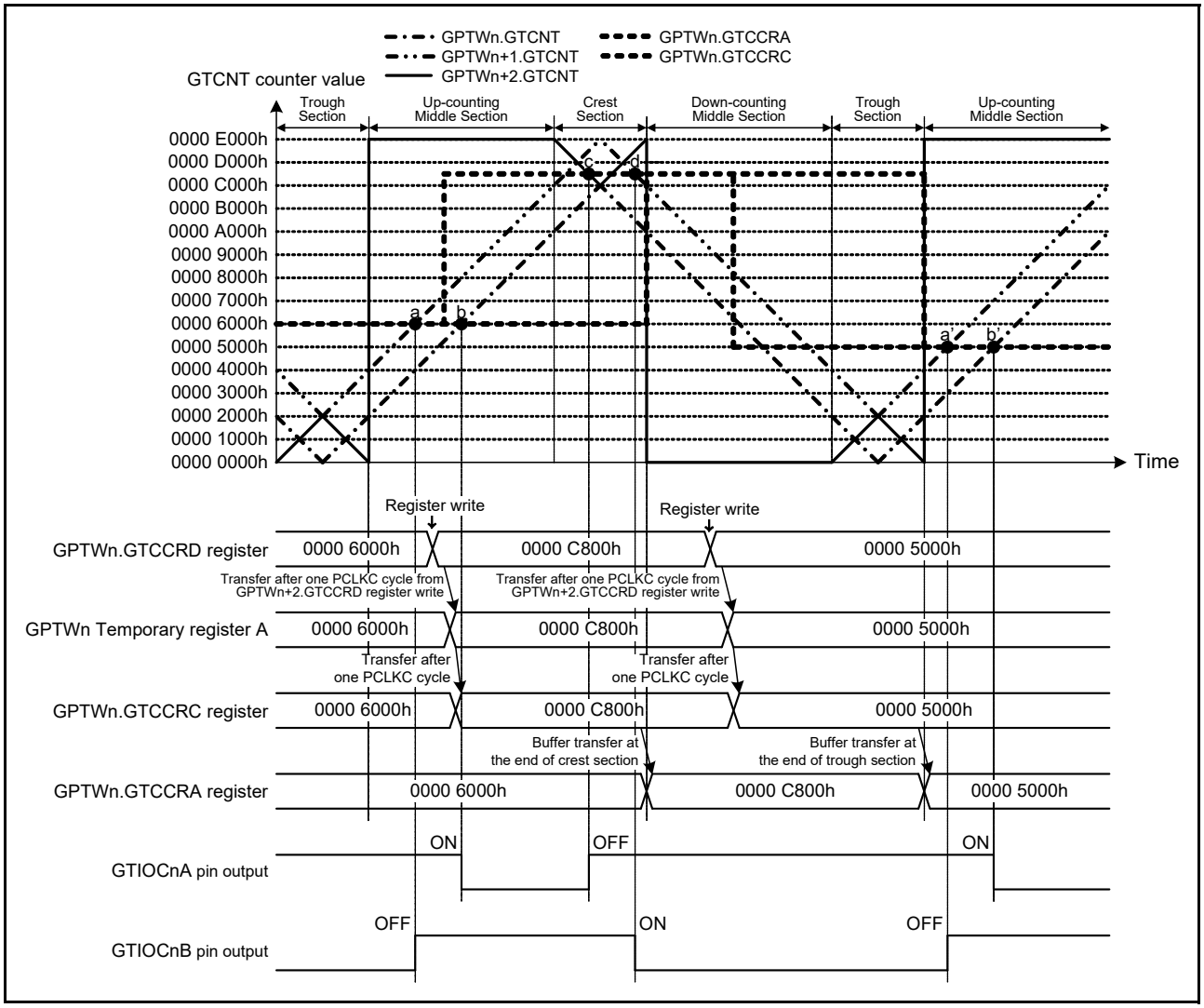


Figure 24.73 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (1) (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0000 2000h, compare match generation order: a → b → c → d) (n = 0, 4)

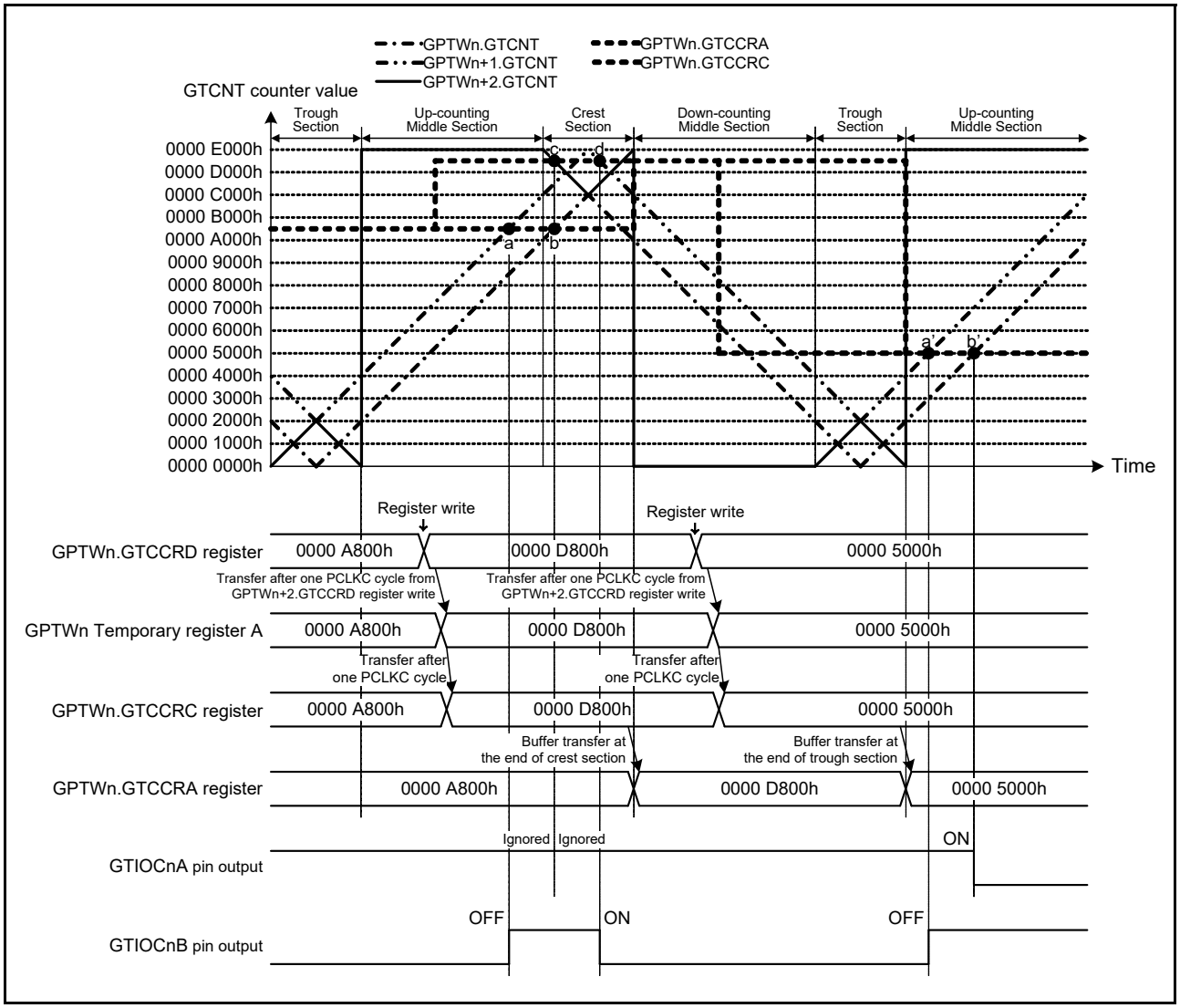


Figure 24.74 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (2) (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0000 2000h, compare match generation order: a → (b, c) → d) (n = 0, 4)

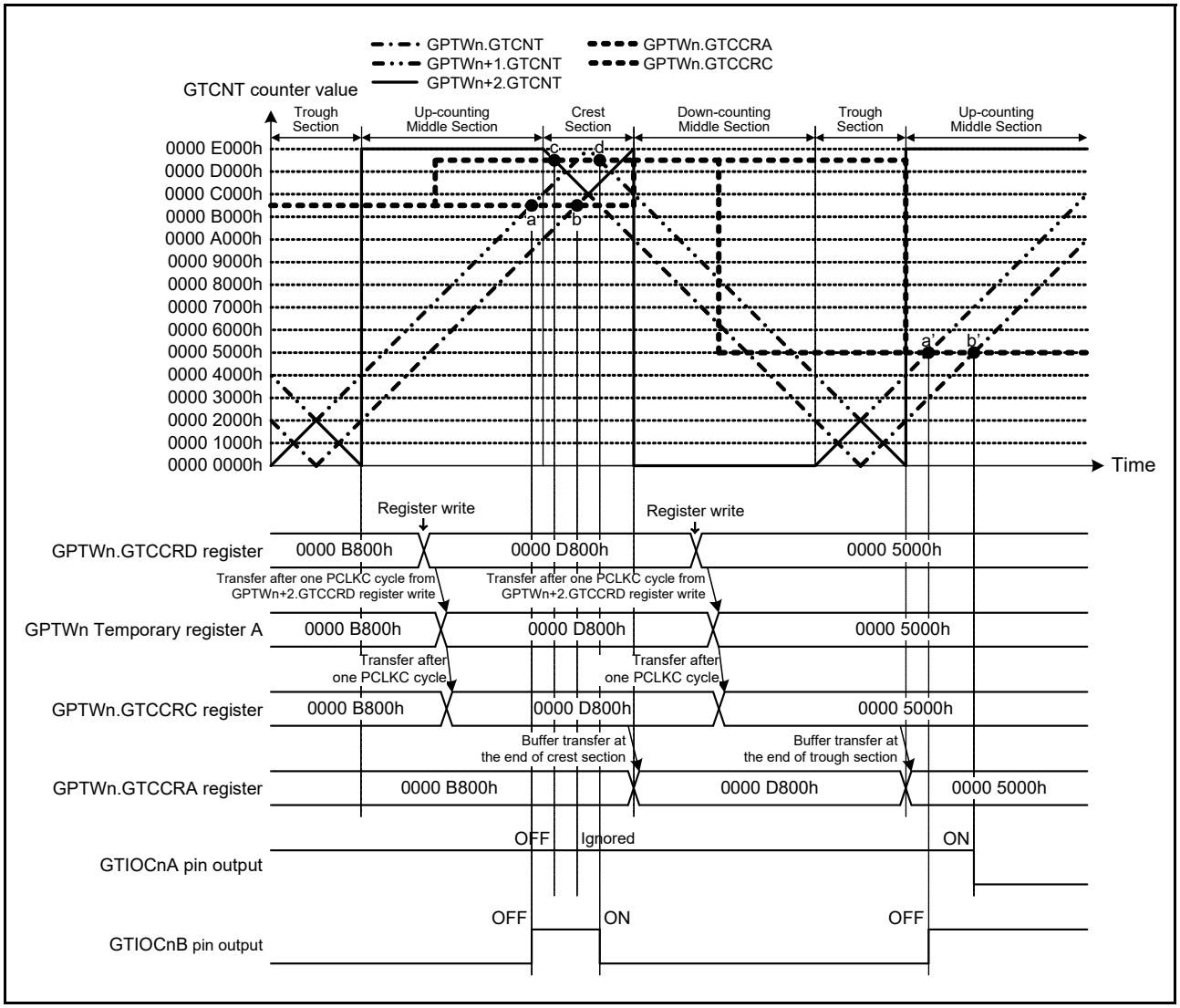


Figure 24.75 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (3) (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0000 2000h, compare match generation order: a → c → b → d) (n = 0, 4)

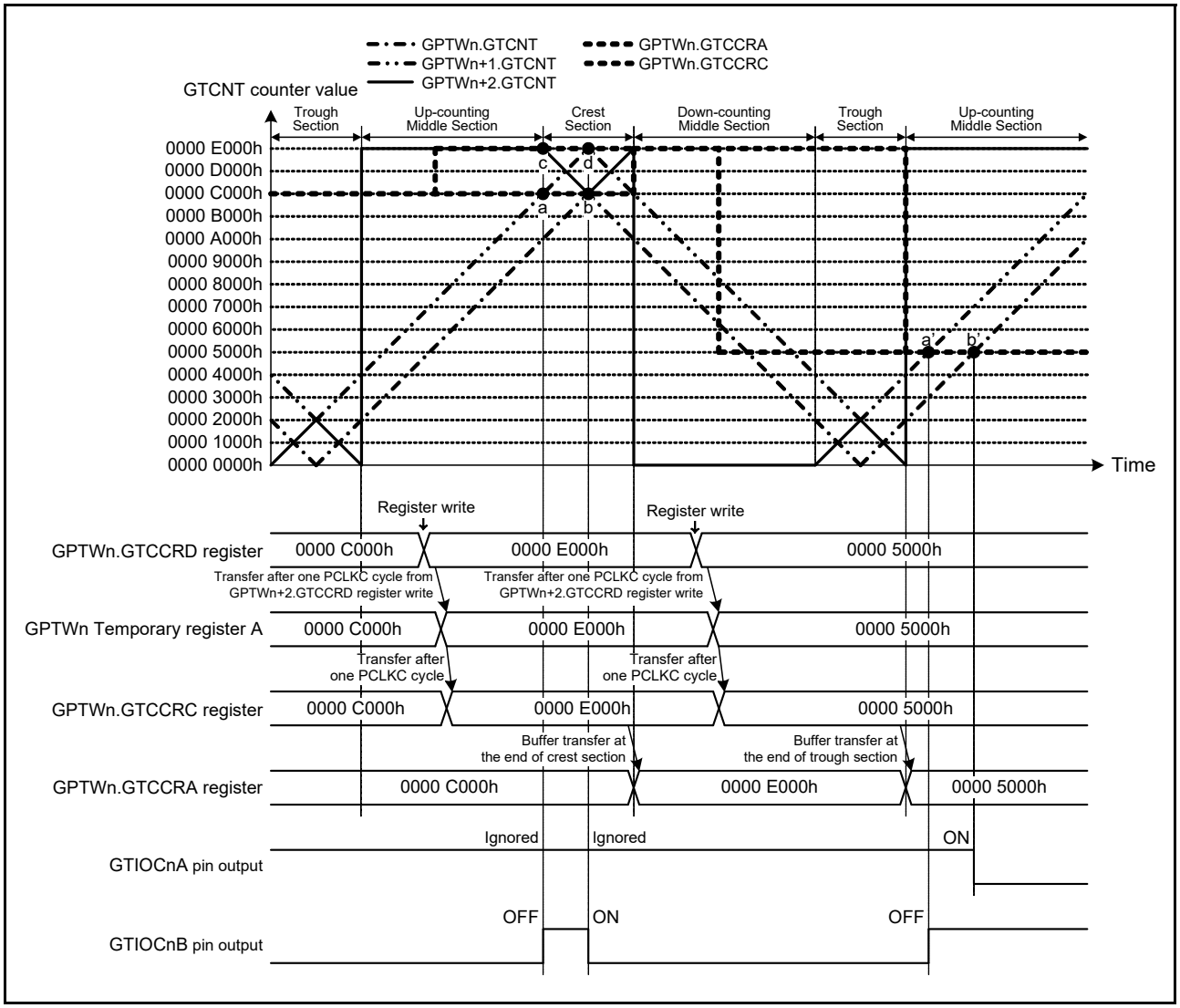


Figure 24.76 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (4) (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0000 2000h, compare match generation order: (a, c) → (b, d)) (n = 0, 4)

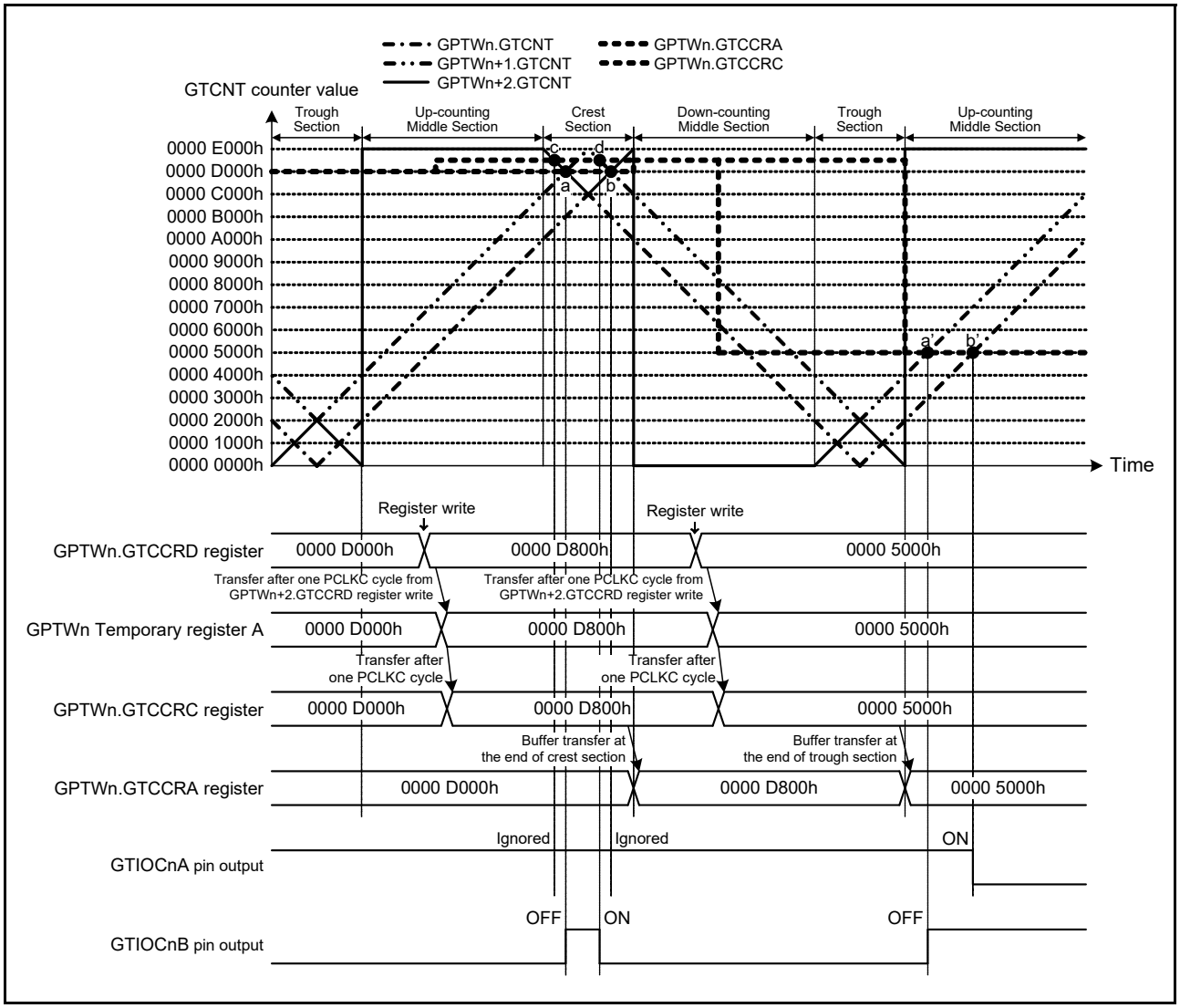


Figure 24.77 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (5) (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0000 2000h, compare match generation order: c → a → d → b) (n = 0, 4)

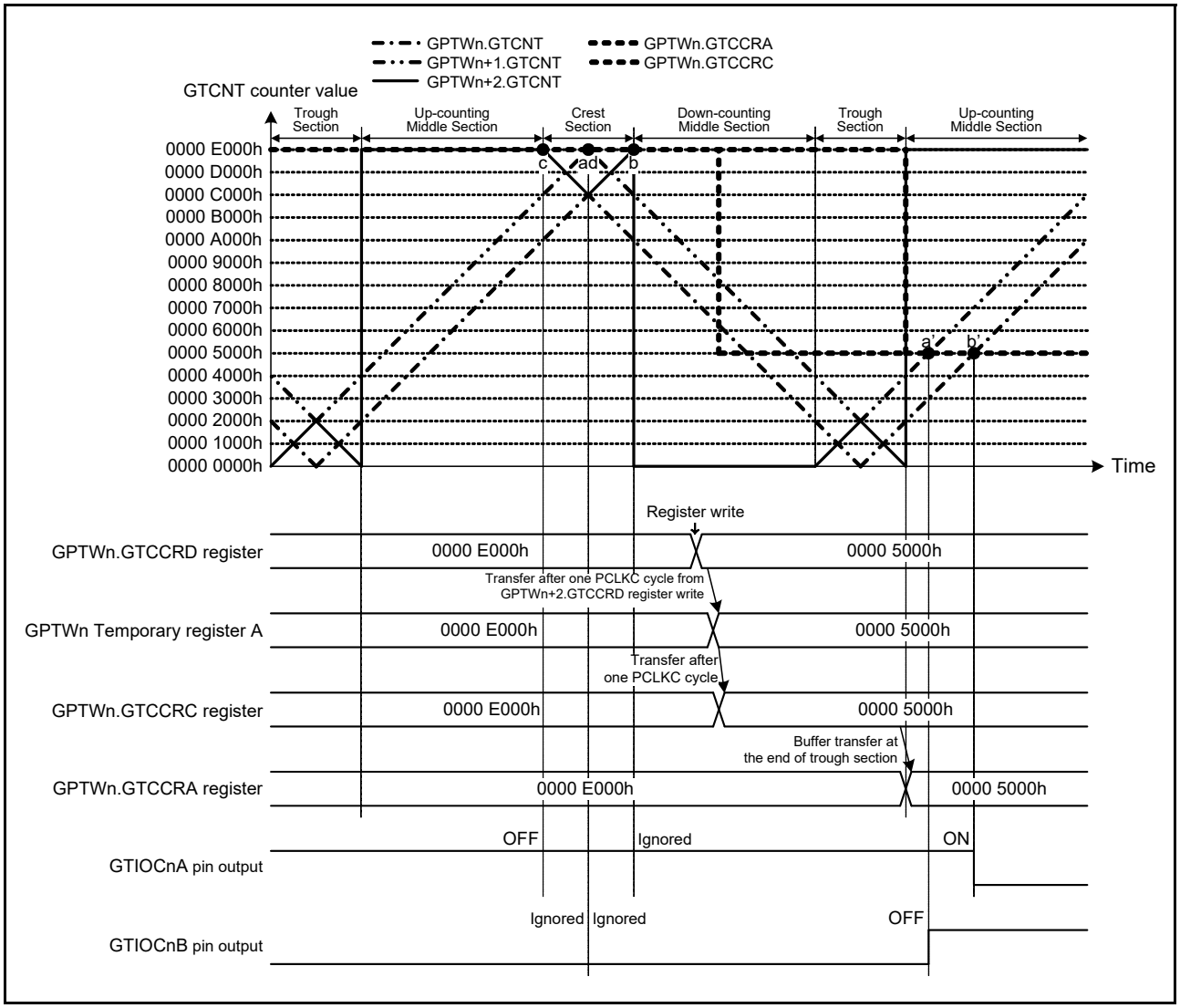


Figure 24.78 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (6) (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0000 2000h, compare match generation order: c → (a, d) → b) (n = 0, 4)

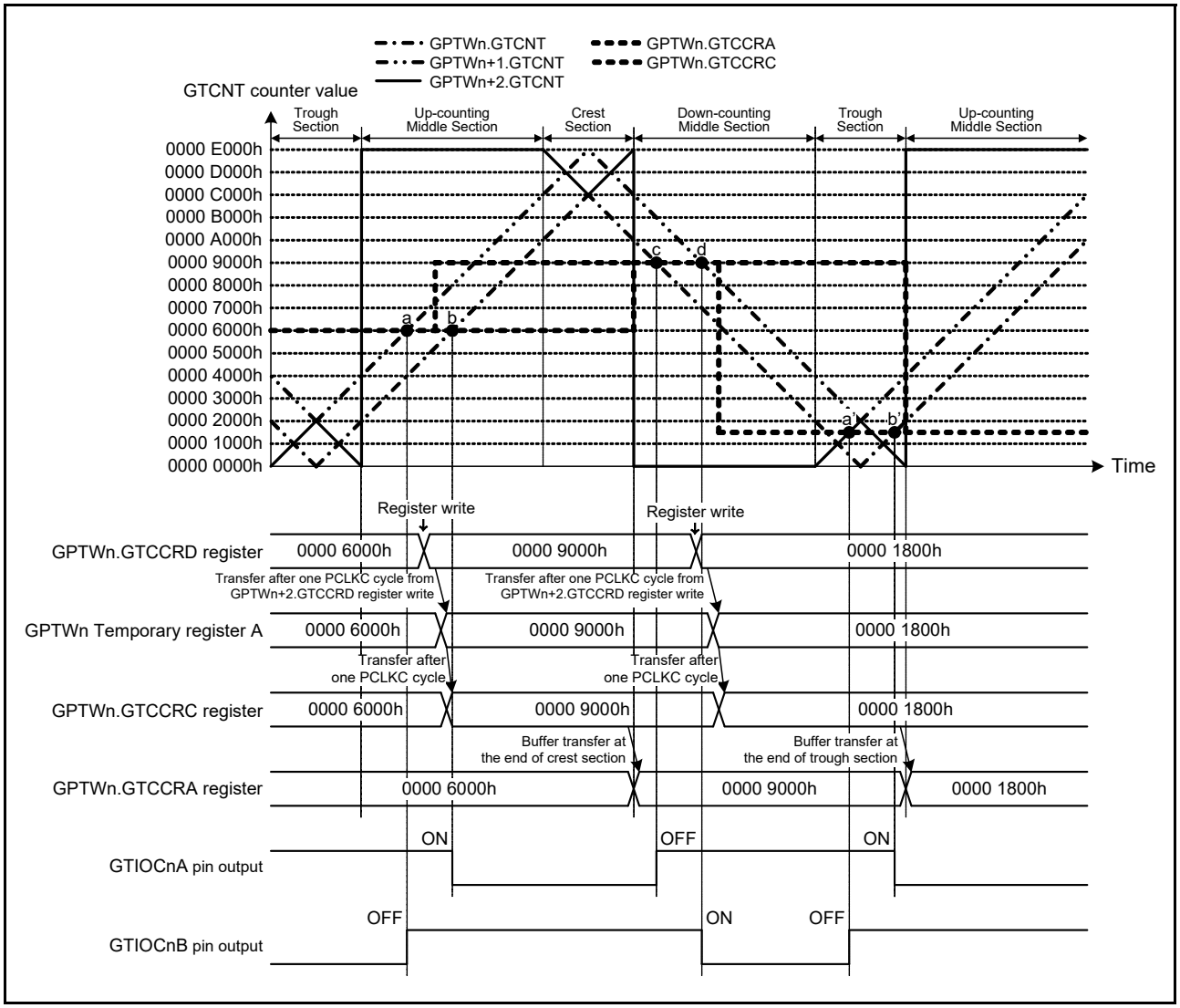


Figure 24.79 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (7) (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0000 2000h, compare match generation order: c → d → a' → b') (n = 0, 4)

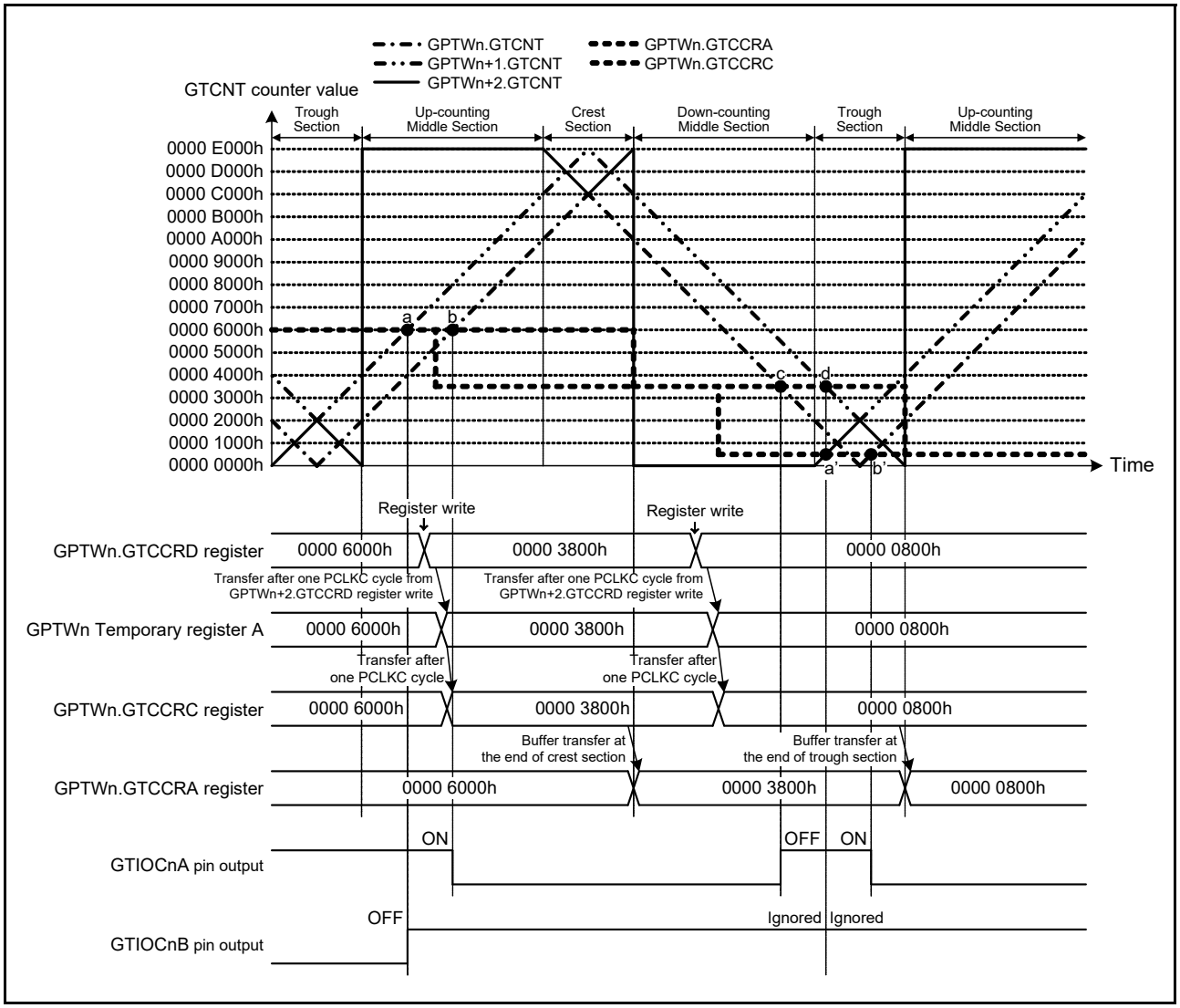


Figure 24.80 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (8) (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0000 2000h, compare match generation order: c → (d, a') → b') (n = 0, 4)

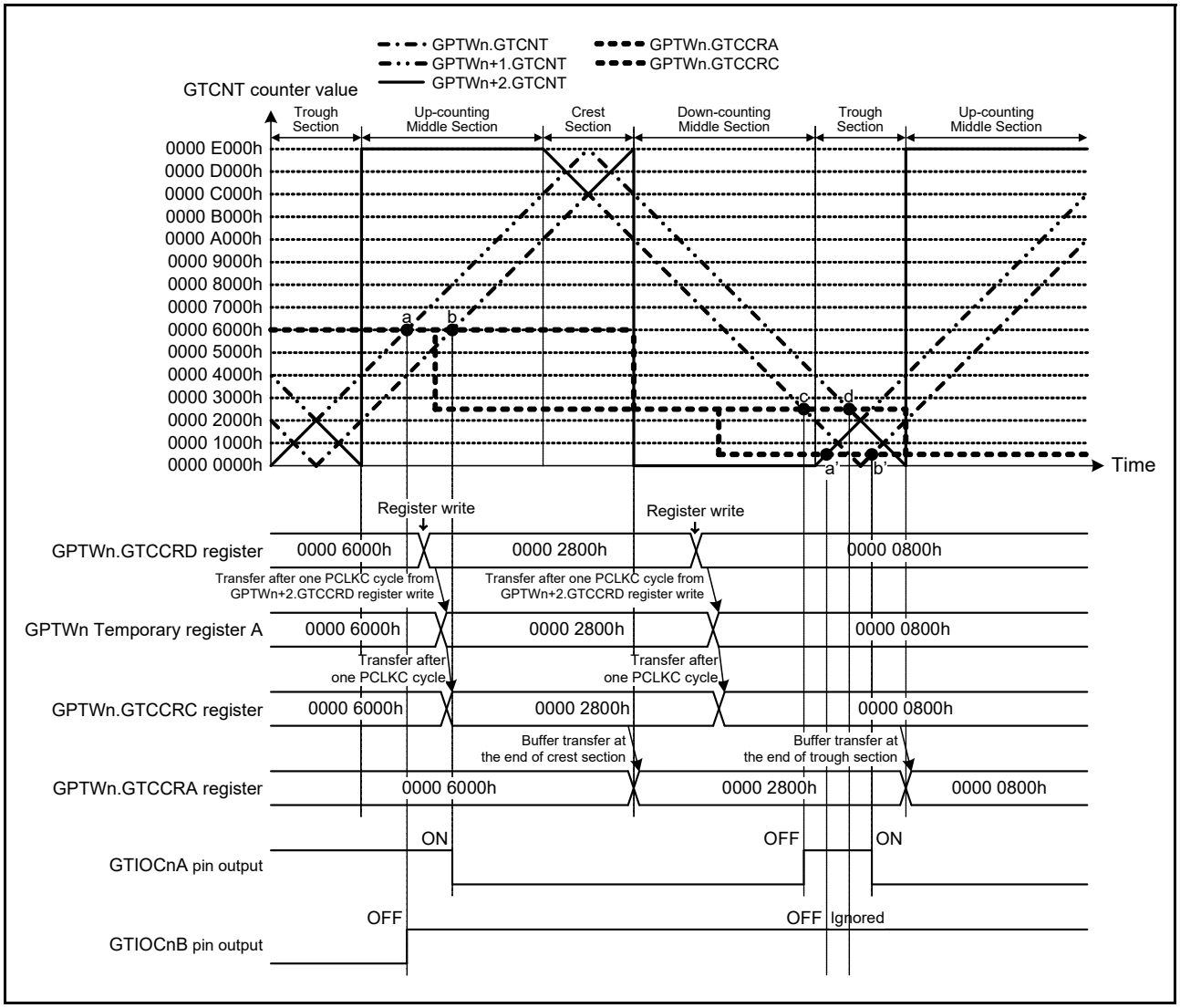


Figure 24.81 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (9) (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0000 2000h, compare match generation order: c → a' → d → b') (n = 0, 4)

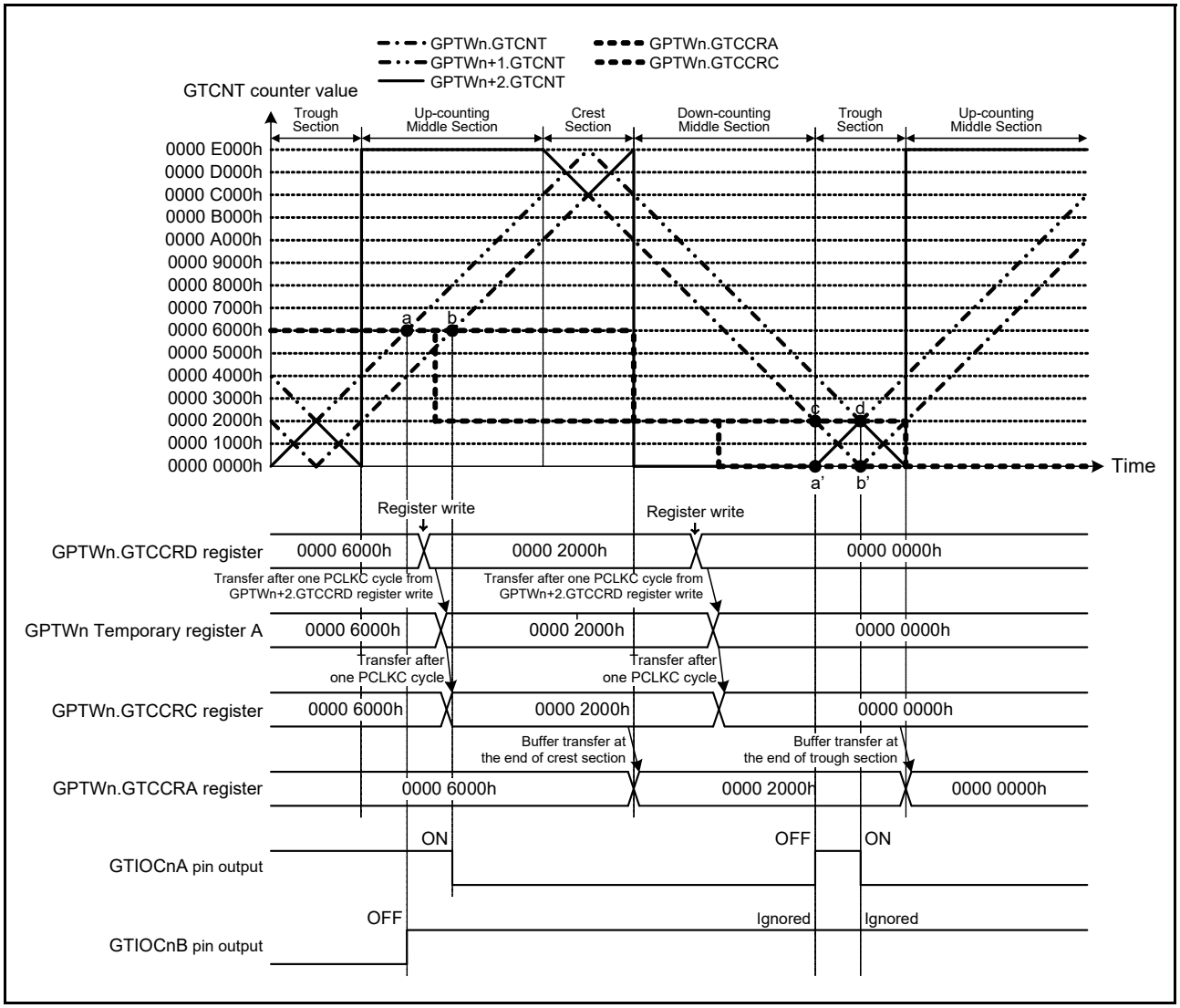


Figure 24.82 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (10) (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up counting, GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0000 2000h, compare match generation order: (c, a') → (d, b')) (n = 0, 4)

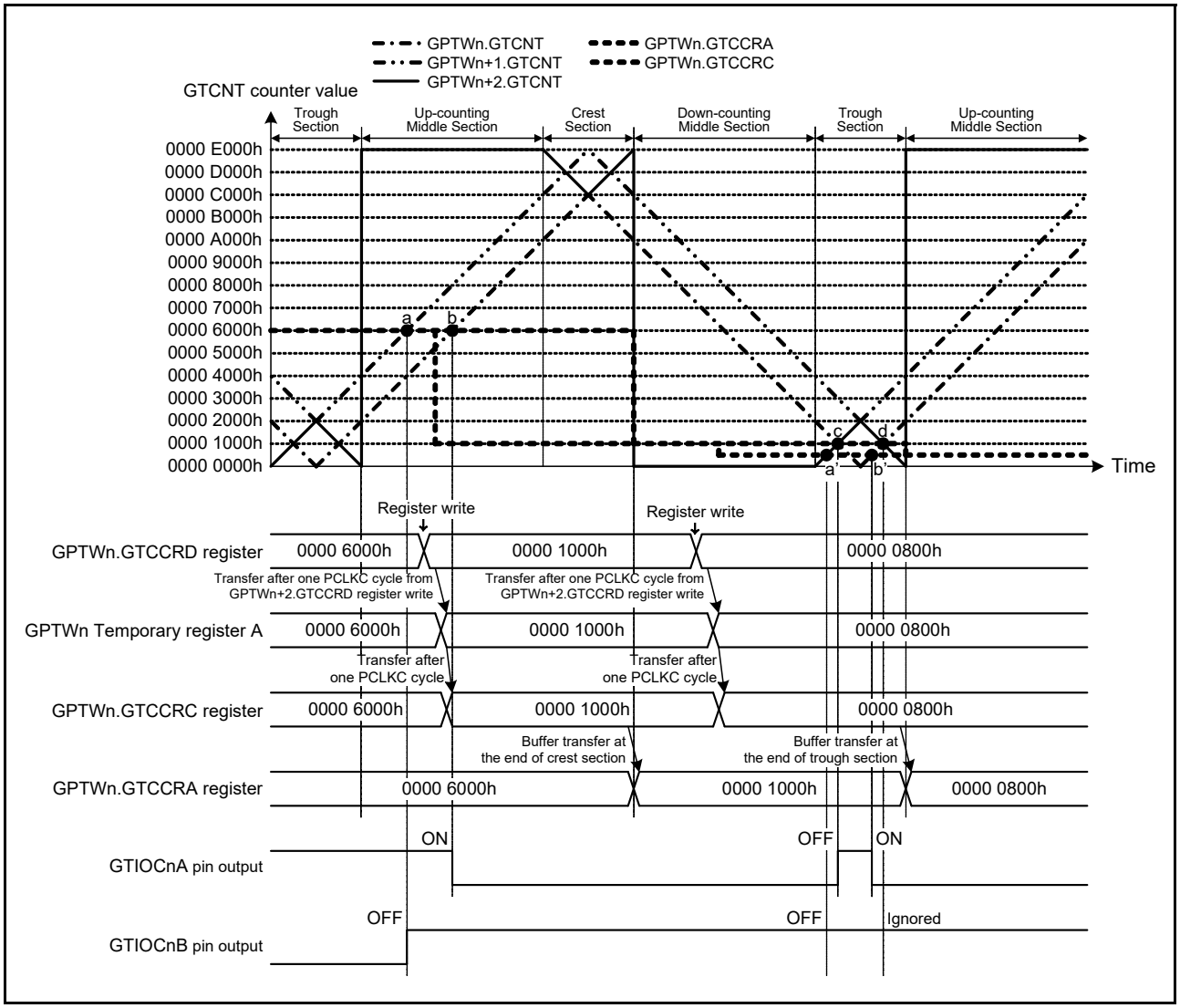


Figure 24.83 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (11) (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up counting, GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0000 2000h, compare match generation order: a' → c → b' → d) (n = 0, 4)

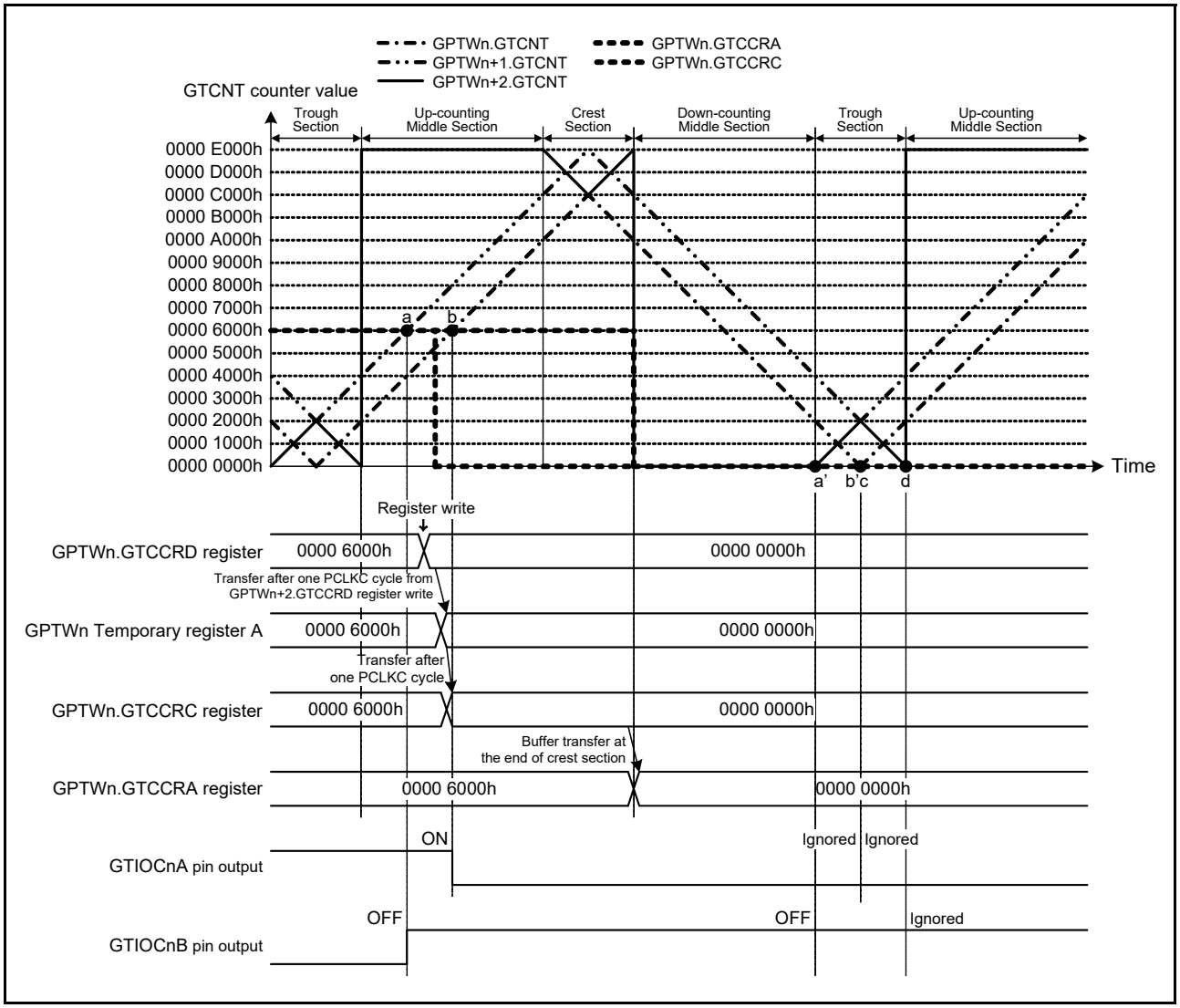


Figure 24.84 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (12) (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0000 2000h, compare match generation order: a' → (b', c) → d) (n = 0, 4)

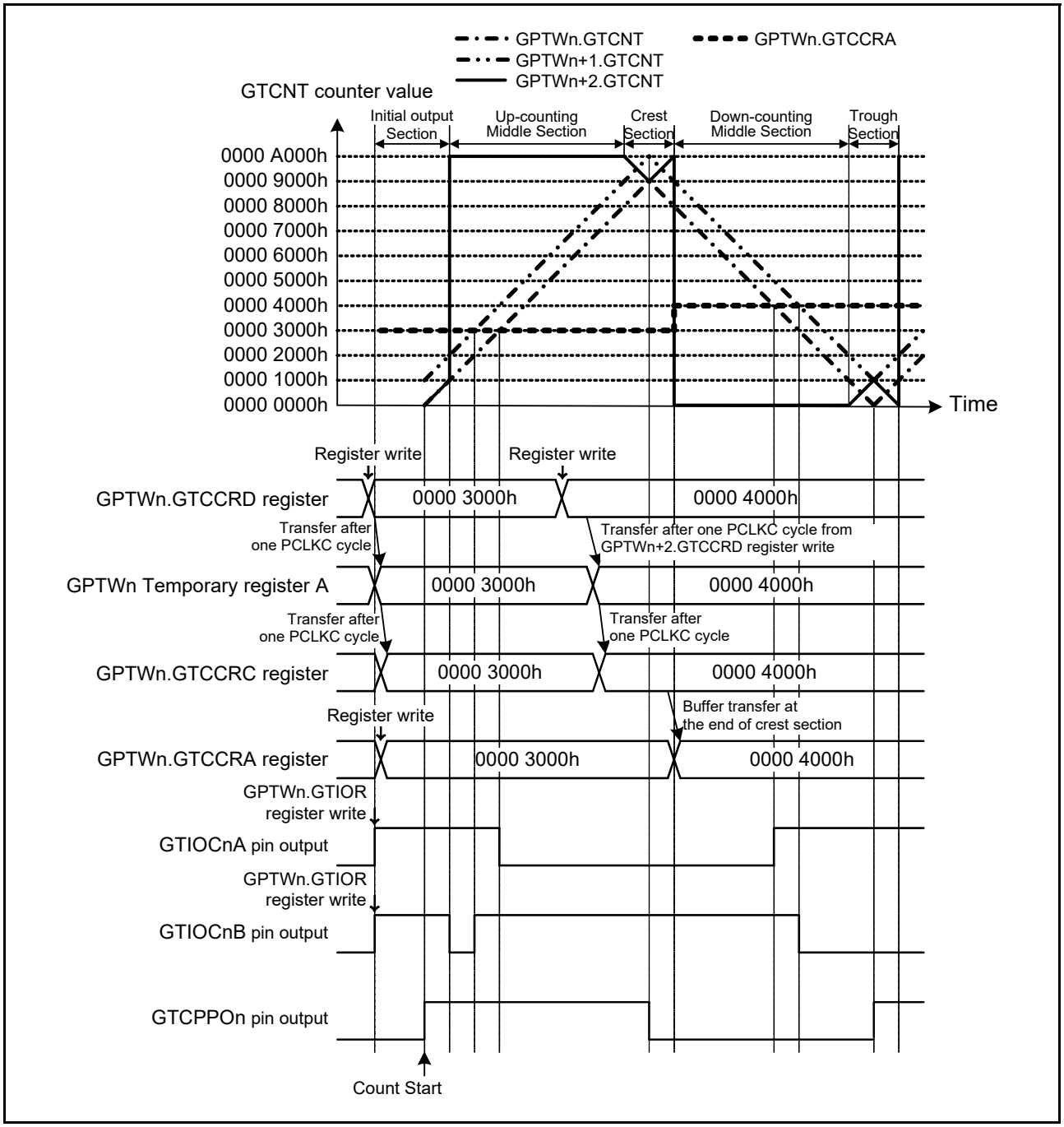


Figure 24.85 Example of Complementary PWM Mode Initial Output Operation (1)
(Complementary PWM mode 1 operation,
GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting,
GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting,
the dead time value is 0000 1000h, in the case that the initial GTCCRA register value is larger than the dead time value) (n = 0, 4)

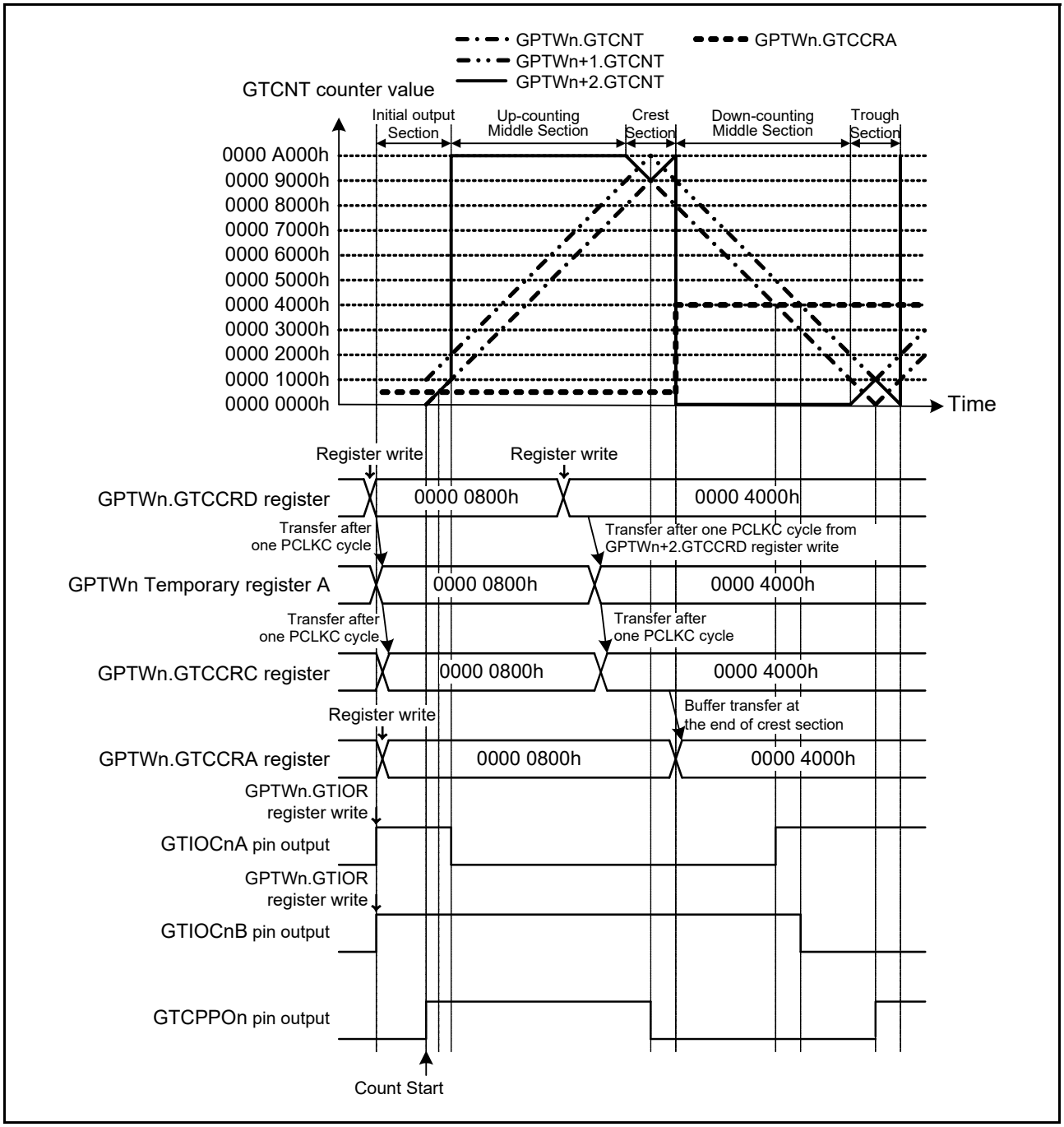


Figure 24.86 Example of Complementary PWM Mode Initial Output Operation (2)
(Complementary PWM mode 1 operation,
GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting,
GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting,
the dead time value is 0000 1000h, in the case that the initial GTCCRA register value is equal to or less than the dead time value) (n = 0, 4)

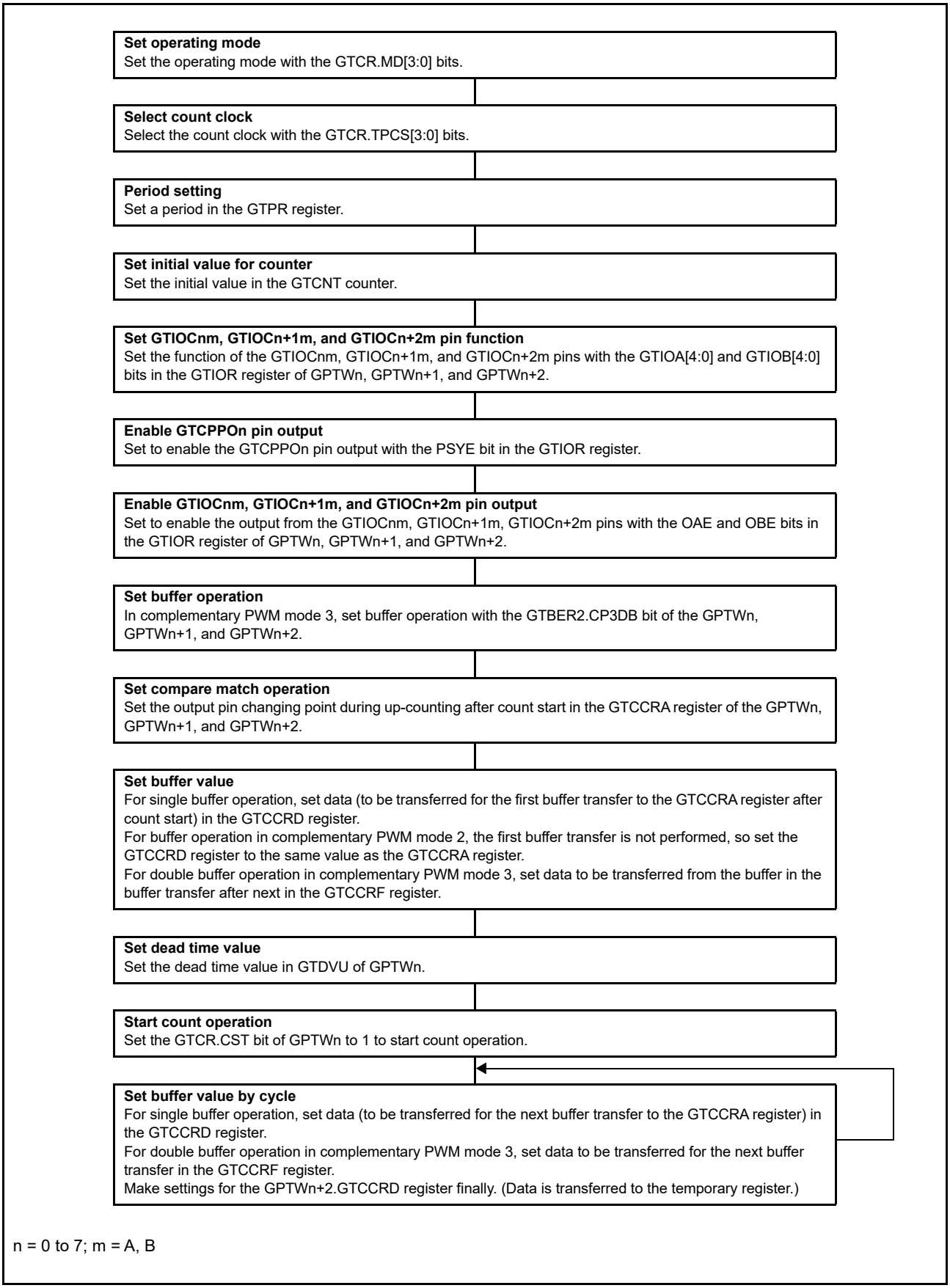


Figure 24.87 Example for Setting Complementary PWM Mode 1, 2, 3

(8) Complementary PWM Mode 4

In complementary PWM mode 4, the value written to the GTCCRD and GTCCRF registers is immediately applied to compare match operation by transferring data also to the GTCCRA register during buffer transfer to a temporary register before crest or trough transfer timing.

Figure 24.88 shows the block diagram in complementary PWM mode 4.

In the configuration of complementary PWM mode 4, a buffer transfer path from the GTCCRD register to the GTCCRC and GTCCRA registers and a buffer transfer path from the GTCCRF register to the GTCCRE and GTCCRA registers are added to other complementary PWM modes shown in Figure 24.64.

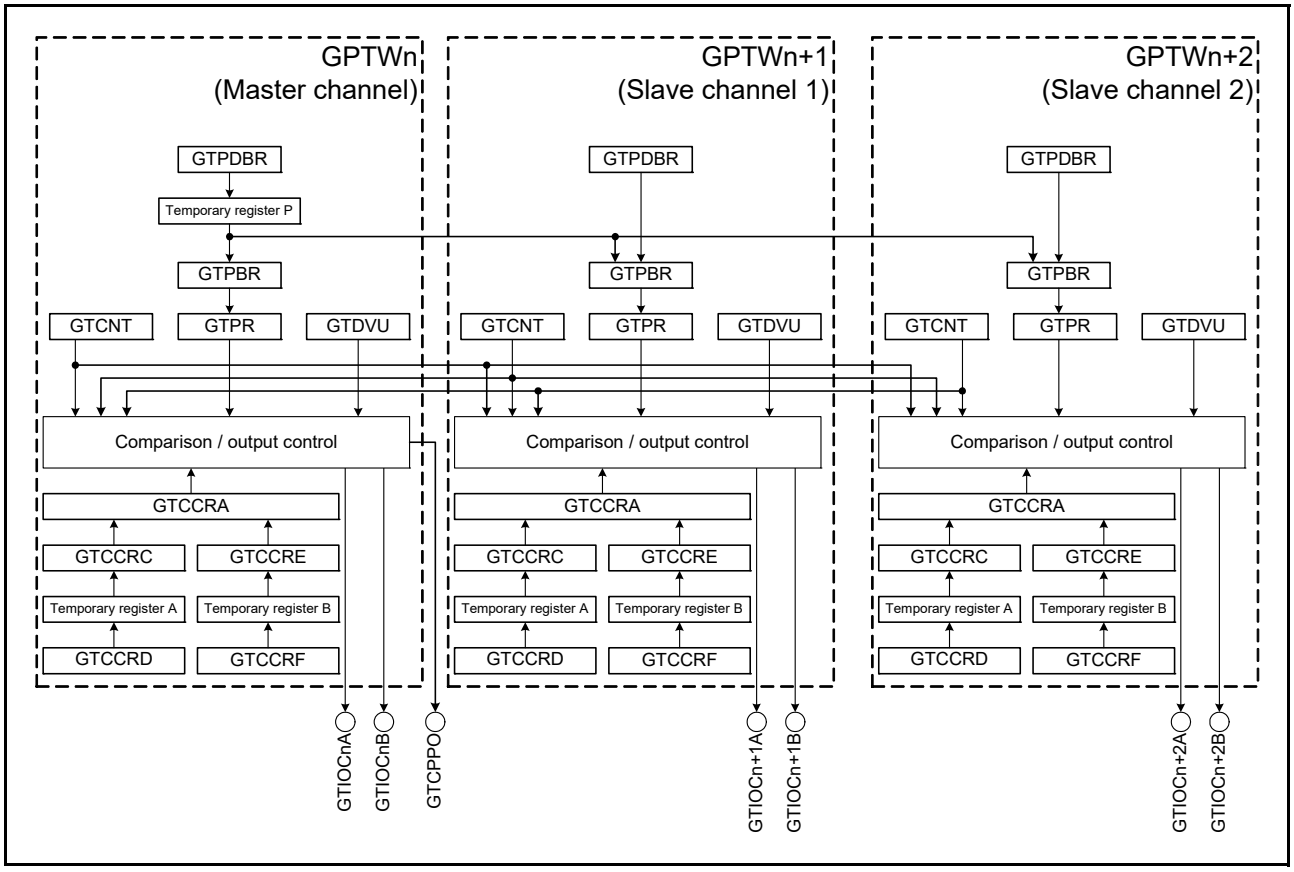


Figure 24.88 Block Diagram in Complementary PWM Mode 4 (n = 0, 4)

Count operation is performed in the same way as complementary PWM modes 1 to 3. See Table 24.12 and Table 24.13.

With respect to buffer operation and PWM waveform changes in complementary PWM mode 4, buffer transfer (shown in Figure 24.88) from the GTCCRD and GTCCRF registers is added based on the complementary PWM mode 3 operation.

Buffer transfer and PWM waveform are controlled according to operation section, state of comparison with the GTCCRA register, and write value.

The double buffer function by writing to the GTCCRF register can be enabled or disabled with the GTBER2.CP3DB bit. In double buffer operation, the value written to the GTCCRD register is used as a compare match value for positive-phase OFF (negative-phase ON) during down-counting, and the value written to the GTCCRF register is used as a compare match value for negative-phase OFF (positive-phase ON) during up-counting. Transfer register, transfer value, and PWM output changes are controlled by operation section (where the value is written), state of comparison with the GTCCRA register, and write value.

Double buffer operation is guaranteed only in the up-counting middle section and down-counting middle section. Setting a value not larger than the dead time value and not less than the count cycle is prohibited.

In single buffer operation, a compare match value is written only to the GTCCRD register, and transfer register, transfer value, and PWM output changes are controlled by operation section (where the value is written), state of comparison with the GTCCRA register, and write value.
Transfer from the GTCCRD register to the temporary register A and transfer from the GTCCRF register to the temporary register B are performed in the same way as other complementary PWM modes. Transfers are concurrently performed in three channels by writing a value to the GPTWn+2.GTCCRD register. Transfer from the GTCCRD register to the GTCCRC register, GTCCRA register, temporary register B, and GTCCRE register and transfer from the GTCCRF register to the GTCCRE register and the GTCCRA register are performed at the same time as the above-mentioned transfer to the temporary register.
Table 24.17 and Table 24.18 show immediate buffer transfer (for transfer to the temporary register by writing a value to the GPTWn+2.GTCCRD register) to the GTCCRC and GTCCRA registers by writing a value to the GTCCRD register during single buffer operation in complementary PWM mode 4 for each compare match state in each operation section. Transfers (from the GTCCRD register to the temporary register A, from the temporary register A to the GTCCRC register, and from the GTCCRC register to the GTCCRA register) other than those shown in Table 24.17 and Table 24.18 are the same as single buffer transfer in complementary PWM mode 3 shown in Table 24.14.
Table 24.19 and Table 24.20 show immediate buffer transfer (for transfer to the temporary register by writing a value to the GPTWn+2.GTCCRD register) to each register by writing a value to the GTCCRD and GTCCRF registers during double buffer operation in complementary PWM mode 4 for each compare match state in each operation section. Transfers (from the GTCCRD register to the temporary register A, from the GTCCRF register to the temporary register B, from the temporary register A to the GTCCRC register, from the temporary register B to the GTCCRE register, and from the GTCCRC and GTCCRE registers to the GTCCRA register) other than those shown in Table 24.19 and Table 24.20 are the same as double buffer transfer in complementary PWM mode 3 shown in Table 24.15.

Table 24.17 Immediate Single Buffer Transfer from the GTCCRD Register in Complementary PWM Mode 4 (1)

Operation Section	Compare Match State	Immediate Transfer Destination Register	
		GTCCRC	GTCCRA
Up-counting middle section	Before up-counting compare match	GTCCRD	• In the case of GTCCRD > GPTWn+1.GTCNT GTCCRD • In the case of GTCCRD ≤ GPTWn+1.GTCNT GPTWn+1.GTCNT Negative-phase OFF
	Up-counting dead time period	GTCCRD	No transfer
	After up-counting compare match	GTCCRD	No transfer
Up-counting crest section	Before up-counting compare match	Before down-counting compare match GTCCRD After down-counting dead time start No transfer	Before down-counting compare match • In the case of GTCCRD > GPTWn+1.GTCNT GTCCRD • In the case of GTCCRD ≤ GPTWn+1.GTCNT GPTWn+1.GTCNT Negative-phase OFF After down-counting dead time start No transfer
	Up-counting dead time period	Before down-counting compare match GTCCRD After down-counting dead time start No transfer	No transfer
	After up-counting compare match	Before down-counting compare match • In the case of GTCCRD < GPTWn+2.GTCNT GTCCRD • In the case of GTCCRD ≥ GPTWn+2.GTCNT GPTWn+2.GTCNT Negative-phase OFF After down-counting dead time start No transfer	No transfer
Down-counting crest section	Before down-counting compare match	Up-counting dead time period • In the case of GTCCRD < GPTWn+1.GTCNT GTCCRD • In the case of GTCCRD ≥ GPTWn+1.GTCNT GPTWn+1.GTCNT Negative-phase OFF After up-counting compare match • In the case of GTCCRD < GPTWn.GTCNT GTCCRD • In the case of GTCCRD ≥ GPTWn.GTCNT GPTWn.GTCNT Positive-phase OFF	No transfer
	Down-counting dead time period	No transfer	No transfer
	After down-counting compare match	No transfer	No transfer

Table 24.18 Immediate Single Buffer Transfer from the GTCCRD Register in Complementary PWM Mode 4 (2)

Operation Section	Compare Match State	Immediate Transfer Destination Register	
		GTCCRC	GTCCRA
Down-counting middle section	Before down-counting compare match	GTCCRD	• In the case of GTCCRD < GPTWn.GTCNT GTCCRD • In the case of GTCCRD ≥ GPTWn.GTCNT GPTWn.GTCNT Positive-phase OFF
	Down-counting dead time period	GTCCRD	No transfer
	After down-counting compare match	GTCCRD	No transfer
Down-counting trough section	Before down-counting compare match	Before up-counting compare match GTCCRD After up-counting dead time start No transfer	Before up-counting compare match • In the case of GTCCRD < GPTWn.GTCNT GTCCRD • In the case of GTCCRD ≥ GPTWn.GTCNT GPTWn.GTCNT Positive-phase OFF After up-counting dead time start No transfer
	Down-counting dead time period	Before up-counting compare match GTCCRD After up-counting dead time start No transfer	No transfer
	After down-counting compare match	Before up-counting compare match • In the case of GTCCRD > GPTWn+2.GTCNT GTCCRD • In the case of GTCCRD ≤ GPTWn+2.GTCNT GPTWn+2.GTCNT Negative-phase OFF After up-counting dead time start No transfer	No transfer
Up-counting trough section	Before up-counting compare match	Down-counting dead time period • In the case of GTCCRD > GPTWn.GTCNT GTCCRD • In the case of GTCCRD ≤ GPTWn.GTCNT GPTWn.GTCNT Positive-phase ON After down-counting compare match • In the case of GTCCRD > GPTWn+1.GTCNT GTCCRD • In the case of GTCCRD ≤ GPTWn+1.GTCNT GPTWn+1.GTCNT Negative-phase OFF	No transfer
	Up-counting dead time period	No transfer	No transfer
	After up-counting compare match	No transfer	No transfer

Table 24.19 Immediate Double Buffer Transfer from GTCCRD and GTCCRF Registers in Complementary PWM Mode 4 (1)

Operation Section	Compare Match State	Immediate Transfer Destination Register		
		GTCCRC	GTCCRE	GTCCRA
Up-counting middle section	Before up-counting compare match	GTCCRD	GTCCRF	• In the case of GTCCRF > GPTWn+1.GTCNT GTCCRF • In the case of GTCCRF ≤ GPTWn+1.GTCNT GPTWn+1.GTCNT Negative-phase OFF
	Up-counting dead time period	GTCCRD	GTCCRF	No transfer
	After up-counting compare match	GTCCRD	GTCCRF	No transfer

Table 24.20 Immediate Double Buffer Transfer from GTCCRD and GTCCRF Registers in Complementary PWM Mode 4 (2)

Operation Section	Compare Match State	Immediate Transfer Destination Register		
		GTCCRC	GTCCRE	GTCCRA
Down-counting middle section	Down-counting compare match	GTCCRD	GTCCRF	• In the case of GTCCRD < GPTWn+1.GTCNT GTCCRD • in the case of GTCCRD ≥ GPTWn.GTCNT GPTWn.GTCNT Positive-phase OFF
	Down-counting dead time period	GTCCRD	GTCCRF	No transfer
	After down-counting compare period	GTCCRD	GTCCRF	No transfer

Figure 24.89 to Figure 24.93 show examples of single buffer operation for each operation section in complementary PWM mode 4.

- Up-Counting Middle Section: Figure 24.89
- Up-Counting Crest Section: Figure 24.90
- Down-Counting Crest Section: Figure 24.91
- Down-Counting Middle Section: Figure 24.92
- Down-Counting Trough Section: Figure 24.93

Figure 24.94 to Figure 24.97 show examples of double buffer operation for each operation section in complementary PWM mode 4.

- Up-Counting Middle Section: Figure 24.94 and Figure 24.95
- Down-Counting Middle Section: Figure 24.96 and Figure 24.97

Figure 24.98 shows an example for setting complementary PWM mode 4.

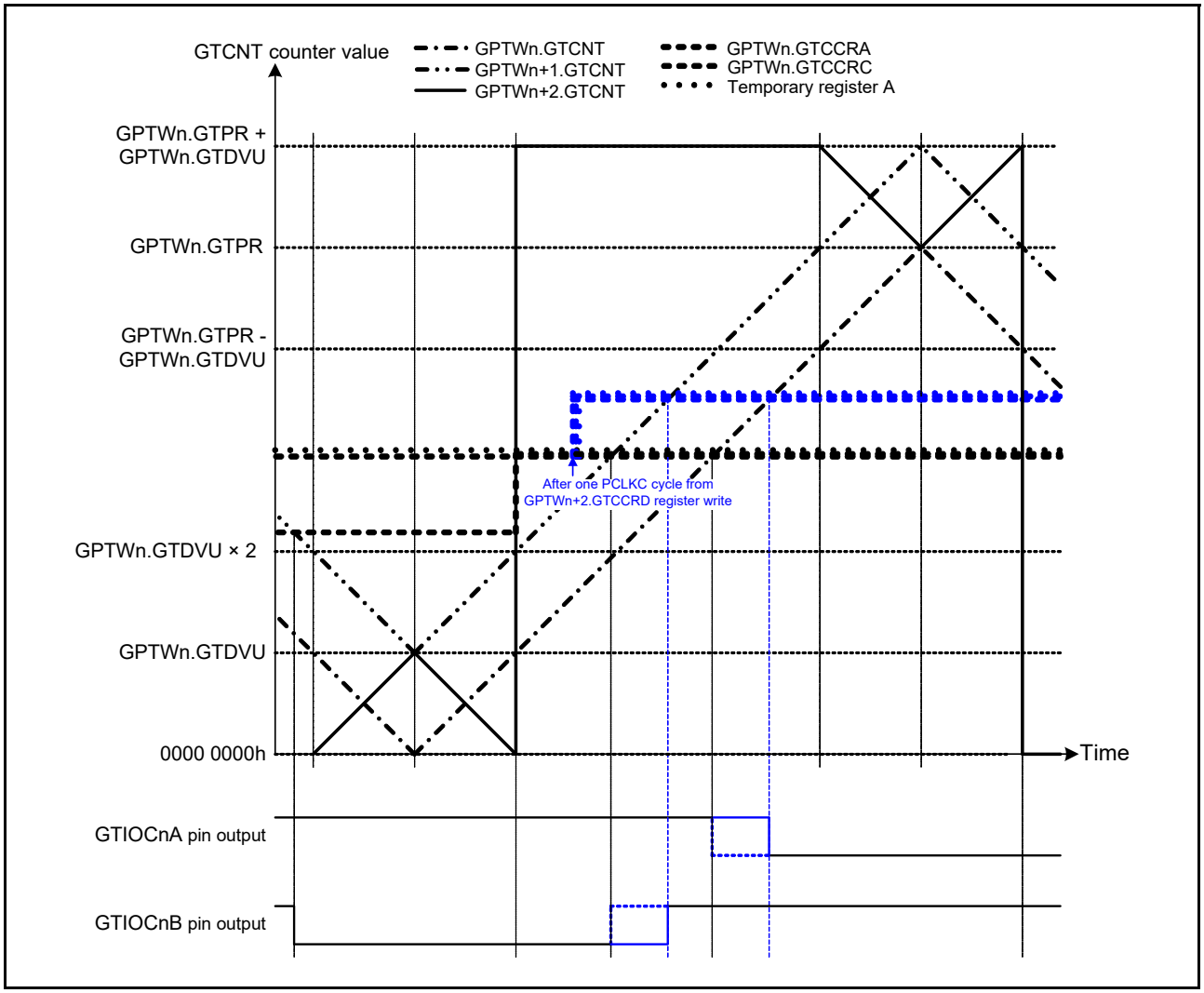


Figure 24.89 Example of Complementary PWM Mode 4 Single Buffer Operation (Up-Counting Middle Section) (Complementary PWM mode 4 single buffer operation, GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting, when a value larger than the GPTWn+1.GTCNT value is written to the GTCCRD register before up-counting compare match) (n = 0, 4)

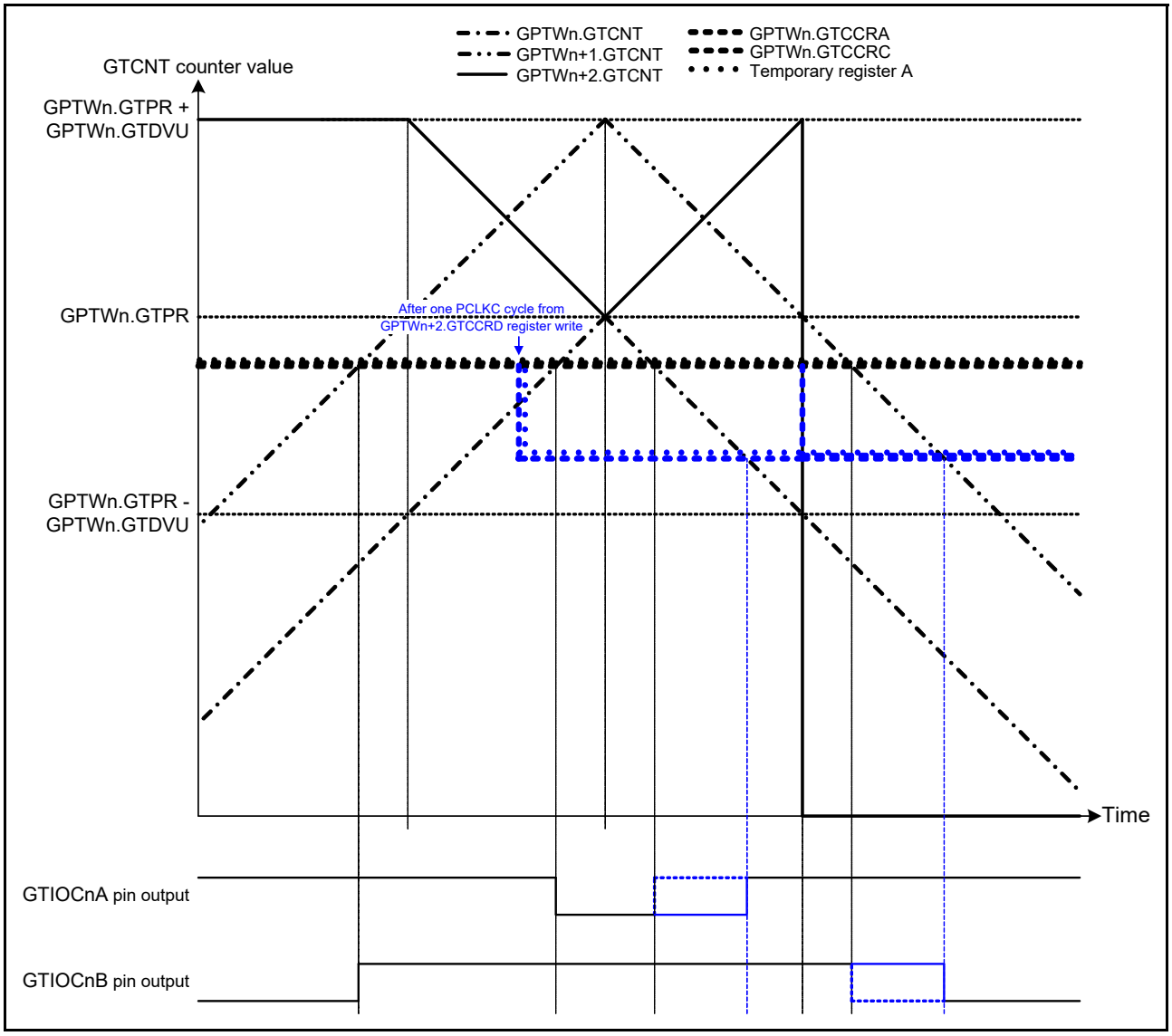


Figure 24.90 Example of Complementary PWM Mode 4 Single Buffer Operation (Up-Counting Crest Section) (Complementary PWM mode 4 single buffer operation, GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting, when a value is written to the GTCCRD register during up-counting dead time) (n = 0, 4)

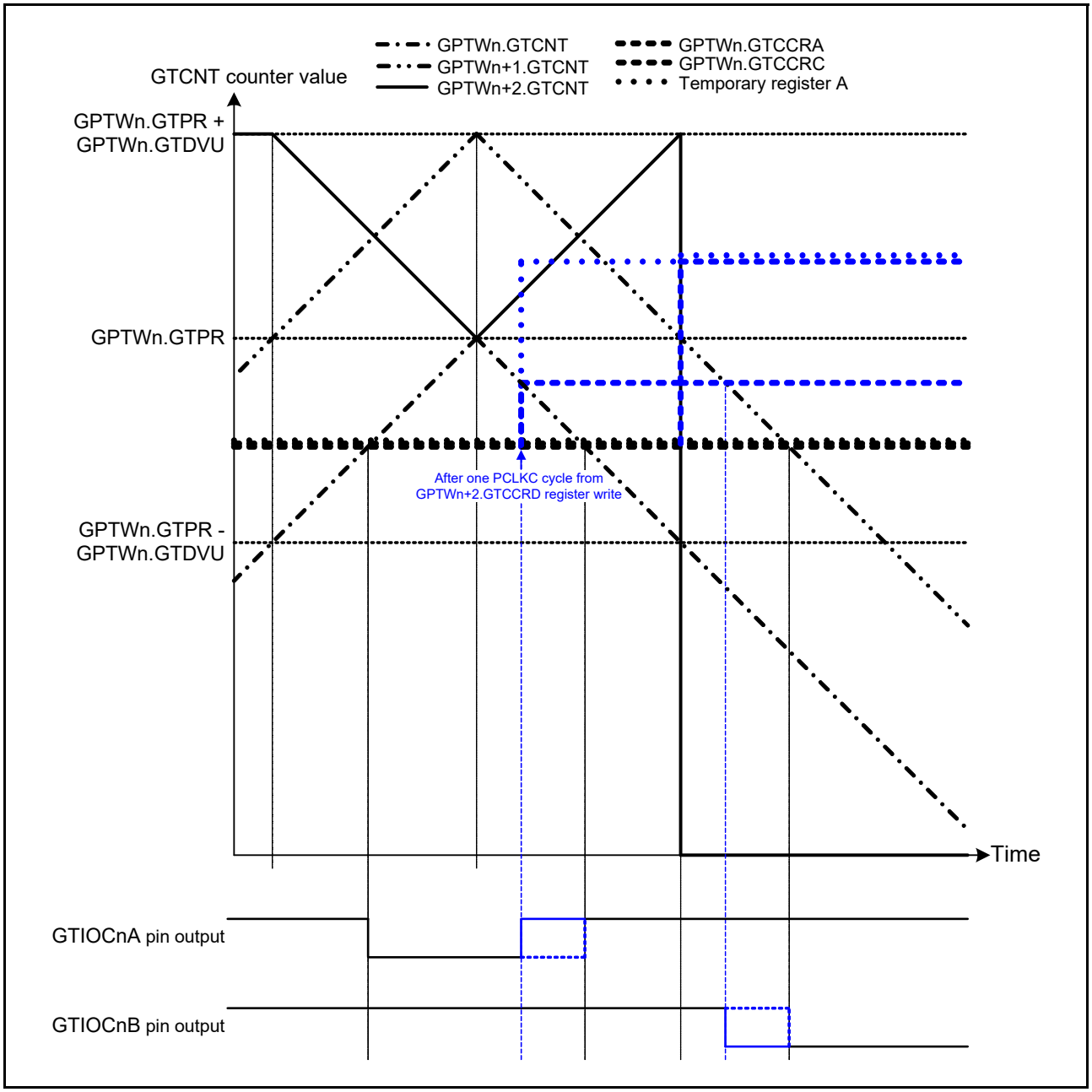


Figure 24.91 Example of Complementary PWM Mode 4 Single Buffer Operation (Down-Counting Crest Section) (Complementary PWM mode 4 single buffer operation, GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting, when a value large than and equal to GPTWn.GTCNT value is written to the GTCCRD register before down-counting compare match and after up-counting compare match) (n = 0, 4)

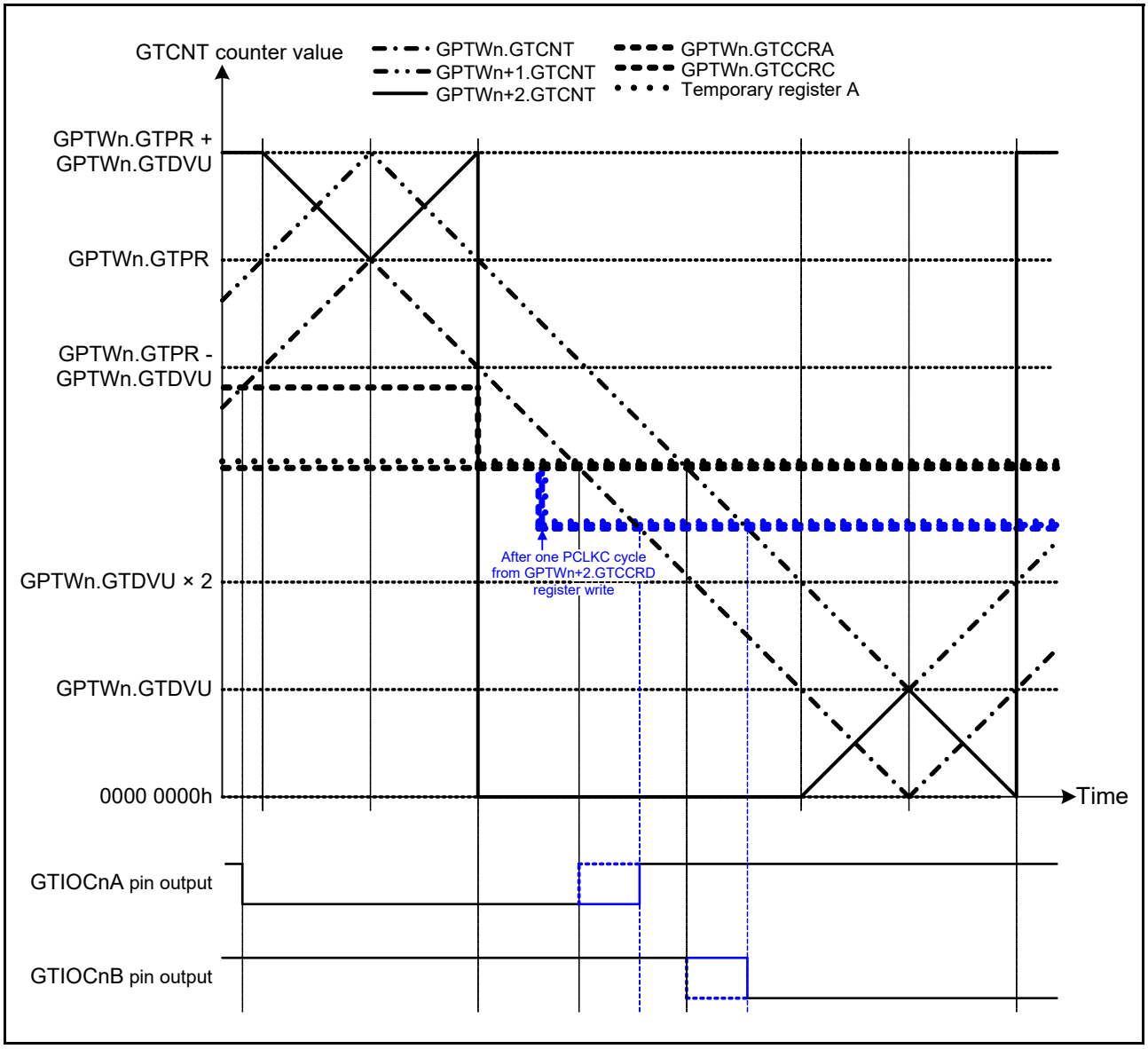


Figure 24.92 Example of Complementary PWM Mode 4 Single Buffer Operation (Down-Counting Middle Section)
(Complementary PWM mode 4 single buffer operation,
GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting,
GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting,
When a value less than GPTWn.GTCNT value is written to the GTCCRD register before down-counting compare match) (n = 0, 4)

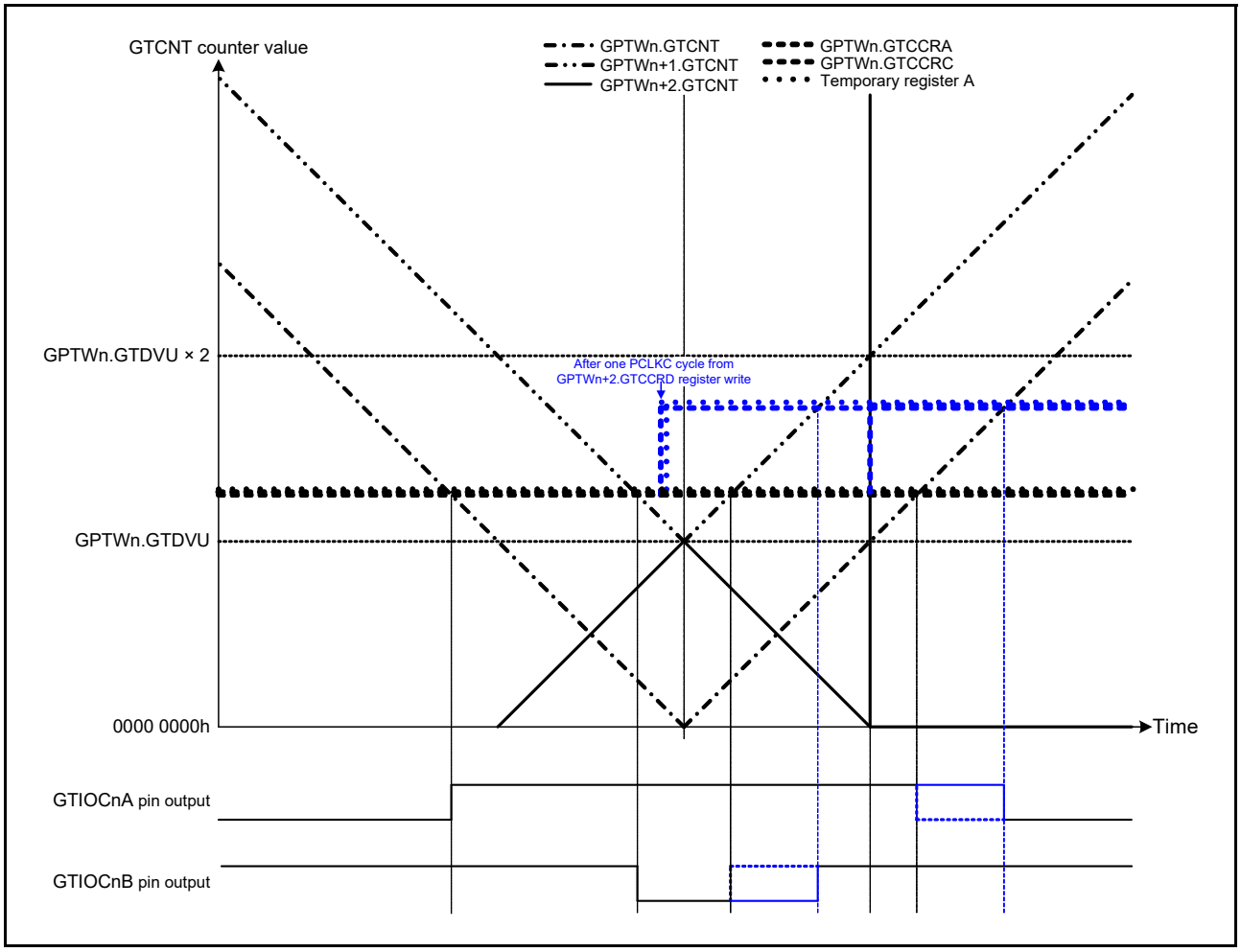


Figure 24.93 Example of Complementary PWM Mode 4 Single Buffer Operation (Down-Counting Trough Section)
(Complementary PWM mode 4 single buffer operation,
GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting,
GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting,
When a value large than GPTWn+2.GTCNT value is written to the GTCCRD register after down-counting compare match) (n = 0, 4)

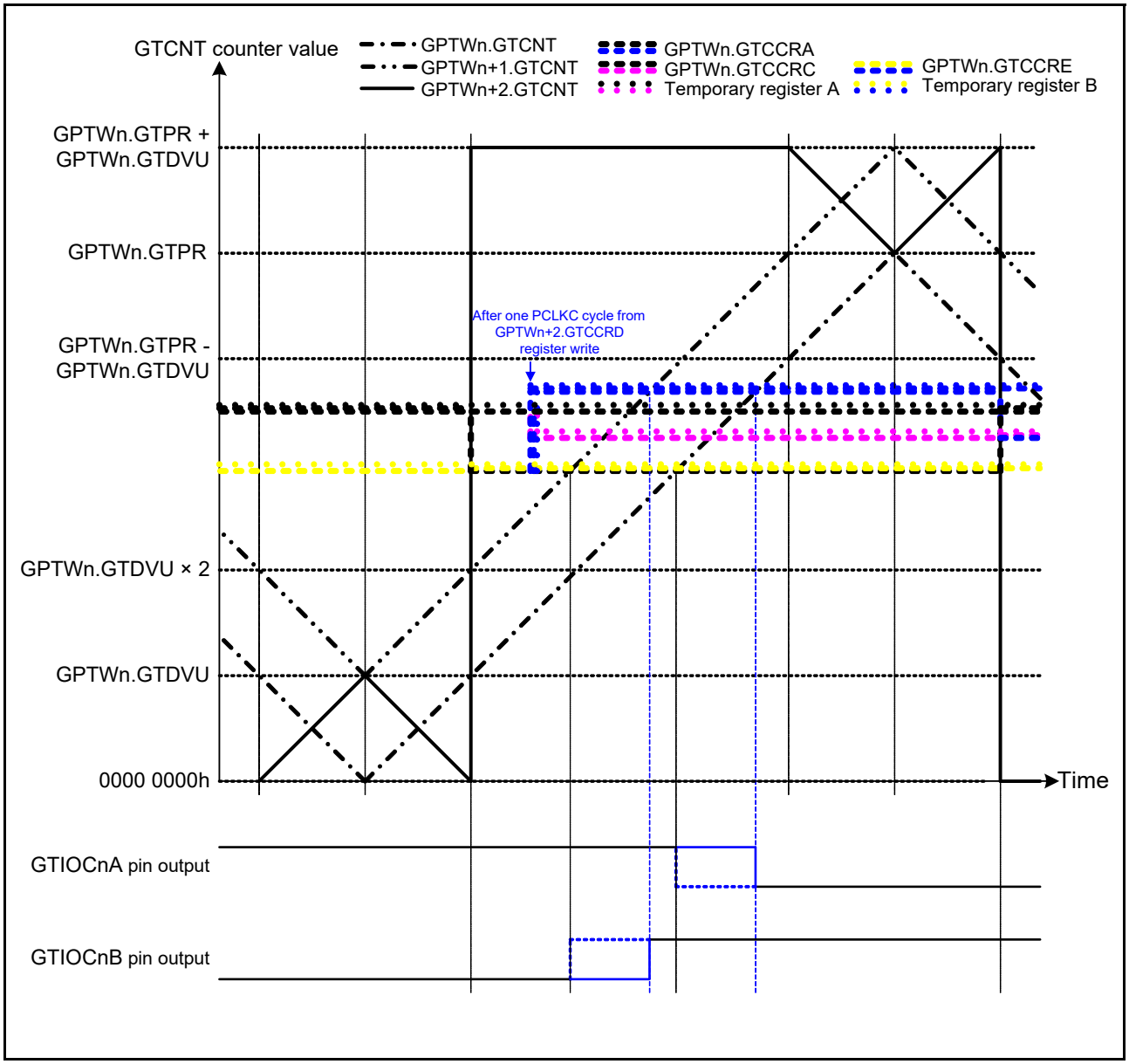


Figure 24.94 Example of Complementary PWM Mode 4 Double Buffer Operation (Up-Counting Middle Section) (Complementary PWM mode 4 Double buffer operation, GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting, When a value larger than GPTWn+1.GTCNT is written to the GTCCRF register before the up-counting compare match) (n = 0, 4)

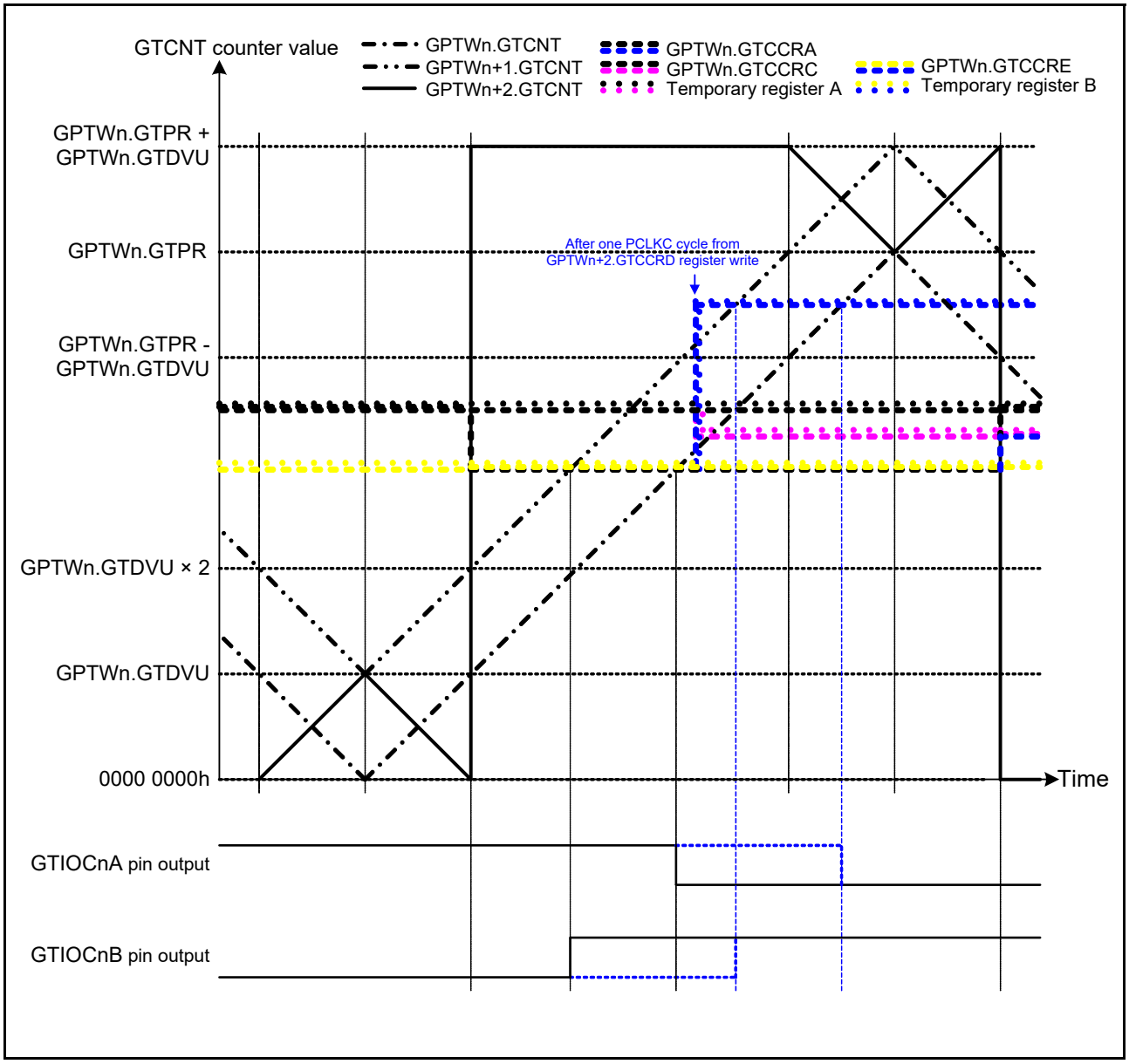


Figure 24.95 Example of Complementary PWM Mode 4 Double Buffer Operation (Up-Counting Middle Section) (Complementary PWM mode 4 Double buffer operation, GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting, When writing to the GTCCRF register after an up-counting compare match) (n = 0, 4)

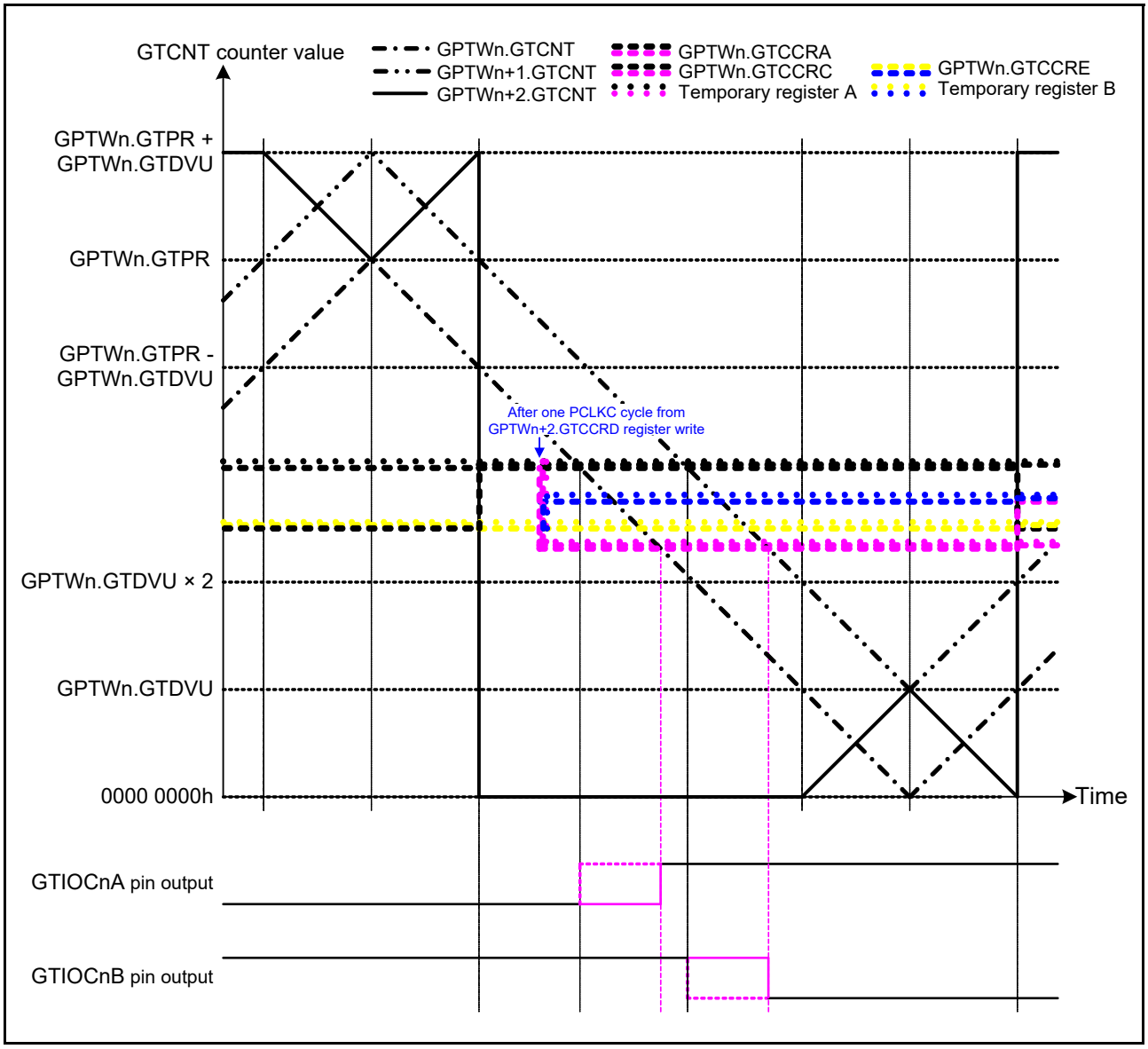


Figure 24.96 Example of Complementary PWM Mode 4 Double Buffer Operation (Down-Counting Middle Section)
(Complementary PWM mode 4 Double buffer operation,
GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting,
GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting,
When a value less than GPTWn.GTCNT is written to the GTCCRD register before the down-counting compare match) (n = 0, 4)

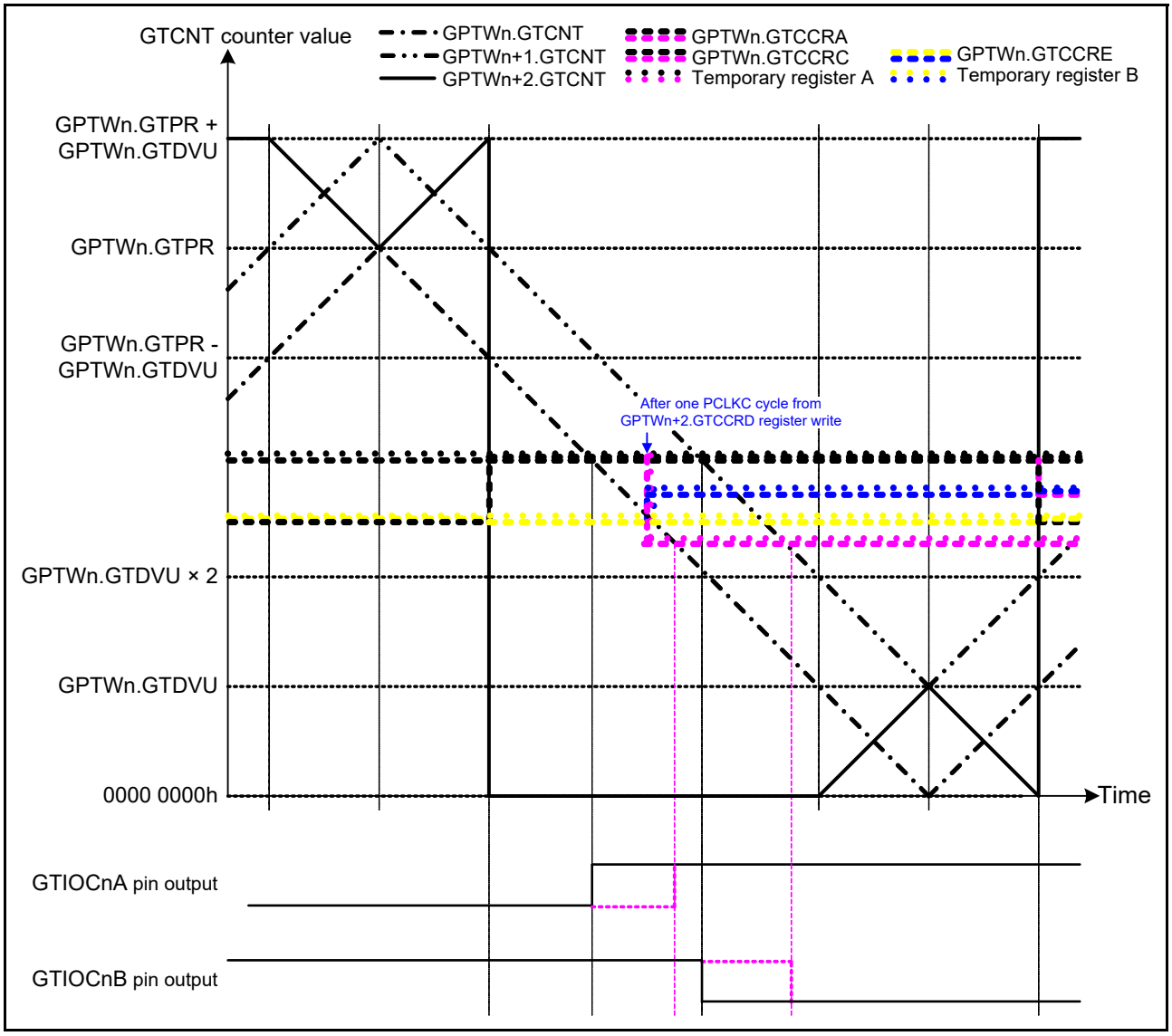
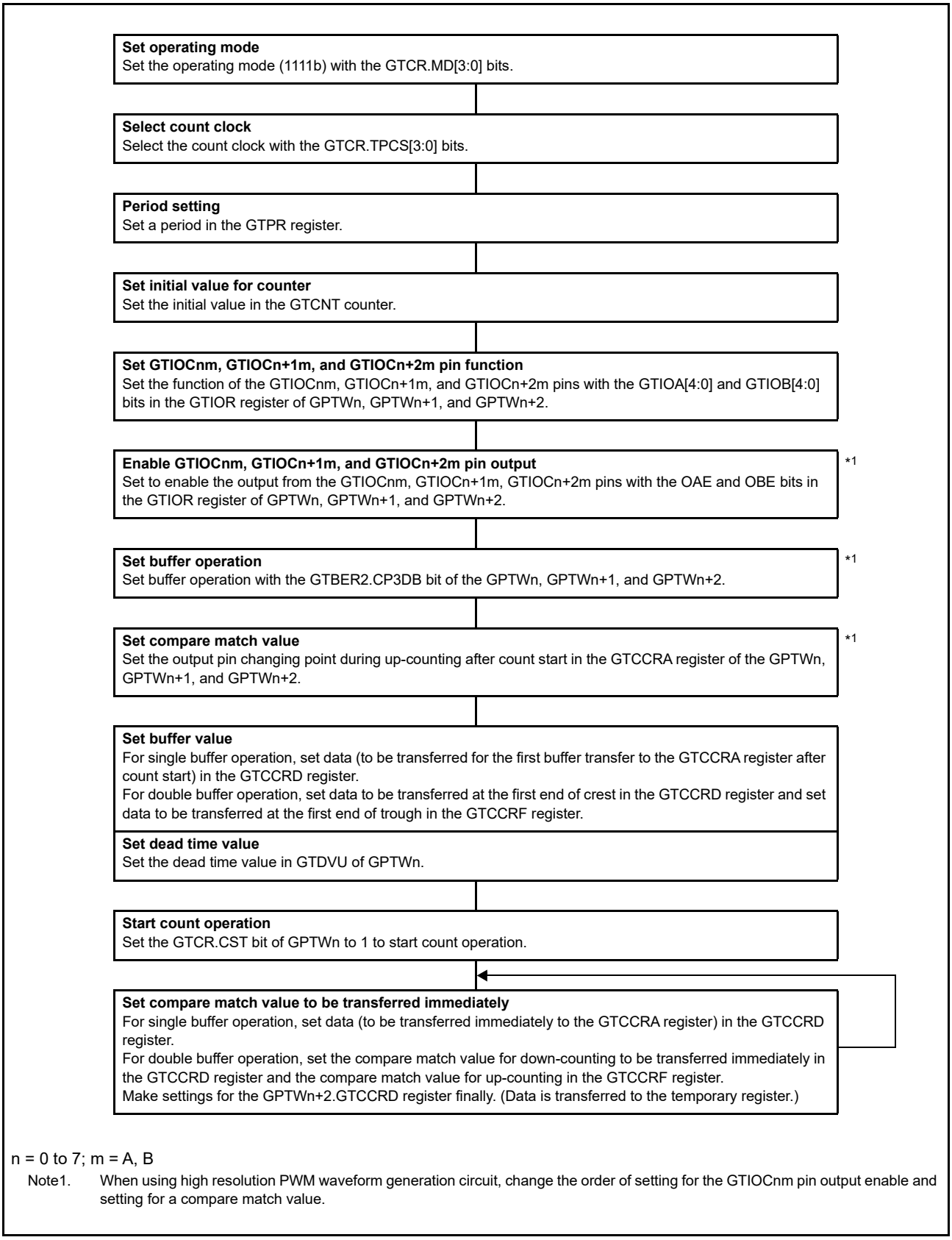


Figure 24.97 Example of Complementary PWM Mode 4 Double Buffer Operation (Down-Counting Middle Section)
(Complementary PWM mode 4 Double buffer operation,
GTIOCnA pin = Low/GTIOCnB pin = High at GTCCRA register compare match during up-counting,
GTIOCnA pin = High/GTIOCnB pin = Low at GTCCRA register compare match during down-counting,
When writing to the GTCCRD register during the down-counting dead time period) (n = 0, 4)



n = 0 to 7; m = A, B

Note1. When using high resolution PWM waveform generation circuit, change the order of setting for the GTIOCnm pin output enable and setting for a compare match value.

Figure 24.98 Example for Setting Complementary PWM Mode 4

24.3.4 Automatic Dead Time Setting Function

By setting the GTDTCR register, a compare match value for a negative waveform with dead time obtained by a compare match value for a positive waveform (the GTCCRA register value) and specified dead time values (the GTDVU and GTDVD registers values) can automatically be set to the GTCCRB register.
The automatic dead time setting function can be used in sawtooth-wave one-shot pulse mode and all the triangle PWM modes.
Dead time can be separately set for the first half and second half of a waveform. Dead time for the changing point in the first half of a negative waveform is set in the GTDVU register and that in the second half is set in the GTDVD register. The same dead time can also be set for the first and second halves by setting the GTDTCR.TDFER bit to 1.
The GTDBU register can be used as a buffer register of the GTDVU register, and the GTDBD register can be used as a buffer register of the GTDVD register. Buffer transfer is performed at the end of the cycle (in sawtooth-wave mode: either of an overflow of the GTCNT counter (up-counting), an underflow (down-counting), or the GTCNT counter clearing; in triangle-wave mode: a trough).
The change point of the negative-phase waveform, which is automatically calculated, is obtained by reading the GTCCRB register. Writing to the GTCCRB register is prohibited when the automatic dead time setting function is used.

Do not set the dead-time that makes the change point of the waveform exceeding the count period. When any dead-time setting which would generate a dead-time error is made, adjust the change points of the positive- and negative-phase waveforms to generate waveforms with secured dead-time as shown in Table 24.21. The adjusted change point of the negative-phase waveform is automatically set in the GTCCRB register. An internal signal is used to judge the change point of the positive-phase waveform, thus the value of the GTCCRA register is not updated by the adjusted value.
If the order of the change point becomes inconsistent by adjustment of the waveform change point due to occurrence of dead time errors, or if the change point exceeds the count period even after the adjustment, the complementary relation between the positive- and negative-phases cannot be guaranteed.
If dead-time exceeds the count period by setting 0000 0000h or a value greater than or equal to the setting value of the GTPR register is set in the GTCCRA register in triangle-wave PWM mode, output change is controlled by the output protection function (refer to section 24.8.4, Output Protection Function for GTIOCnm Pin Output (n = 0 to 7; m = A, B)). When GTCCRA register is greater than or equal to [GTPR register + GTDVm register], [GTPR register – 1] is set in the GTCCRB register as the upper limit. (m = U, D)
Automatic setting for a dead time value to the GTCCRB register is performed at the next count clock after the register value for calculating the automatic setting value is updated. In triangle-wave mode, it also can be done at the next count clock from the current crest.

Table 24.21 Adjustment of the Waveform Change Point When a Dead-Time Error Occurs

PWM Output Operating Mode	Count Direction	Period	Condition for Dead-Time Error	Change Point of the Positive-Phase Waveform after Adjustment	Change Point of the Negative-Phase Waveform after Adjustment
Sawtooth-wave one-shot pulse mode	Up-counting	First half	GTCCRA – GTDVU < 0	GTDVU	0
		Second half	GTCCRA + GTDVD > GTPR (GTCCRA + GTDVU > GTPR)*1	GTPR – GTDVD (GTPR – GTDVU)*1	GTPR
	Down-counting	First half	GTCCRA + GTDVU > GTPR	GTPR – GTDVU	GTPR
		Second half	GTCCRA – GTDVD < 0 (GTCCRA + GTDVU < 0)*1	GTDVD (GTDVU)*1	0
Triangle-wave PWM mode 1/2/3	Up-counting	(First half)	GTCCRA – GTDVU≤0	GTDVU + 1	1
	Down-counting	(Second half)	GTCCRA – GTDVD < 0 (GTCCRA + GTDVU < 0)*1	GTDVD (GTDVU)*1	0

Note 1. In the case of GTDTCR.TDFER = 1.

Figure 24.99 to Figure 24.102 show examples of automatic dead time setting function operation. Figure 24.103 and Figure 24.104 show the setting examples.

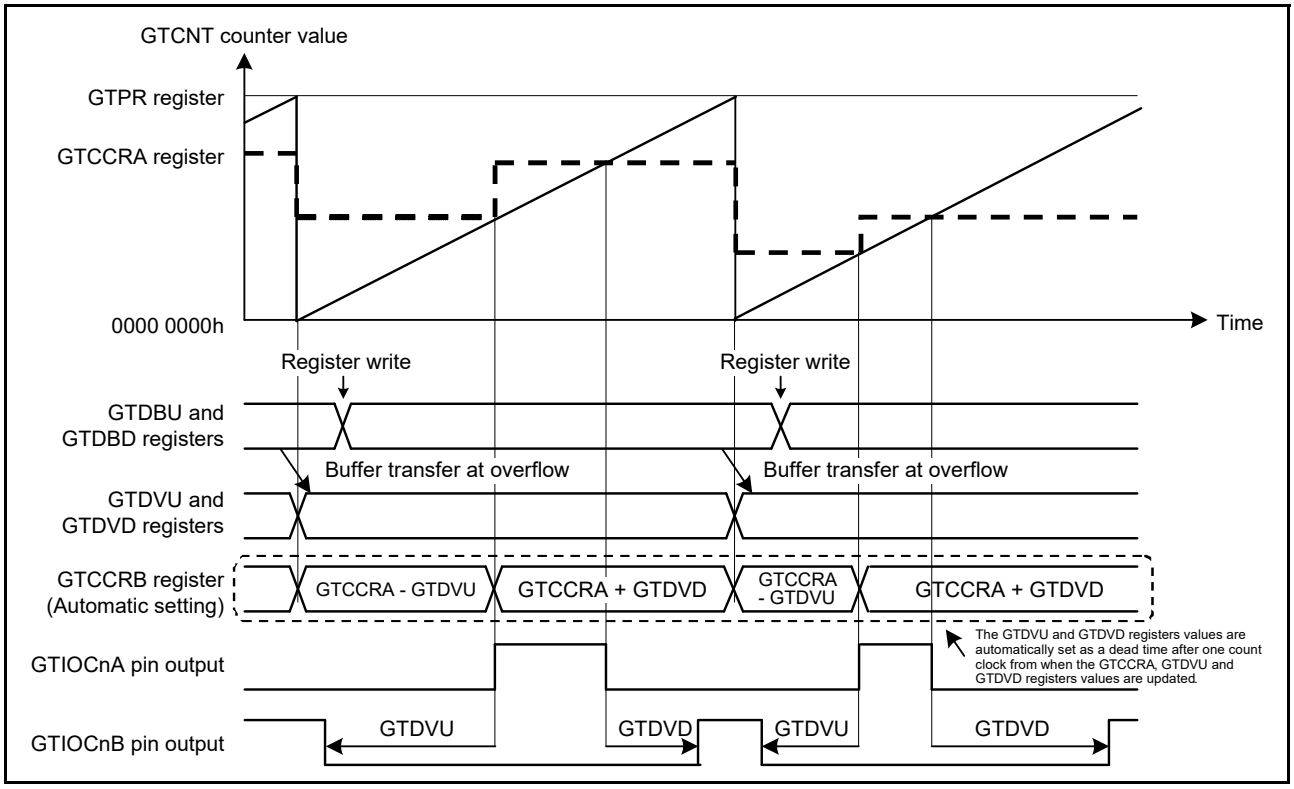


Figure 24.99 Example of Automatic Dead Time Setting Function Operation (Sawtooth-Wave One-Shot Pulse Mode, Up-Counting, GTDVU and GTDVD Registers Set to Buffer Operation, Active Level: High) (n = 0 to 7)

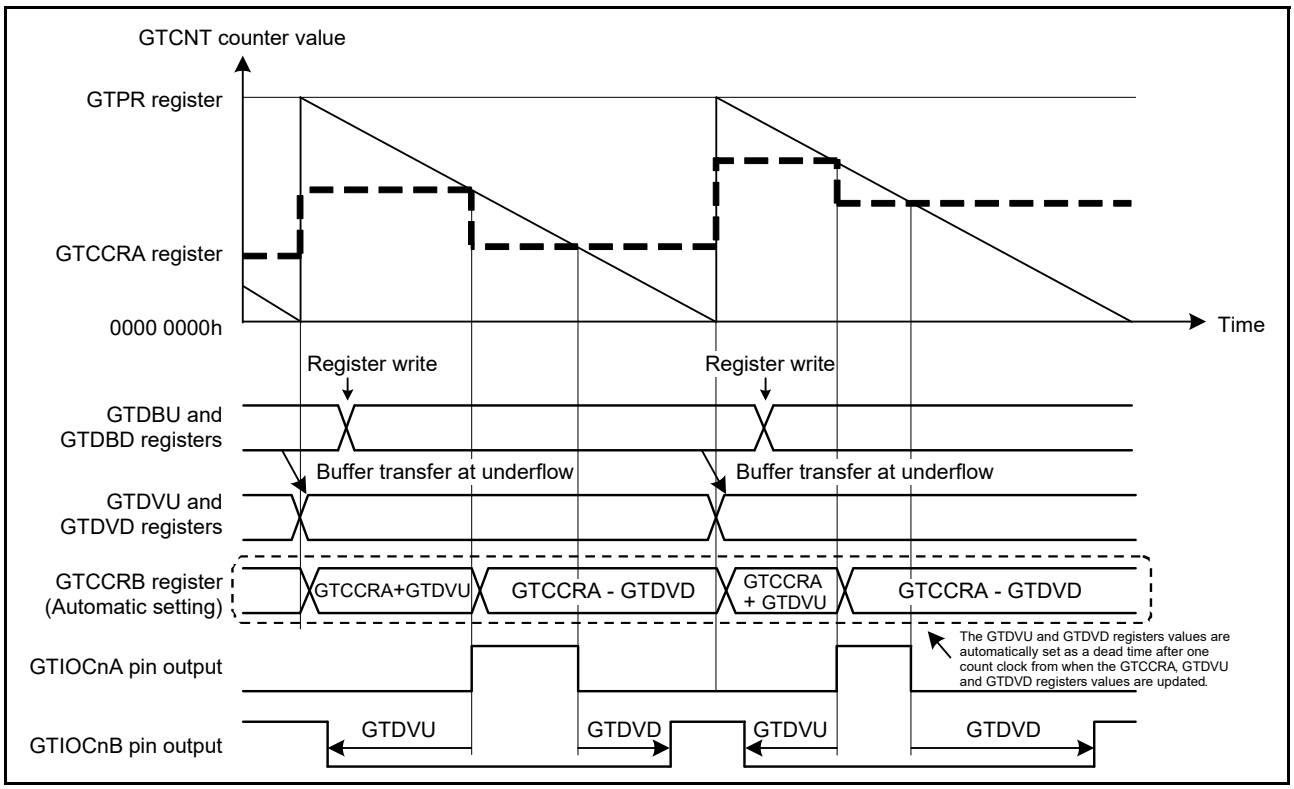


Figure 24.100 Example of Automatic Dead Time Setting Function Operation (Sawtooth-Wave One-Shot Pulse Mode, Down-Counting, GTDVU and GTDVD Registers Set to Buffer Operation, Active Level: High) (n = 0 to 7)

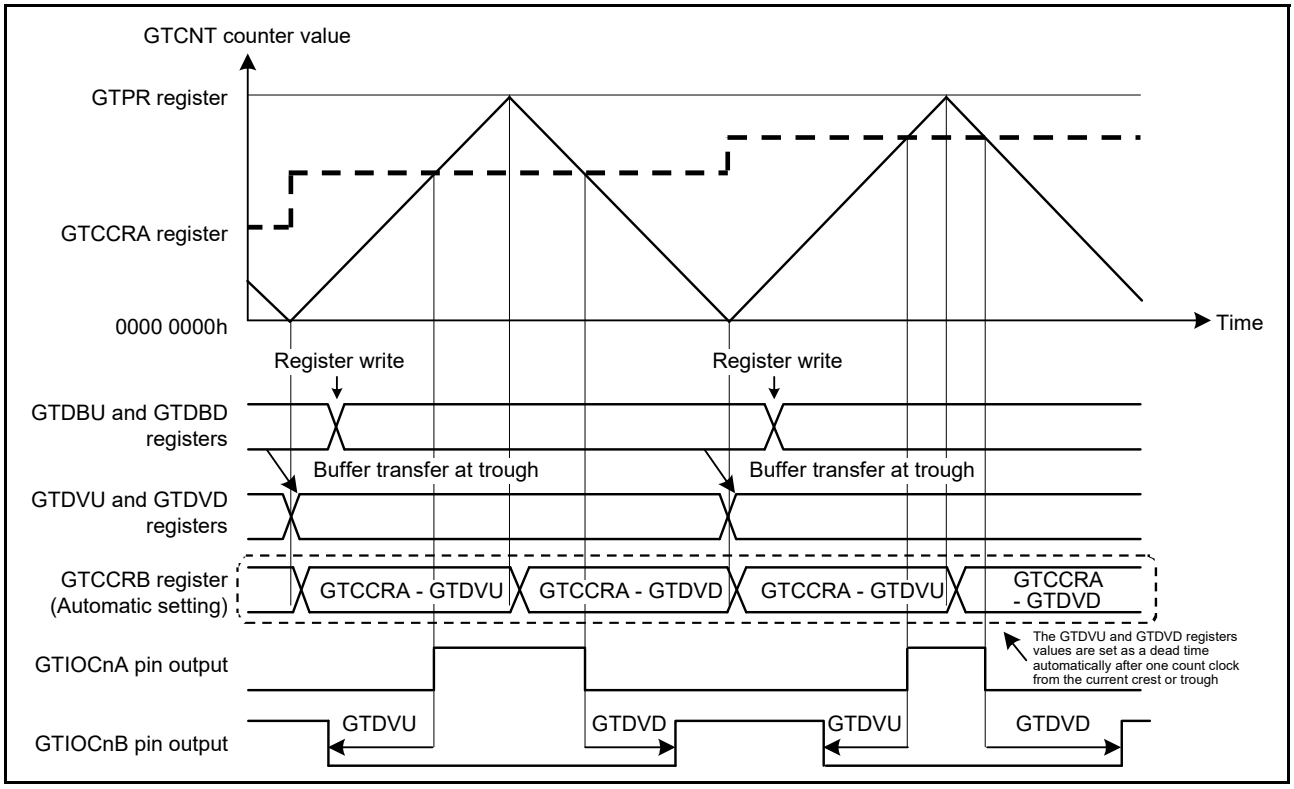


Figure 24.101 Example of Automatic Compare-Match Value Setting Function with Dead Time (Triangle-Wave PWM Mode 1, GTDVU and GTDVD Registers Set to Buffer Operation, Active Level: High) (n = 0 to 7)

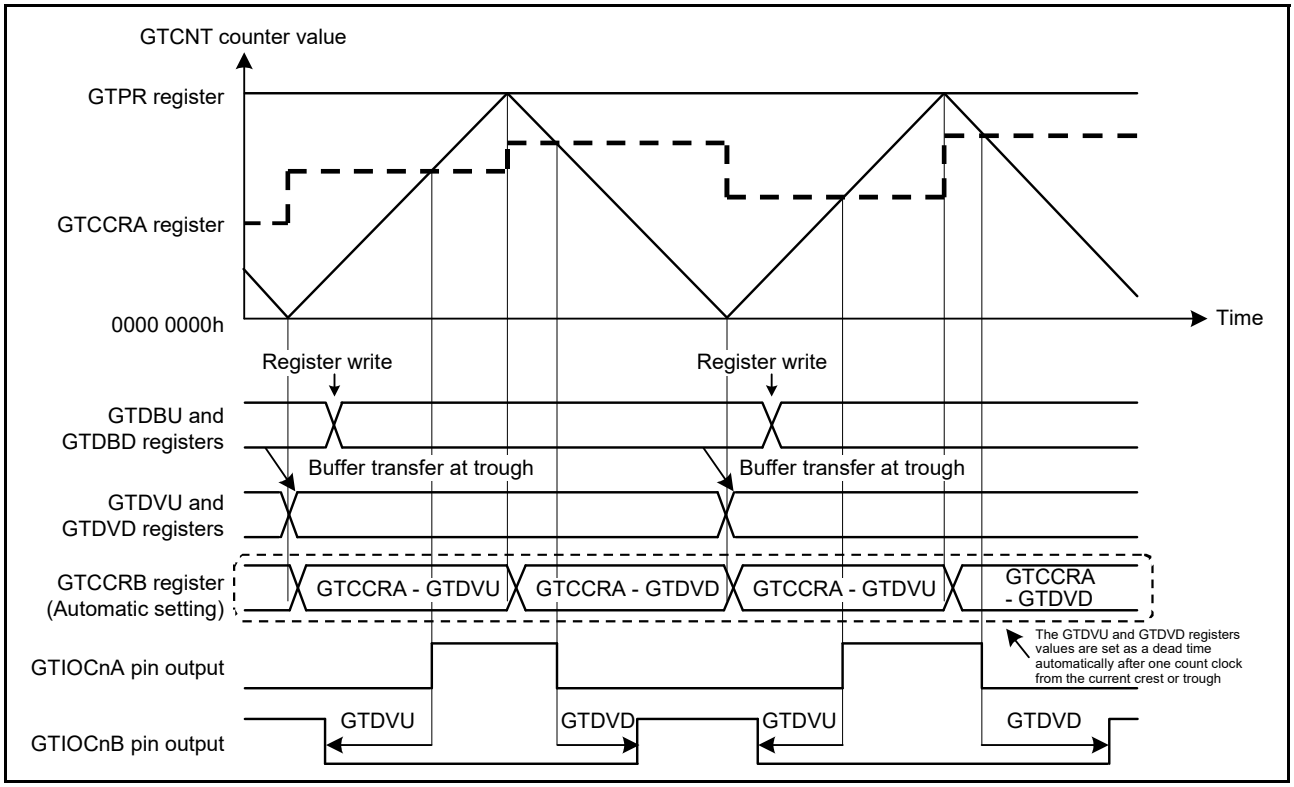
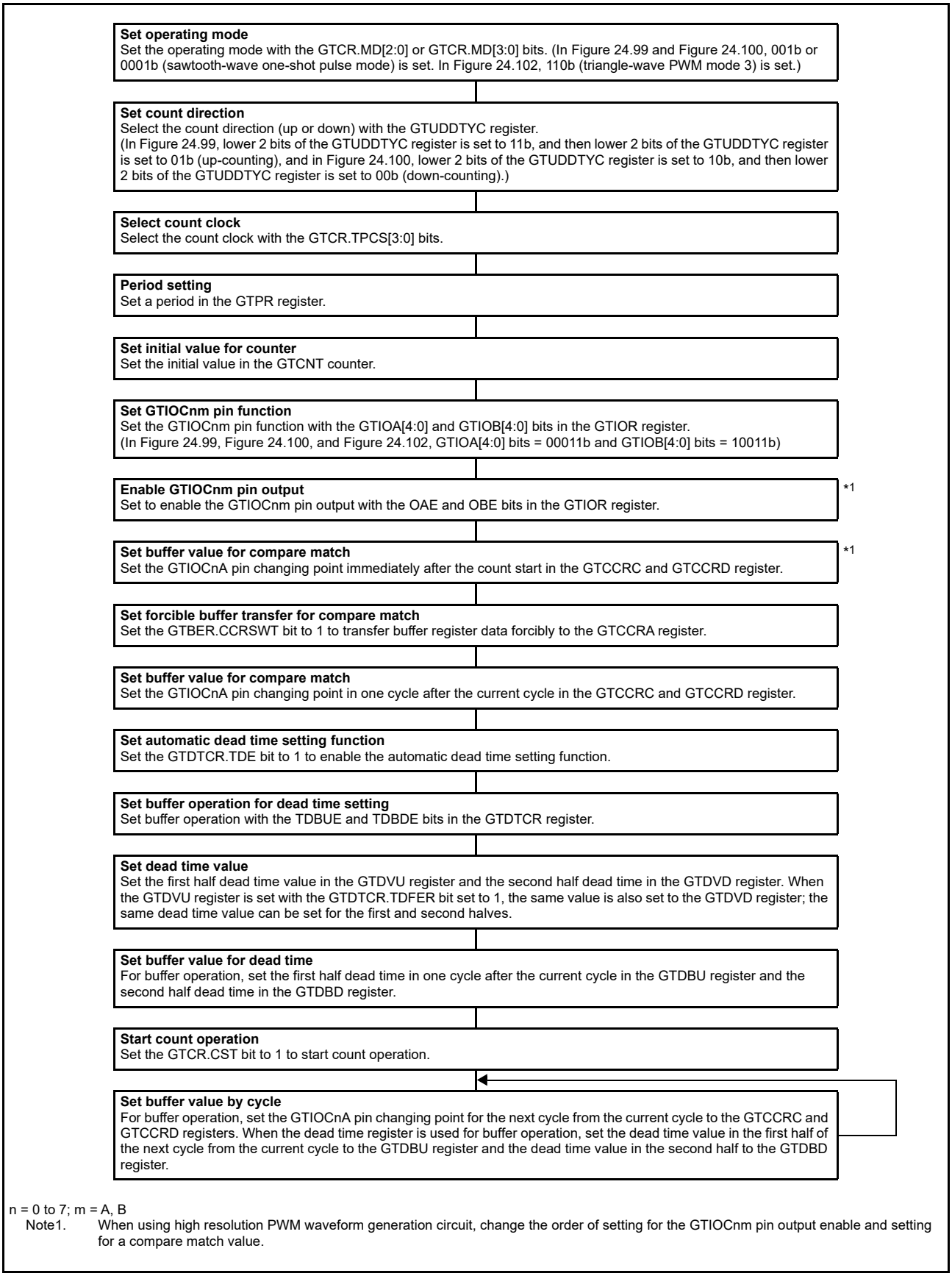


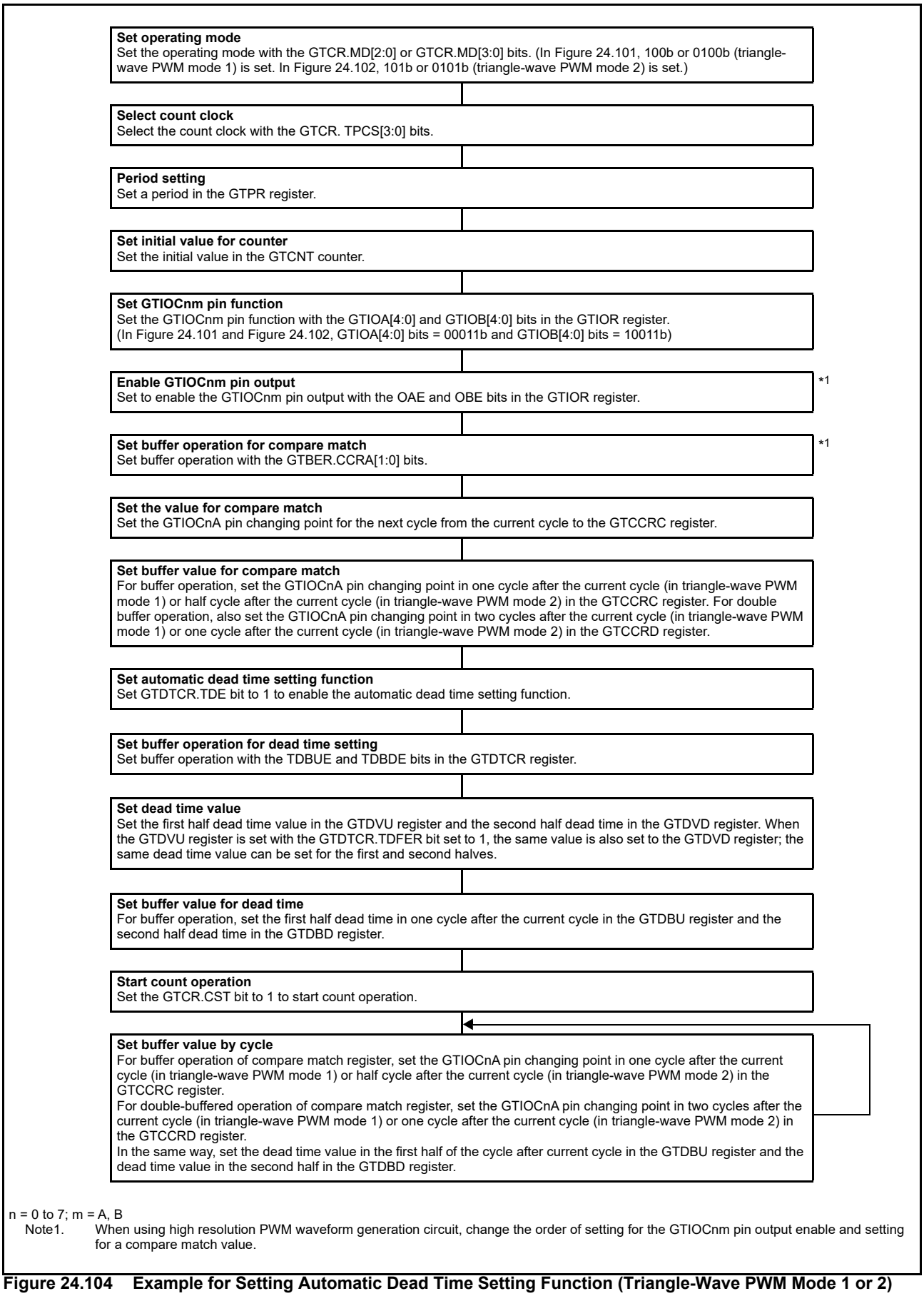
Figure 24.102 Example of Automatic Compare-Match Value Setting Function with Dead Time (Triangle-Wave PWM Mode 2 or 3, GTDVU and GTDVD Registers Set to Buffer Operation, Active Level: High) (n = 0 to 7)



n = 0 to 7; m = A, B

Note1. When using high resolution PWM waveform generation circuit, change the order of setting for the GTIOCnm pin output enable and setting for a compare match value.

Figure 24.103 Example for Setting Automatic Dead Time Setting Function (Sawtooth-Wave One-Shot Pulse Mode, Triangle-Wave PWM Mode 3)



n = 0 to 7; m = A, B

Note1. When using high resolution PWM waveform generation circuit, change the order of setting for the GTIOCnm pin output enable and setting for a compare match value.

Figure 24.104 Example for Setting Automatic Dead Time Setting Function (Triangle-Wave PWM Mode 1 or 2)

24.3.5 Count Direction Changing Function

The count direction of the GTCNT counter can be changed by modifying the GTUDDTYC.UD bit.
In sawtooth-wave mode, if the UD bit is modified during count operation, the count direction is changed at an overflow (when modified during up-counting) or an underflow (when modified during down-counting). If the UD bit is modified while count operation is stopped and the GTUDDTYC.UDF bit is 0, the UD bit modification is not reflected at the start of counting and the count direction is changed at an overflow or an underflow. If the UDF bit is set to 1 while count operation is stopped, the UD bit value at that time is reflected at the start of counting.
In triangle-wave mode, changing the value of the GTUDDTYC.UD bit during counting does not switch the direction of counting. Likewise, changing the value of the GTUDDTYC.UD bit while the GTUDDTYC.UDF bit is 0 and counting is stopped does not actually update the value of the bit. If 1 is written to the GTUDDTYC.UDF bit while counting is stopped, the value of the GTUDDTYC.UD bit at that time is reflected from the time counting is started.
When counting direction is switched during count operation in sawtooth-wave mode, the value of the GTPR register after start of up-counting is reflected to the count period in up-counting operation, and the value of the GTPR register after the start of down-counting is reflected to the count period in down-counting operation.
Figure 24.105 shows an example of count direction changing function operation.

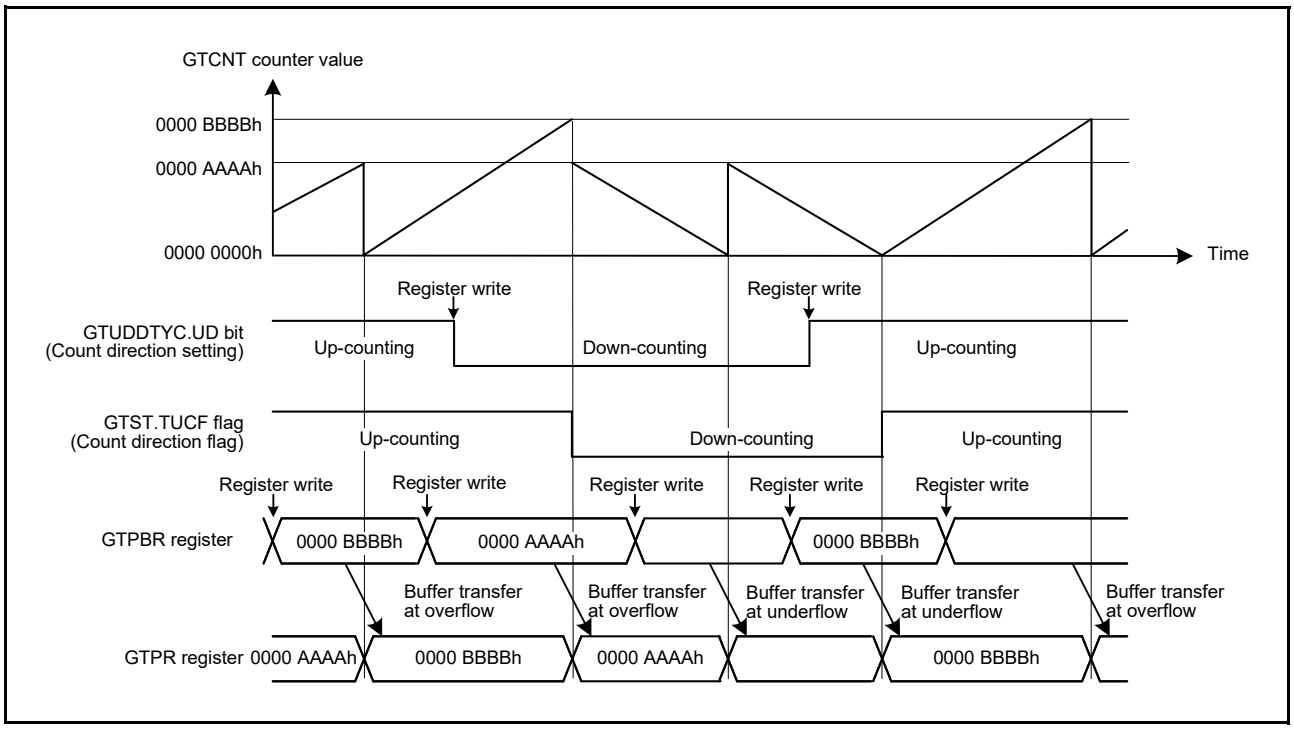


Figure 24.105 Example of Count Direction Changing Function Operation (during Buffer Operation)

24.3.6 Duty Cycle 0%/100% Output Function

Changing the value of the GTUDDTYC.OADTY[1:0] bits and GTUDDTYC.OBDTY[1:0] bits specifies the output duty setting on the GTIOCnA and GTIOCnB pins to 0% or 100%. (n = 0 to 7)

This function is invalid in sawtooth-wave PWM mode 2 or complementary PWM mode.

In sawtooth-wave mode, if the value of the OADTY[1:0] bits or OBDTY[1:0] bits is changed during the count operation, the output duty-cycle setting is reflected at an overflow (when changed during up-counting) or an underflow (when changed during down-counting). When the GTUDDTYC.OADTYF bit or GTUDDTYC.OBDTYF bit is 0 while count operation is stopped and the value of the OADTY[1:0] bits or OBDTY[1:0] bits is changed, the output duty-cycle setting changed at the start of counting is not reflected, but the output duty-cycle setting changed at an overflow or underflow is reflected. When the OADTYF bit or OBDTYF bit is 1 while count operation is stopped and the value of the OADTY[1:0] bits or OBDTY[1:0] bits is changed, the value of the OADTY[1:0] bits or OBDTY[1:0] bits at that time is reflected at the start of counting.

In triangle-wave mode, if the value of the OADTY[1:0] bits or OBDTY[1:0] bits is changed during the count operation, the output duty-cycle setting changed at an underflow is reflected.

When the OADTYF bit or OBDTYF bit is 0 while count operation is stopped and the value of the OADTY[1:0] bits or OBDTY[1:0] bits is changed, the output duty-cycle setting changed at the start of counting is not reflected, but the output duty-cycle setting changed at an underflow is reflected. When the OADTYF bit or OBDTYF bit is 1 while count operation is stopped and the value of the OADTY[1:0] bits or OBDTY[1:0] bits is changed, the value of the OADTY[1:0] bits or OBDTY[1:0] bits at that time is reflected at the start of counting.

During the operation under the settings of duty cycle for 0%/100%, compare match operation inside the GPTW continues to perform interrupt outputs and buffer operations.

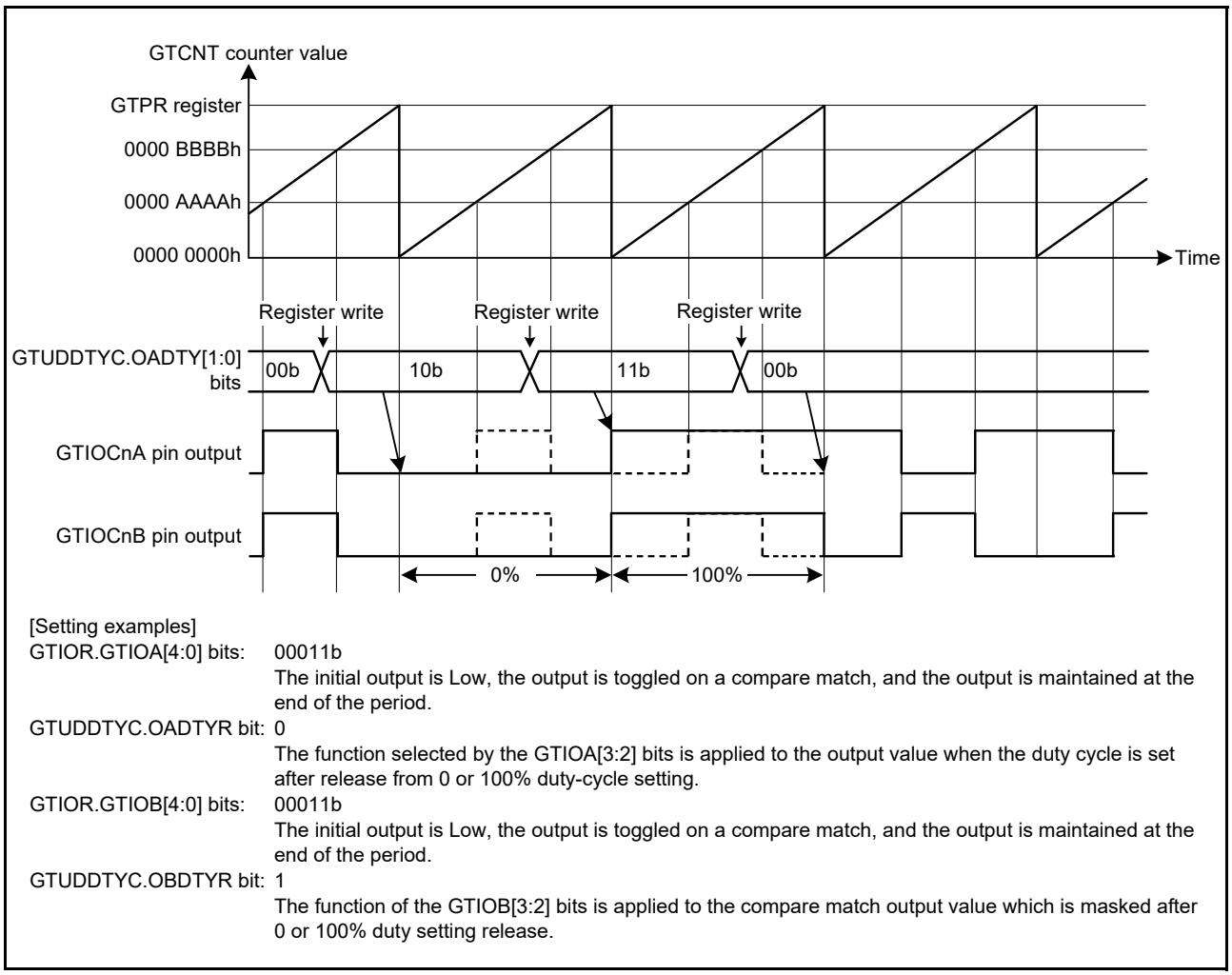
When the output duty setting is changed from 0% or 100% due to a compare match, the GTIOR.GTIOA[3:2] bits and the OADTYR bits determine the level output on the GTIOCnA pin and GTIOR.GTIOB[3:2] bits and OBDTYR bits determine the level output on the GTIOCnB pin at the end of the cycle. When the GTIOA[3:2], GTIOB[3:2] bits are set as 01b, output is driven Low at the end of the cycle, and when set as 10b, output is driven High at the end of the cycle. The output is retained at the end of the cycles when the GTIOA[3:2] or GTIOB[3:2] bits are 00b, and toggled at the end of the cycles when the GTIOA[3:2] or GTIOB[3:2] bits are 11b. A value to be output is selectable in the OADTYR or OBDTYR bit when the corresponding GTIOA[3:2] or GTIOB[3:2] bits are either 00b or 11b. Table 24.22 lists the output values at the end of the cycle when the output setting is changed from duty 0% or 100% to compare match.

Table 24.22 Output Value after Release of Duty 0%/100%

GTIOR.GTIOm[3:2]	Value at Compare Match Output at the End of the Cycle in the Case of Masking by a Duty Cycle 0% or 100%	The GTUDDTYC.OmDTYR Bit at Duty 0% Setting		The GTUDDTYC.OmDTYR Bit at Duty 100% Setting	
		0	1	0	1
00b (Output retained at the end of the cycle)	0	0	0	1	0
	1	0	1	1	1
01b (Low output at the end of the cycle)	—	0	0	0	0
10b (High output at the end of the cycle)	—	1	1	1	1
11b (Toggle output at the end of the cycle)	0	1	1	0	1
	1	1	0	0	0

m = A, B

Figure 24.106 shows an example of operation of the output of the duty cycle 0% or 100% (n = 0 to 7).



[Setting examples]

GTIOR.GTIOA[4:0] bits: 00011b
The initial output is Low, the output is toggled on a compare match, and the output is maintained at the end of the period.

GTUDDTYC.OADTYR bit: 0
The function selected by the GTIOA[3:2] bits is applied to the output value when the duty cycle is set after release from 0 or 100% duty-cycle setting.

GTIOR.GTIOB[4:0] bits: 00011b
The initial output is Low, the output is toggled on a compare match, and the output is maintained at the end of the period.

GTUDDTYC.OBDTYR bit: 1
The function of the GTIOB[3:2] bits is applied to the compare match output value which is masked after 0 or 100% duty setting release.

Figure 24.106 Example of Operation of Output of Duty Cycle 0% or 100% (n = 0 to 7)

24.3.7 Hardware Count Start/Count Stop and Clear Operation

The GTCNT counter can be started, stopped, or cleared by hardware sources in this MCU. There are 3 types of hardware sources, including external trigger input, ELC event input, and GTIOCnm pin input (n = 0 to 7; m = A, B)

24.3.7.1 Hardware Start Operation

The GTCNT counter can be started by a hardware source. Select a hardware source to start counting using the GTSSR register, and enable to start counting.
Figure 24.107 shows an example of count start operation by a hardware source. Figure 24.108 shows the setting example.

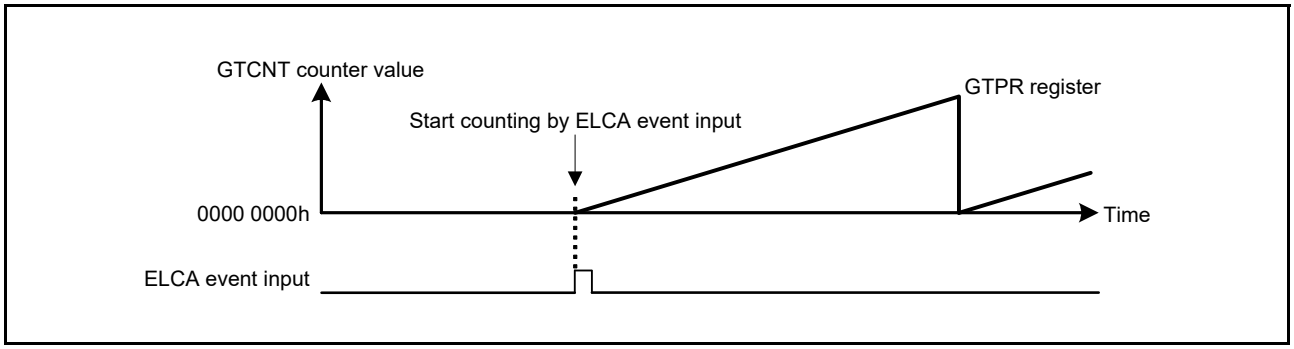


Figure 24.107 Example of Count Start Operation by Hardware Source (At starting by ELCA event)

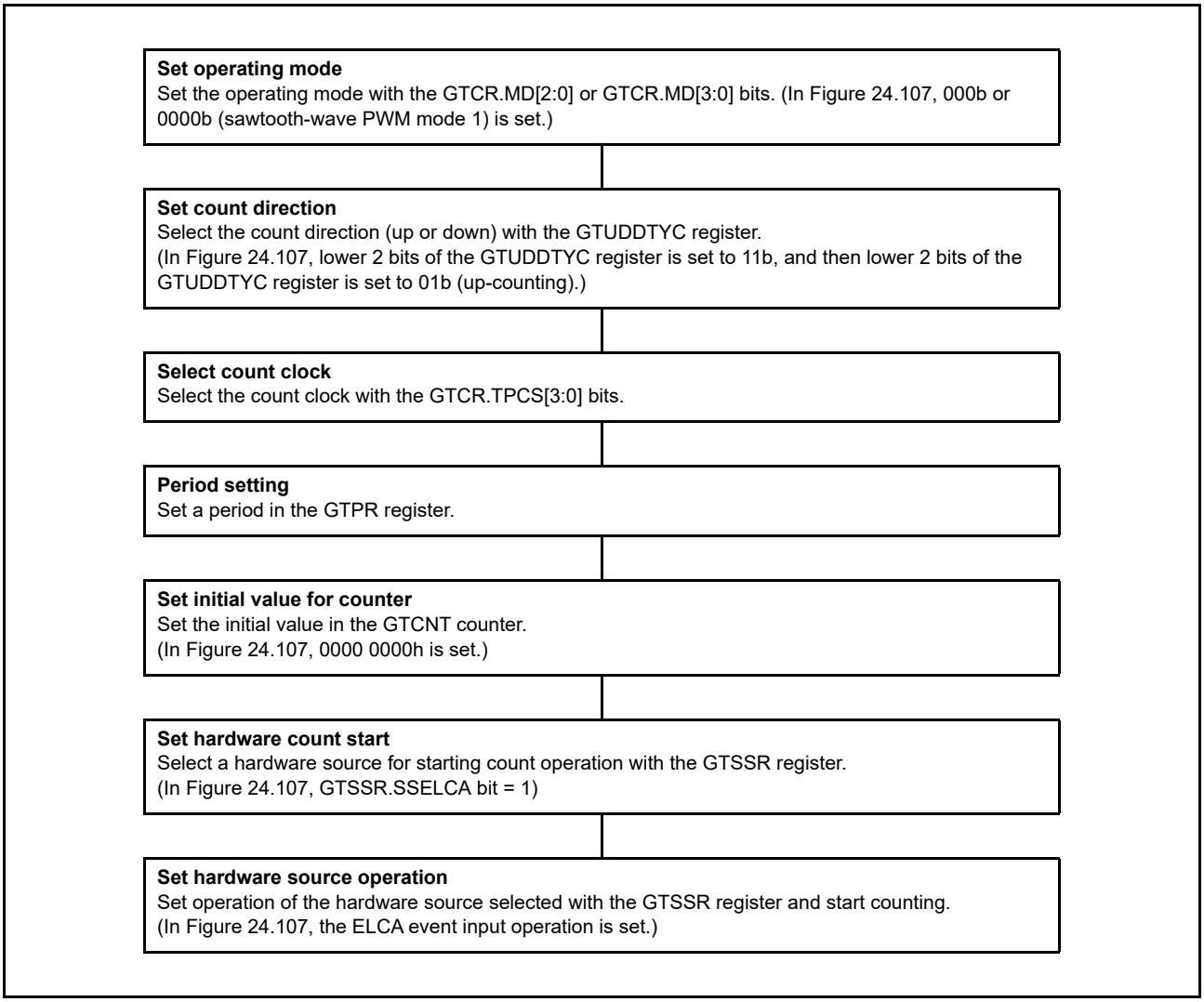


Figure 24.108 Example for Setting Count Start Operation by Hardware Source

Figure 24.109 shows an example of timing of operations to start counting in response to a rising edge of the input on the GTETRGA pin.

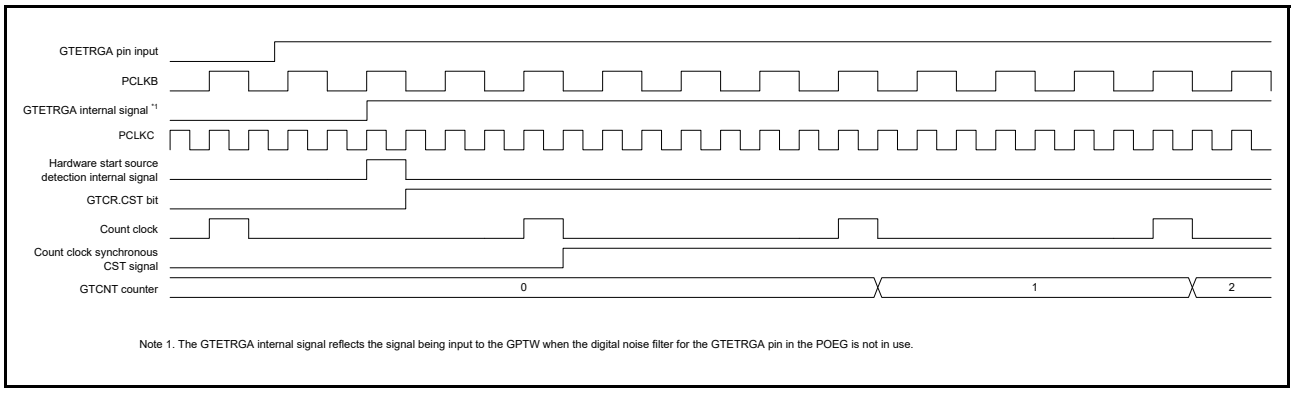


Figure 24.109 Example of Timing of Operations to Start Counting in Response to a Rising Edge of the Input on the GTETRGA Pin

Figure 24.110 shows an example of timing of operations to start counting in response to a rising edge of the input on the GTIOCnA pin.

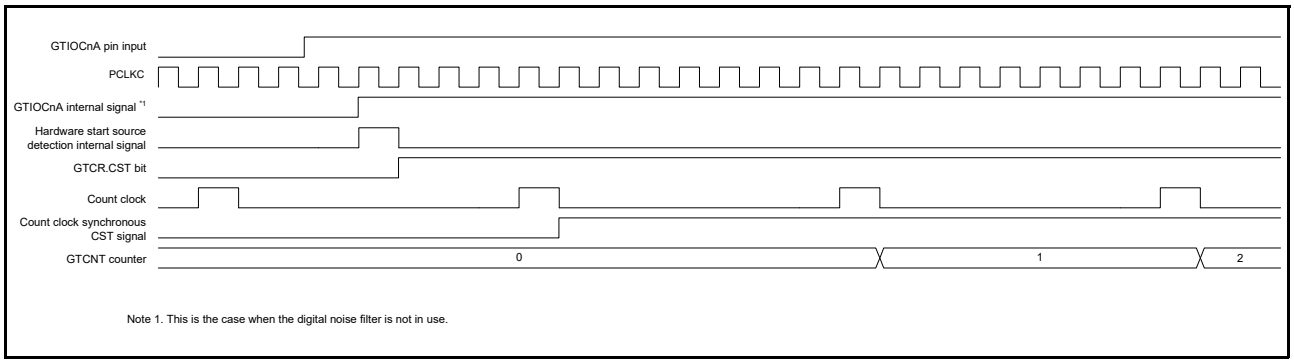


Figure 24.110 Example of Timing of Operations to Start Counting in Response to a Rising Edge of the Input on the GTIOCnA Pin

Figure 24.111 shows an example of timing of operations to start counting in response to event input from the ELCA. This is an example of operations to start counting by the GPTW1.GTCNT counter in response to a signal. An event signal is output to the ELC after matches in comparison with the GPTW0.GTCCRA register. This is selected as a trigger for output to the GPTW by the ELC as event source A. The GPTW0 compare match A signal is synchronized with PCLKC. The ELC receives the signal in synchronization with PCLKB, and outputs the GPTW event source A signal after 1 cycle of PCLKB has elapsed.

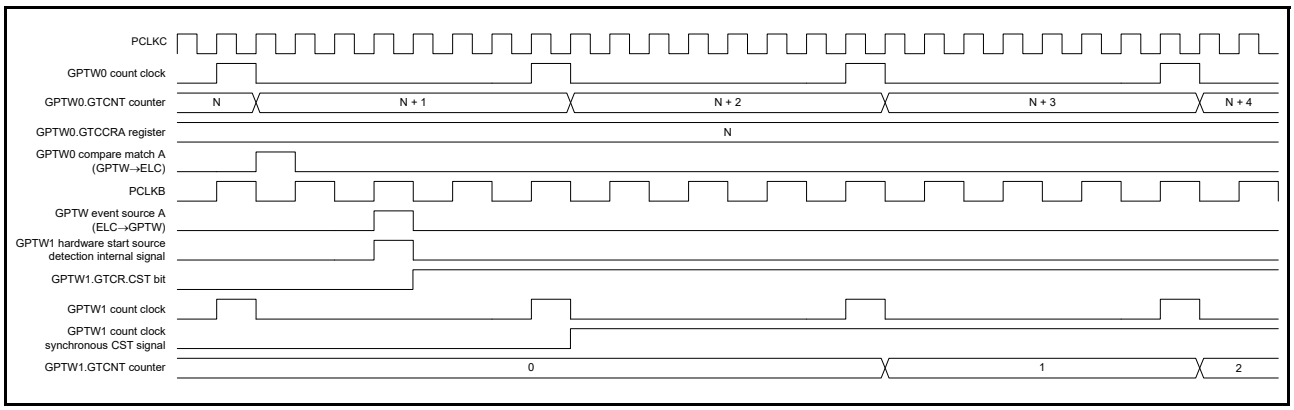


Figure 24.111 Example of Timing of Operations to Start Counting in Response to Event Input from the ELCA

24.3.7.2 Hardware Stop Operation

The GTCNT counter can be stopped by a hardware source. Select a hardware source to stop counting with the GTPSR register, and enable to stop counting.

Figure 24.112 shows an example of count stop operation by a hardware source. Figure 24.113 shows the setting example. In this example, the count operation is stopped by the ELCA event input, and is restarted by the ELCB event input.

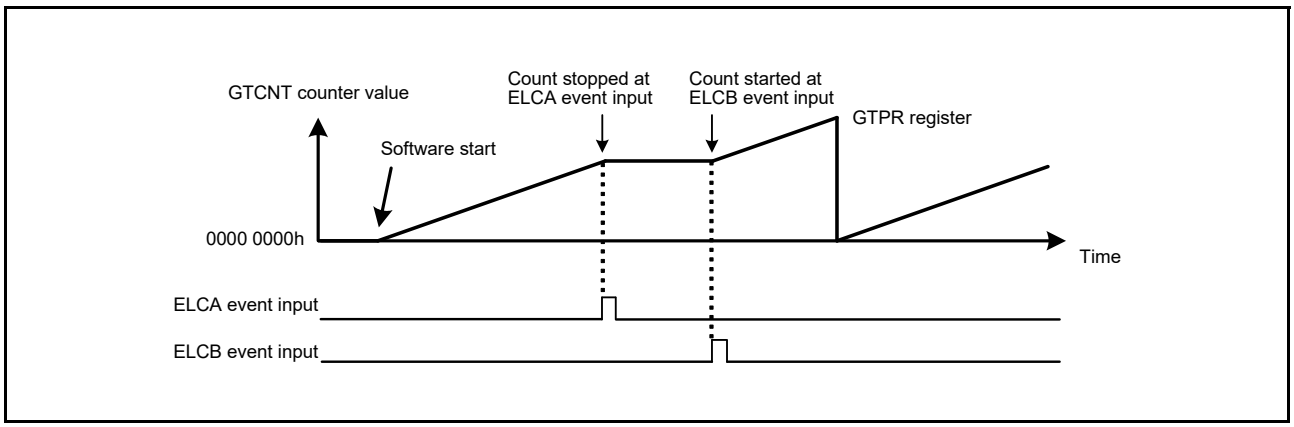


Figure 24.112 Example of Count Stop Operation by Hardware Source (Started by Software, Stopped at ELCA event input, and Restarted at ELCB event input)

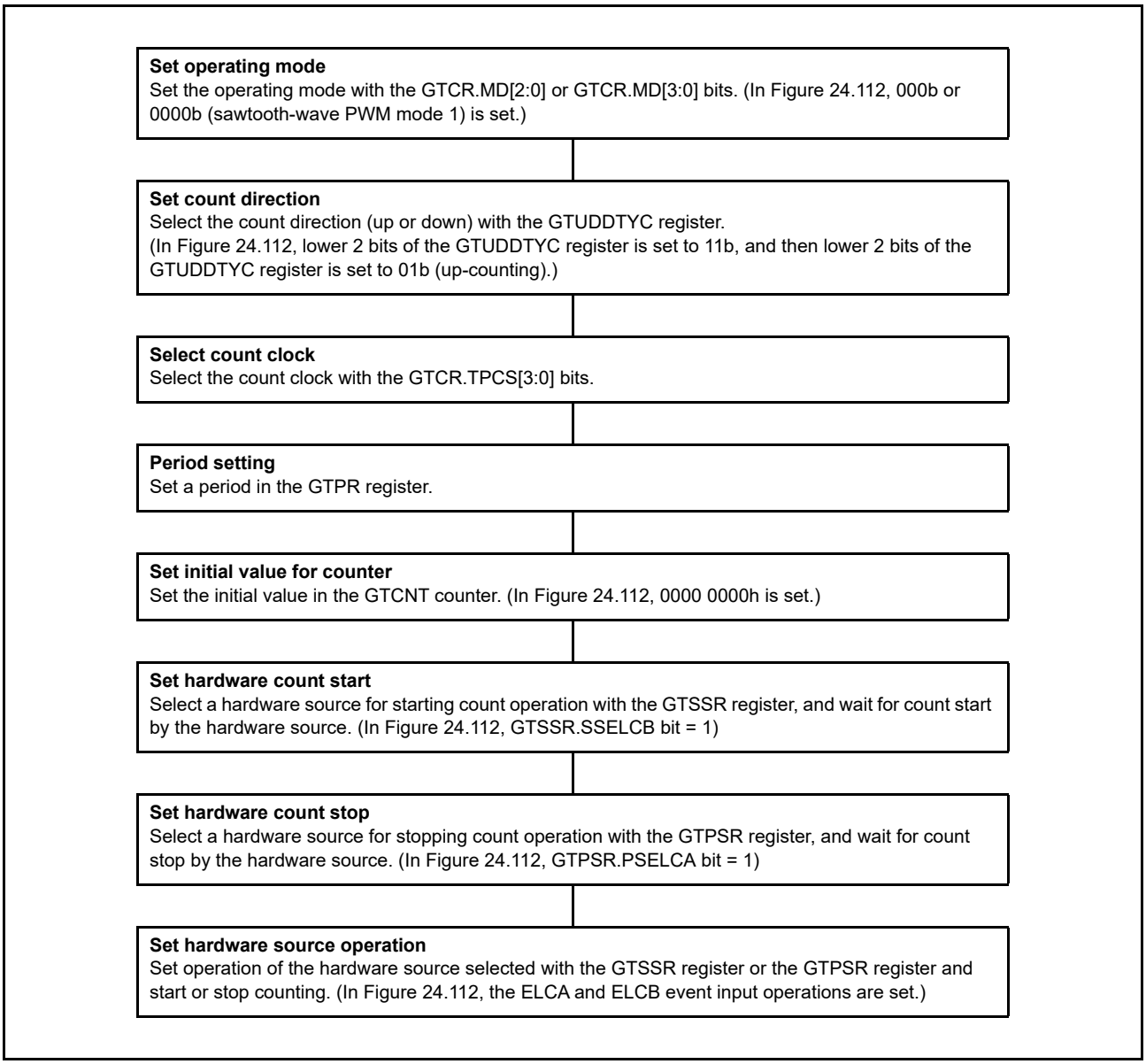


Figure 24.113 Example for Setting Count Stop Operation by Hardware Source

Figure 24.114 shows an example of count start/stop operation by a hardware source. Figure 24.115 shows the setting example. In this example, the counter operates during the high-level periods of the external trigger input GTETRGA pin.

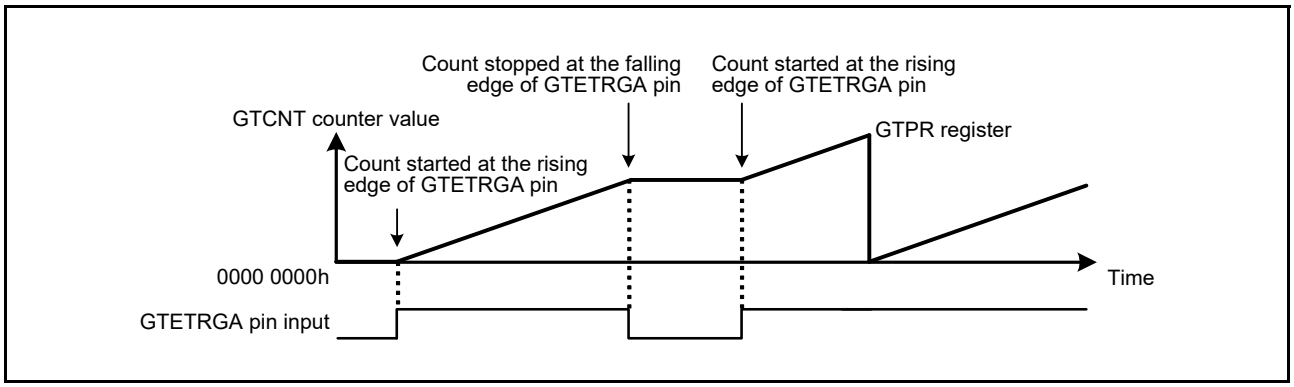


Figure 24.114 Example of Count Start/Stop Operation by Hardware Source (Started at Rising Edge of GTETRGA Pin Input, Stopped at Falling Edge of GTETRGA Pin Input)

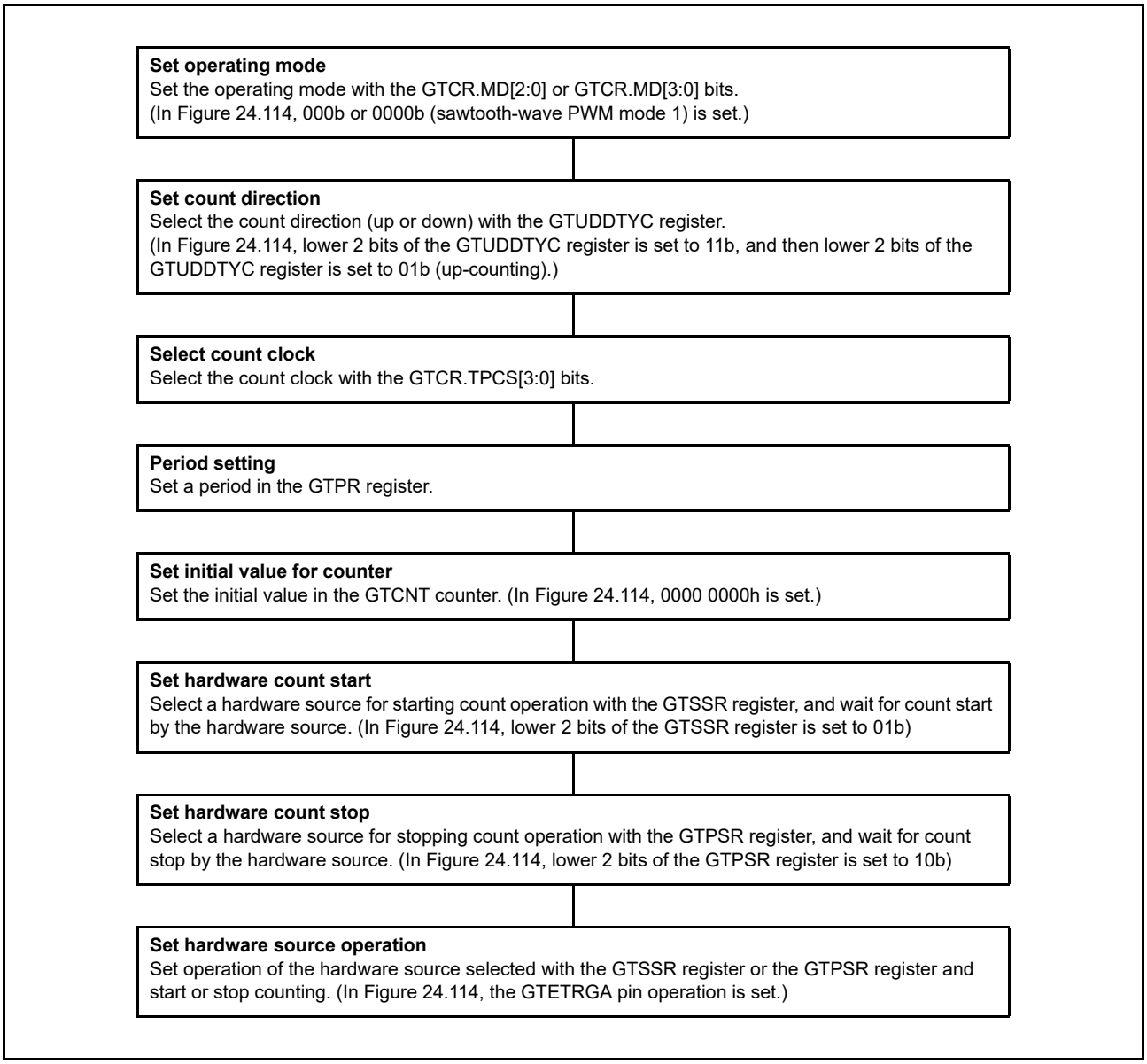


Figure 24.115 Example for Setting Count Start/Stop Operation by Hardware Source

Figure 24.116 shows an example of timing of operations to stop counting in response to a rising edge of the input on the GTETRGA pin.

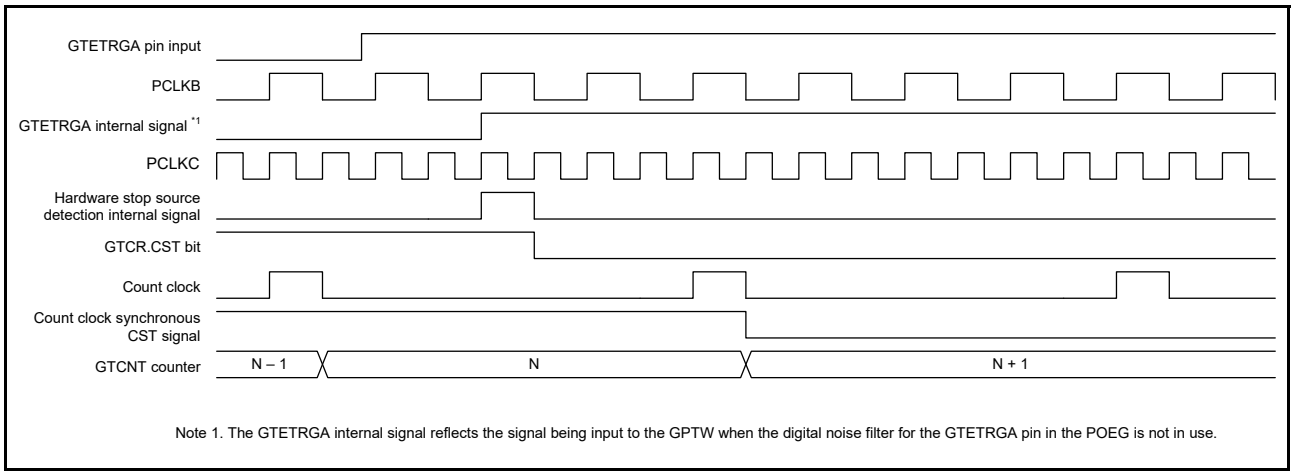


Figure 24.116 Example of Timing of Operations to Stop Counting in Response to a Rising Edge of the Input on the GTETRGA Pin

Figure 24.117 shows an example of timing of operations to stop counting in response to a rising edge of the input on the GTIOCnA pin.

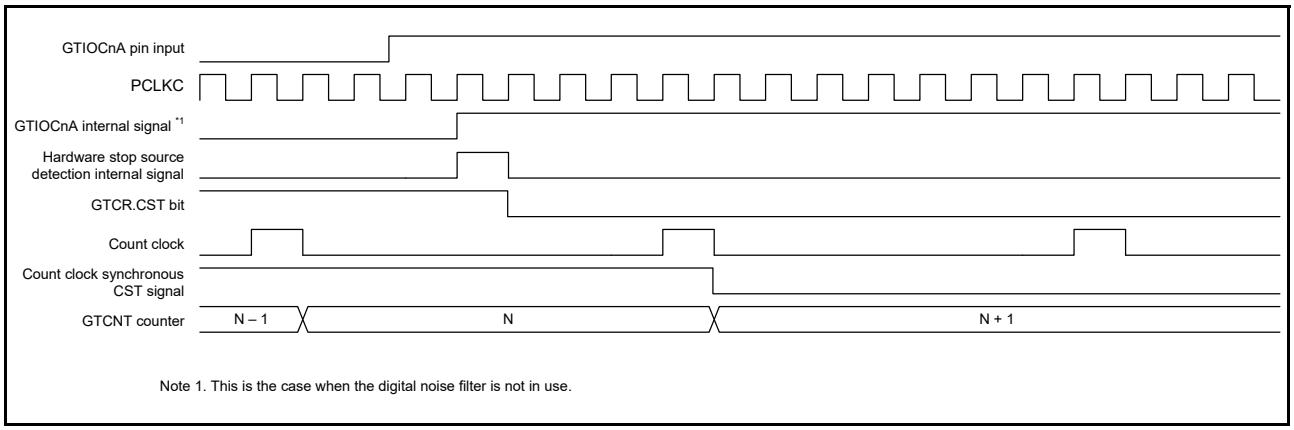


Figure 24.117 Example of Timing of Operations to Stop Counting in Response to a Rising Edge of the Input on the GTIOCnA Pin

Figure 24.118 shows an example of timing of operations to stop counting in response to event input from the ELCA. This is an example of operations to stop counting by the GPTW1.GTCNT counter in response to a signal. An event signal is output to the ELC after matches in comparison with the GPTW0.GTCCRA register. This is selected as a trigger for output to the GPTW by the ELC as event source A. The GPTW0 compare match A signal is synchronized with PCLKC. The ELC receives the signal in synchronization with PCLKB, and outputs the GPTW event source A signal after 1 cycle of PCLKB has elapsed.

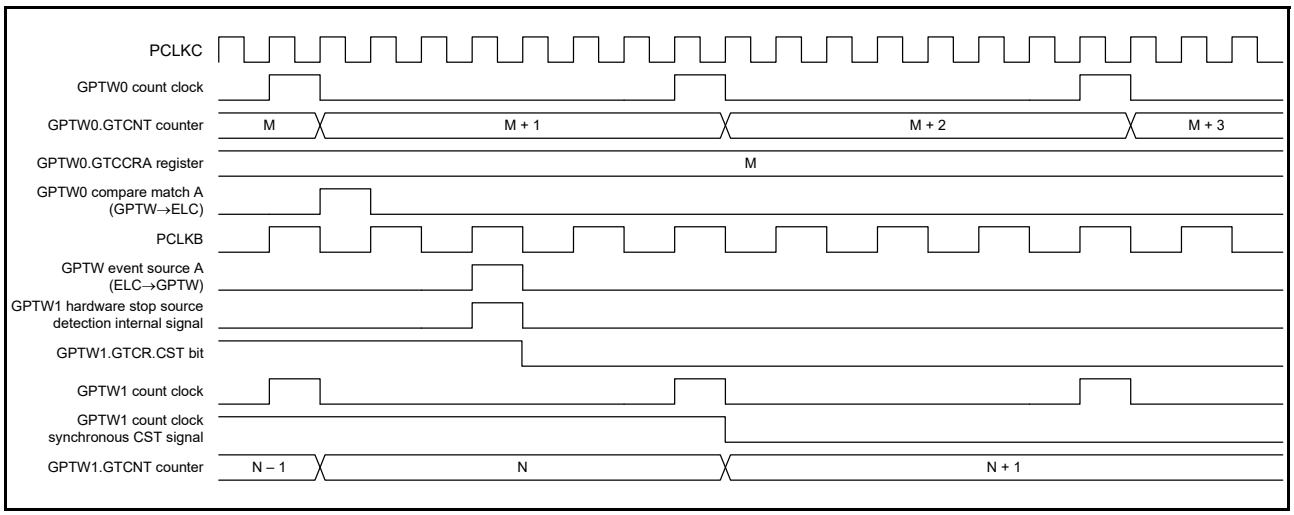


Figure 24.118 Example of Timing of Operations to Stop Counting in Response to Event Input from the ELCA

24.3.7.3 Hardware Clear Operation

The GTCNT counter can be cleared by a hardware source. Select a hardware source to clear the counter with the GTCSR register, and enable clearing the counter.

Note that the GTCIV/GTCIU interrupt (overflow/underflow interrupt) is not generated when the GTCNT counter is cleared by a hardware source or by software.

Figure 24.119 and Figure 24.120 show examples of the GTCNT counter clearing operation by a hardware source. Figure 24.121 shows the setting example. In this example, the GTCNT counter is started at the ELCA event input, and is stopped/cleared at the ELCB event input.

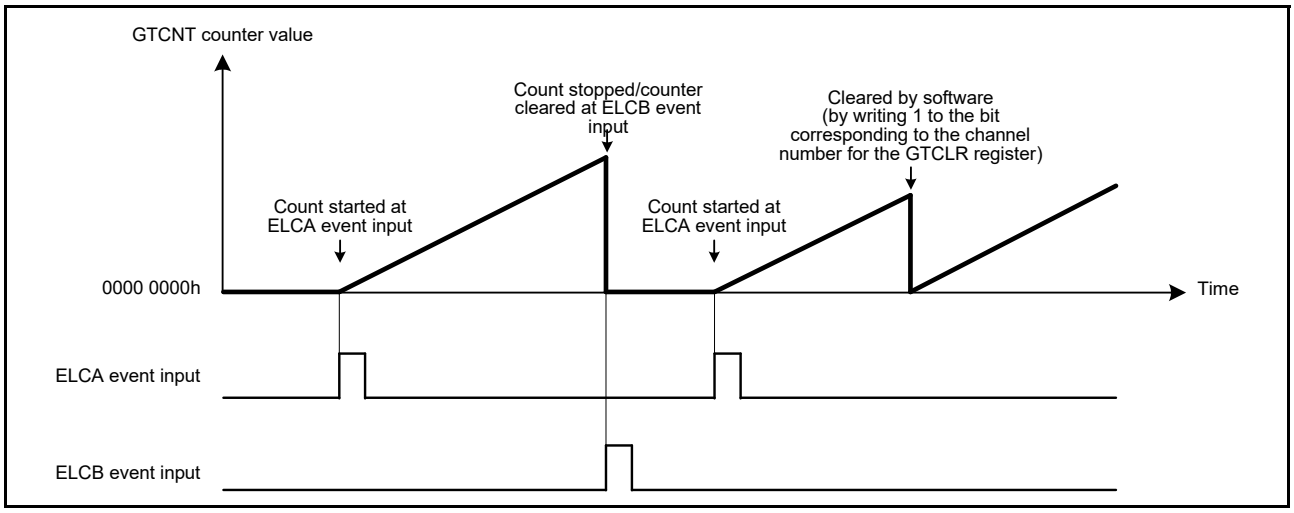


Figure 24.119 Examples of Counter Clearing Operation by Hardware Source (Sawtooth-Wave Up-Counting, Started at ELCA event input, and Count Stopped/Counter Cleared at ELCB event input)

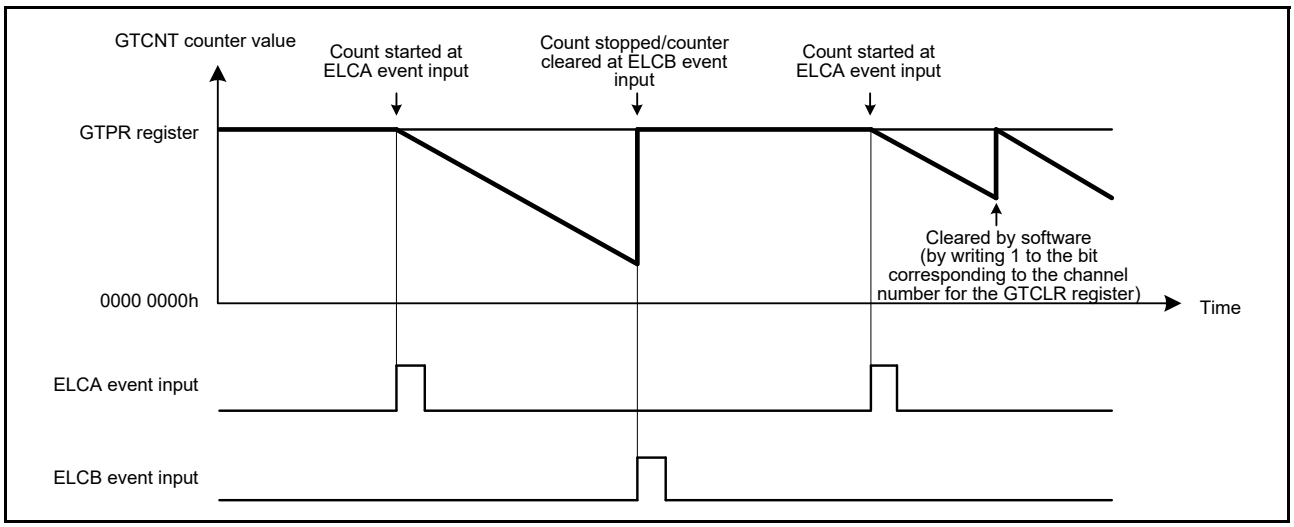


Figure 24.120 Examples of Counter Clearing Operation by Hardware Source (Sawtooth-Wave Down-Counting, Started at ELCA event input, and Count Stopped/Counter Cleared at ELCB event input)

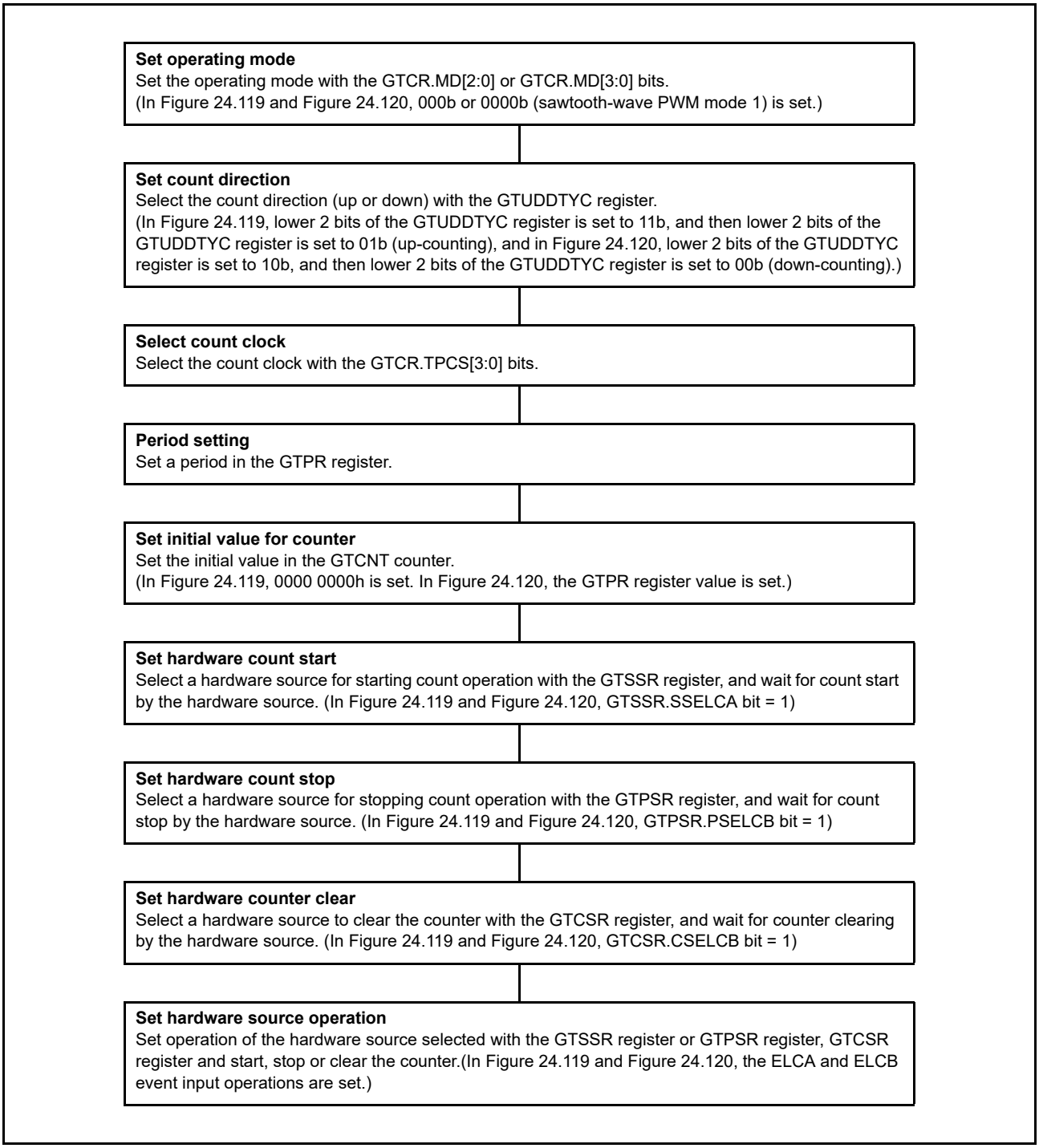


Figure 24.121 Example for Setting Counter Clearing Operation by Hardware Source

The GTCIV/GTCIU interrupt (overflow/underflow interrupt) does not occur when the counter is cleared by a hardware source. In the same way, the GTCIV/GTCIU interrupt does not occur when the counter is cleared by software.
Figure 24.122 shows the relationship between counter clearing by a hardware source and the GTCIV interrupt.

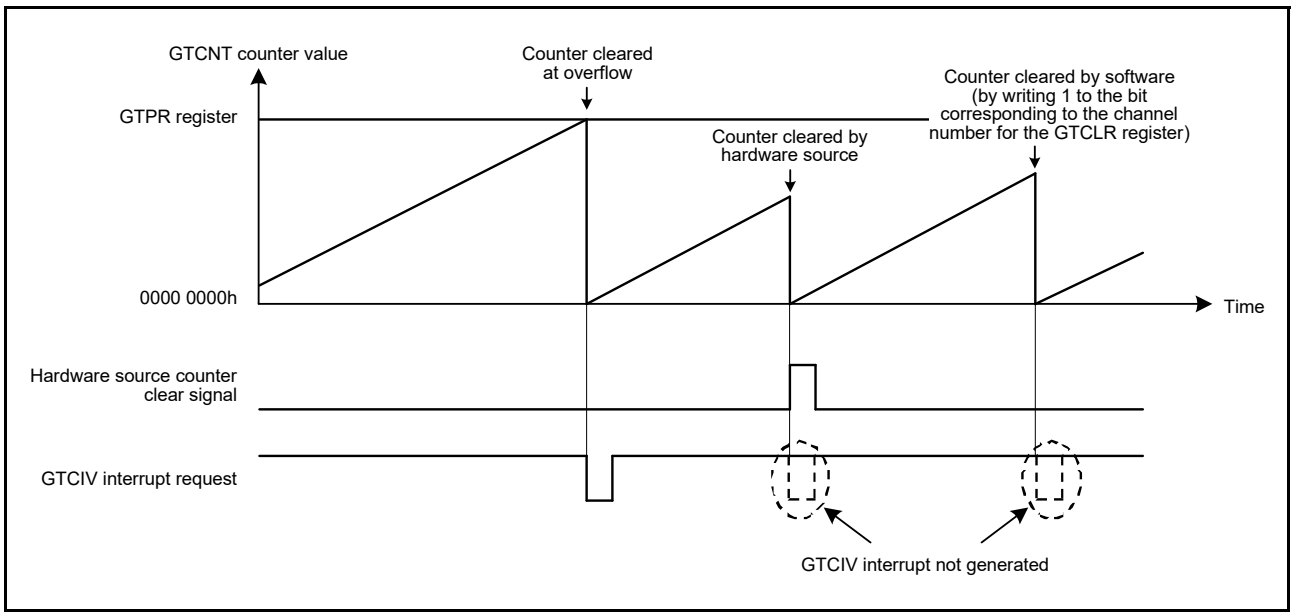


Figure 24.122 Relationship between Counter Clearing by Hardware Source and GTCIV Interrupt

Figure 24.123 shows an example of the timing of operations to clear the counter in response to a rising edge of the input on the GTETRGA pin when a clock signal produced by frequency-dividing the PCLKC signal is used as the counter clock for the GTCNT counter.
The GTCNT counter is cleared when counting is in progress after the GPTW has detected the internal clearing signal.

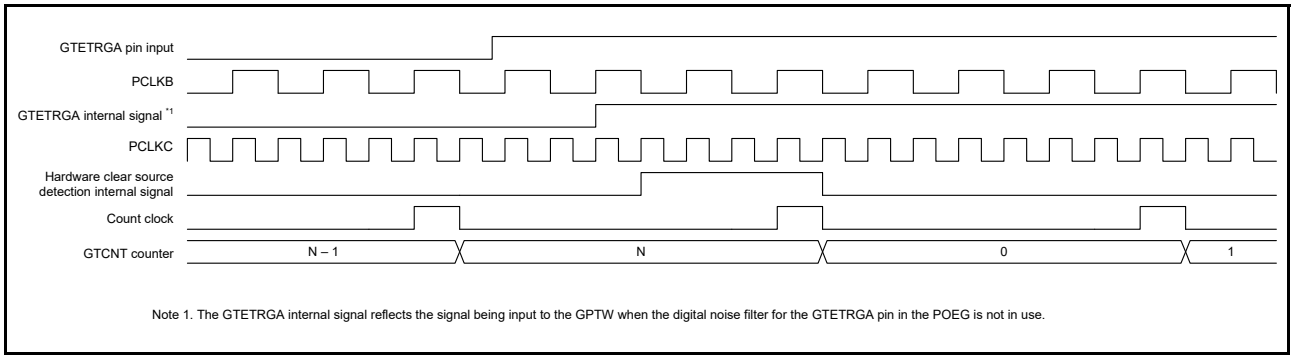


Figure 24.123 Example of the Timing of Operations for Counter Clearing in Response to a Rising Edge of the Input on the GTETRGA Pin (During the Counting of Cycles of Clock Signal Produced by Dividing the PCLKC Frequency)

Figure 24.124 shows an example of the timing of operations to clear the counter in response to a rising edge of the input on the GTIOCnA pin when a clock signal produced by frequency-dividing the PCLKC signal is used as the counter clock for the GTCNT counter.
The GTCNT counter is cleared when counting is in progress after the GPTW has detected the internal clearing signal.

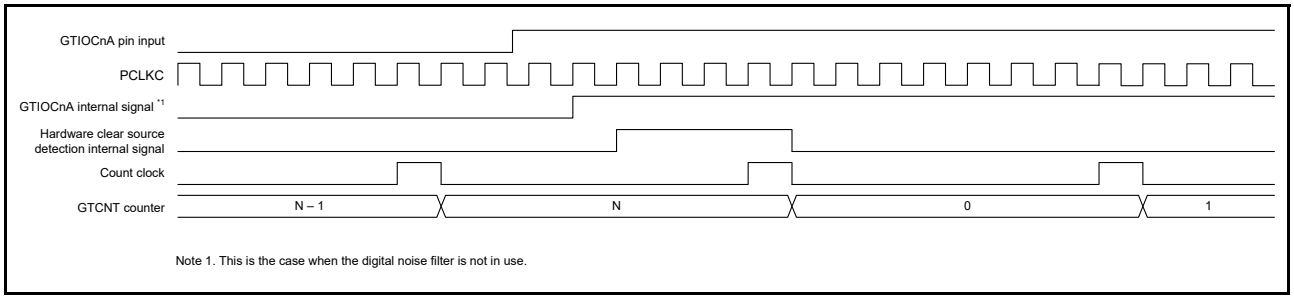


Figure 24.124 Example of the Timing of Operations for Counter Clearing in Response to a Rising Edge of the Input on the GTIOCnA Pin (During the Counting of Cycles of Clock Signal Produced by Dividing the PCLKC Frequency)

Figure 24.125 shows an example of the timing of operations to clear the counter in response to event input from the ELCA when a clock signal produced by frequency-dividing the PCLKC signal is used as the counter clock for the GTCNT counter.
This is an example of operations to clear the GPTW1.GTCNT counter in response to a signal. An event signal is output to the ELC after matches in comparison with the GPTW0.GTCCRA register. This is selected as a trigger for output to the GPTW by the ELC as event source A.
The GPTW0 compare match A signal is synchronized with PCLKC. The ELC receives the signal in synchronization with PCLKB, and outputs the GPTW event source A signal after 1 cycle of PCLKB has elapsed. The GTCNT counter is cleared when counting is in progress after the GPTW has detected the internal clearing signal.

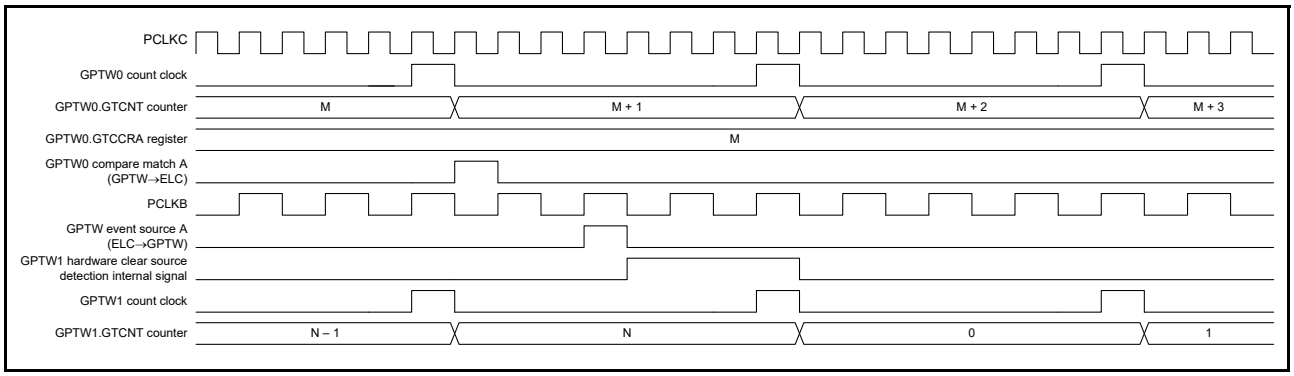


Figure 24.125 Example of the Timing of Operations for Counter Clearing in Response to Event Input from the ELCA (During the Counting of Cycles of Clock Signal Produced by Dividing the PCLKC Frequency)

Figure 24.126 shows an example of the timing of operations to clear the counter in response to a rising edge of the input on the GTETRGA pin input when counting is triggered by a hardware source.
The GTCNT counter is cleared in synchronization with PCLKC after the GPTW has detected the internal clearing signal.

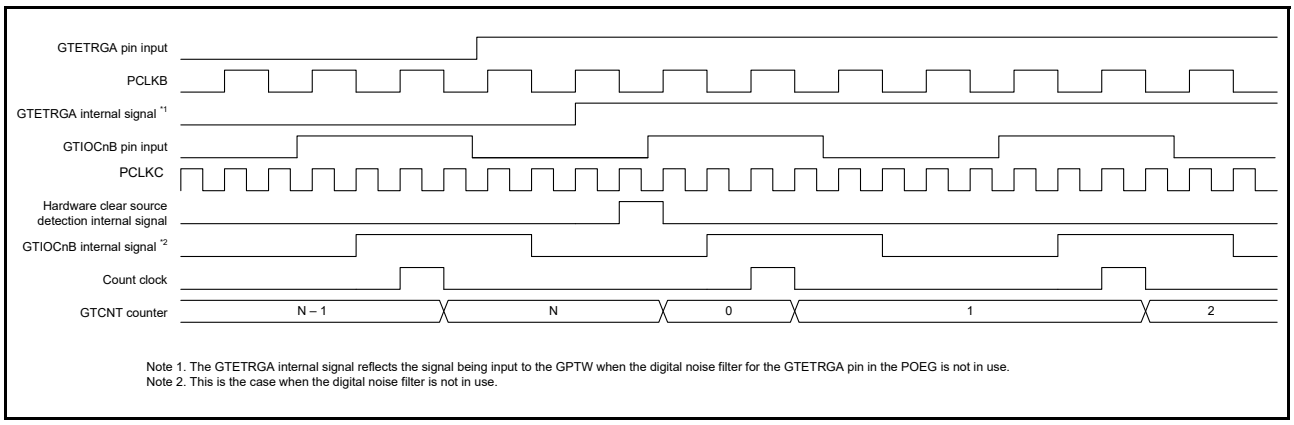


Figure 24.126 Example of the Timing of Operations for Counter Clearing in Response to a Rising Edge of the Input on the GTETRGA Pin (During Counting Triggered by Hardware Source)

Figure 24.127 shows an example of the timing of operations to clear the counter in response to a rising edge of the input on the GTIOCnA pin input when counting is triggered by a hardware source.
The GTCNT counter is cleared in synchronization with PCLKC after the GPTW has detected the internal clearing signal.

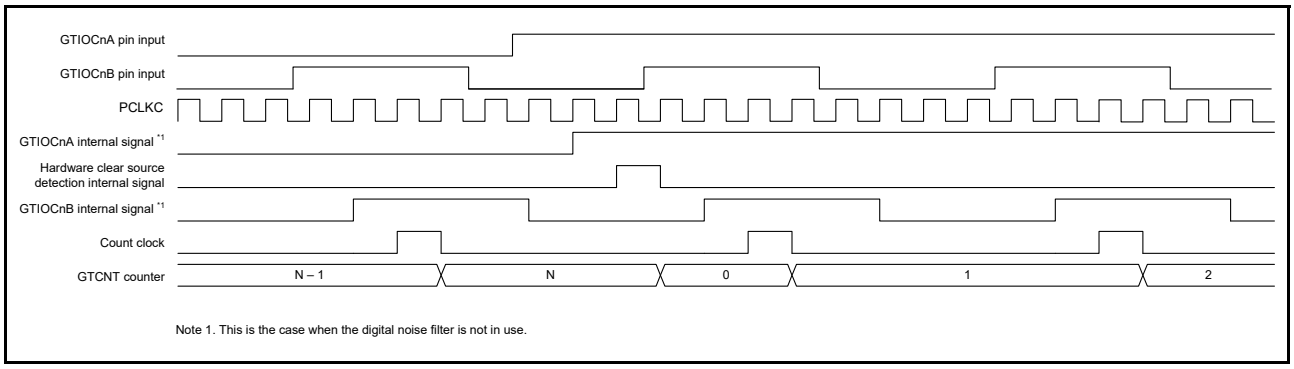


Figure 24.127 Example of the Timing of Operations for Counter Clearing in Response to a Rising Edge of the Input on the GTIOCnA Pin (During Counting Triggered by Hardware Source)

Figure 24.128 shows an example of the timing of operations for clearing the counter in response to the input of an event signal from the ELCA when counting is triggered by a hardware source.
This is an example of operations to clear the GPTW1.GTCNT counter in response to a signal. An event signal is output to the ELC after matches in comparison with the GPTW0.GTCCRA register. This is selected as a trigger for output to the GPTW by the ELC as event source A.
The GPTW0 compare match A signal is synchronized with PCLKC. The ELC receives the signal in synchronization with PCLKB, and outputs the GPTW event source A signal after 1 cycle of PCLKB has elapsed. The GTCNT counter is cleared in synchronization with PCLKC after the GPTW has detected the internal clearing signal.

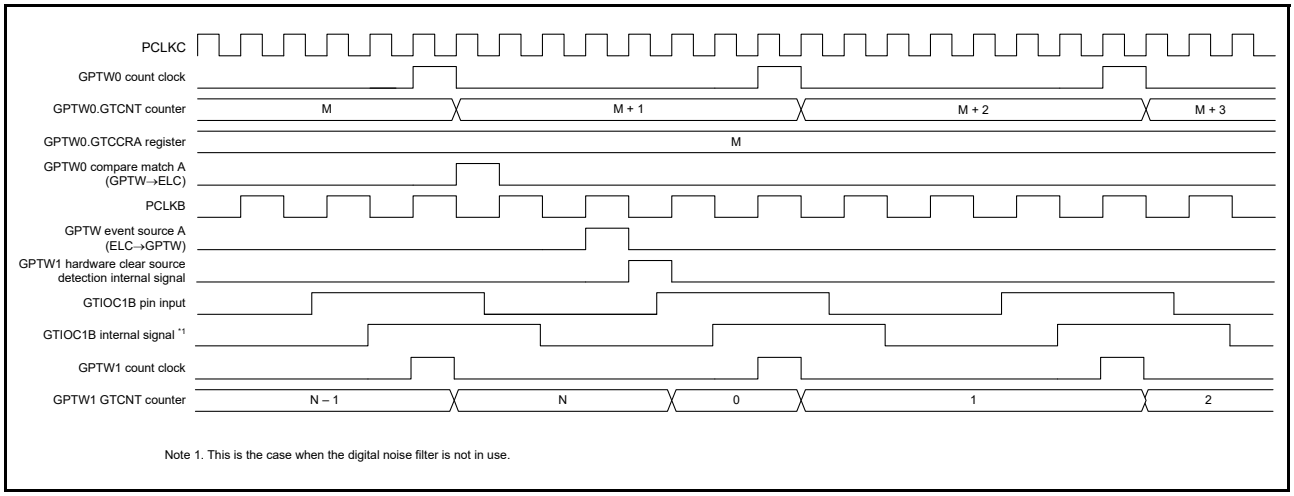


Figure 24.128 Example of the Timing of Operations for Counter Clearing in Response to Event Input from the ELCA (During Counting Triggered by Hardware Source)

24.3.8 Synchronous Operation

Synchronous operation on channels (synchronous start/stop/clear operation) can be performed.

24.3.8.1 Synchronous Operation by Software

The count operations for respective channels can be started, stopped and cleared simultaneously by setting multiple bits of the GTSTR, GTSTP, and GTCLR registers to 1 at the same time.
Count start with phase differences among channels is possible firstly by setting the GTCNT counter value before counting starts, and then, by setting simultaneously multiple bits in the GTSTR register to 1.
The GTCNT counter value settings are synchronized by setting the GTCR.SSCEN bit to 1, and can be written simultaneously to multiple channels set in the same group by the GTCR.SSCGRP[1:0] bits. Synchronous sets are not valid for channels that are set to complementary PWM mode.
When either the SSCE bit or the SSCD bit of the GTSECR register is set to 1, the GTCR.SSCEN bits of the channels that selected on the GTSECSR register are set to value and the GTCNT synchronous set/clear function of multiple channels are enabled or disabled at the same time.
Because the clock of count operation is selected by GTCR.TPCS[3:0] bits in respective channels, if the clock period of each channel that performs synchronous operation (count start/stop/clear) is different from others, the synchronous operation timings of every channels are not exact same.
The Figure 24.129 shows an example of simultaneous start/stop/clearing of four channels by software, and Figure 24.130 shows an example of phase shift start among four channels by software. Figure 24.131 to Figure 24.133 show examples of simultaneous start/stop/clearing with different count period.

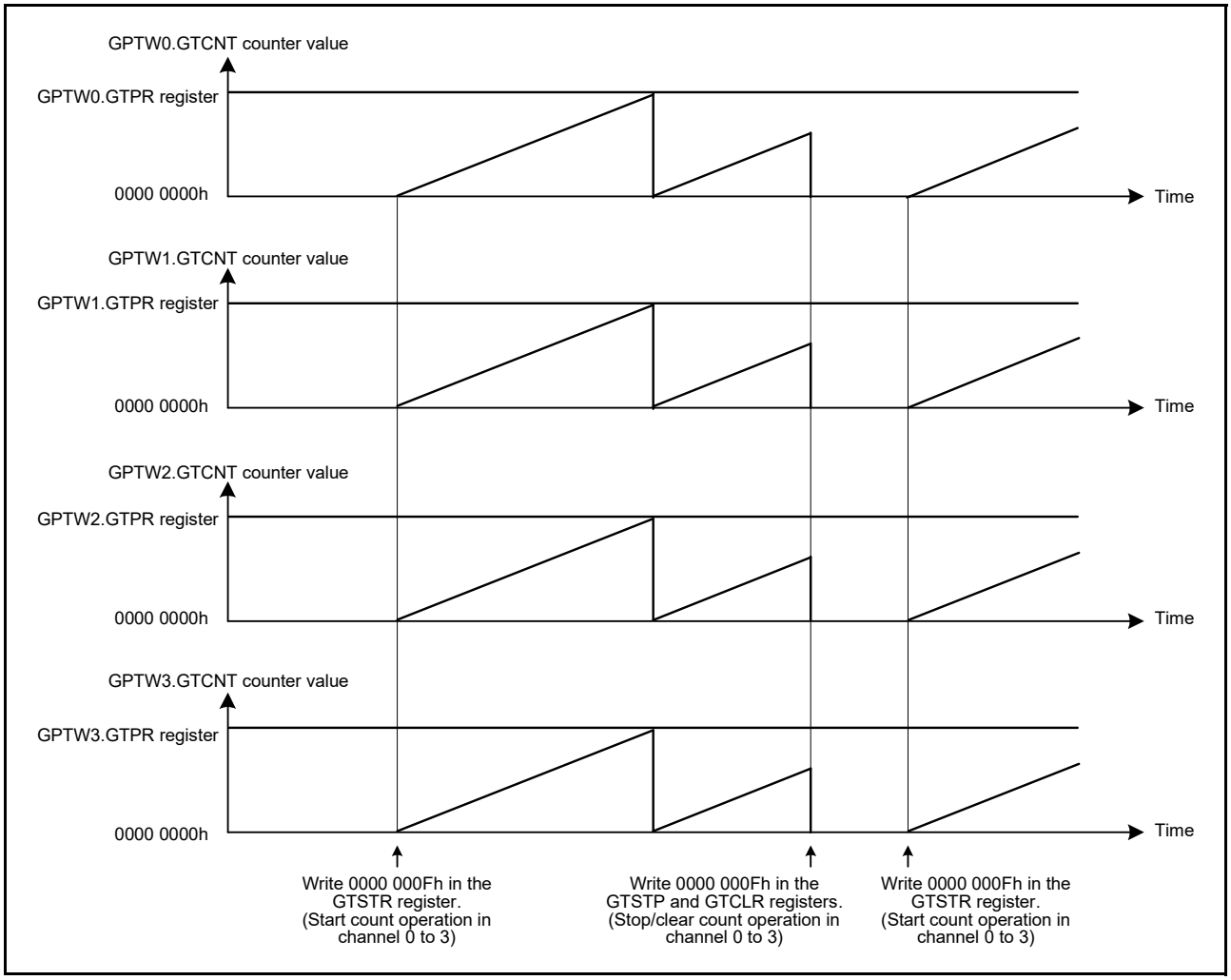


Figure 24.129 Example of Simultaneous Start/Stop/Clearing Operation by Software (with the Same Count Period (GTPR Register Value))

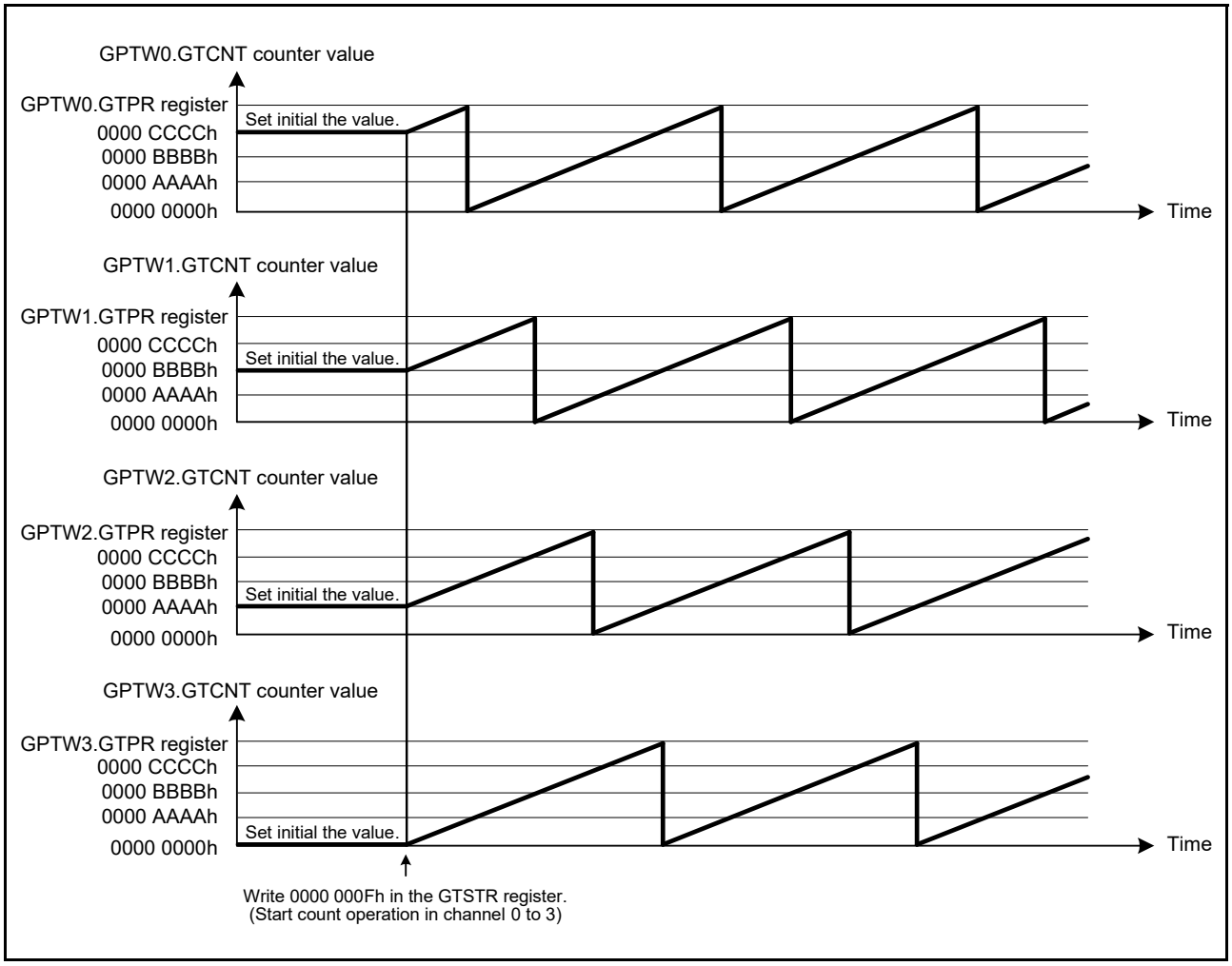


Figure 24.130 Example of Software Phase Shift Start (with Same Count Period (GTPR Register Value))

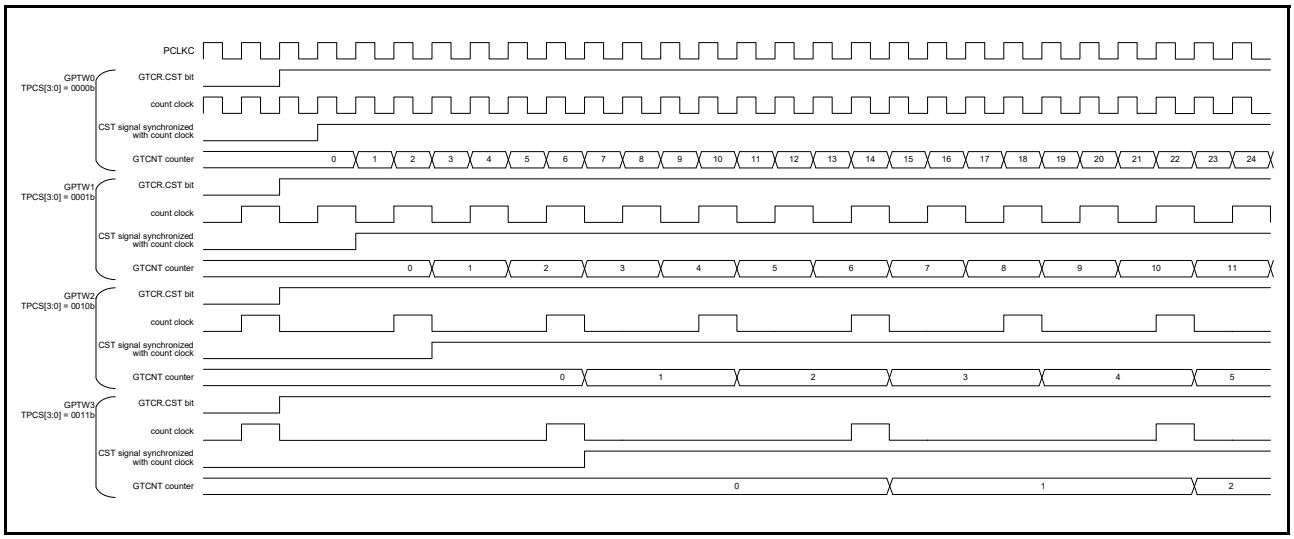


Figure 24.131 Example of Simultaneous Start Operation by Software (with Different Count Period)

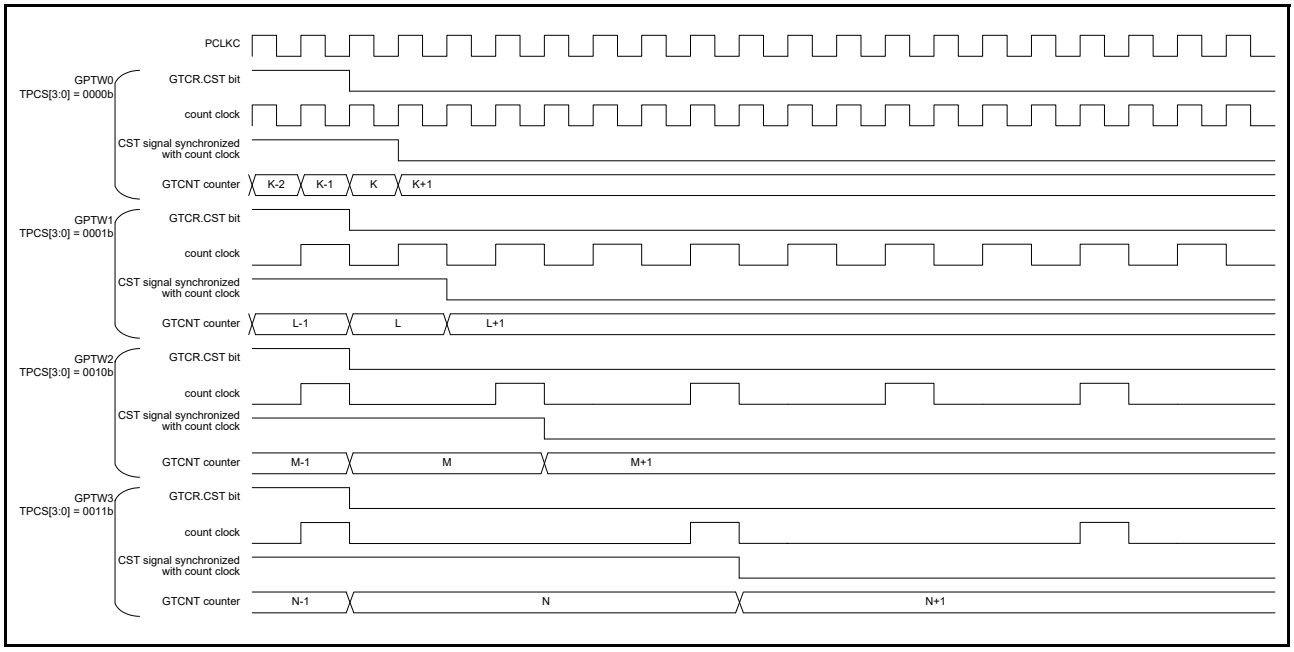


Figure 24.132 Example of Simultaneous Stop Operation by Software (with Different Count Period)

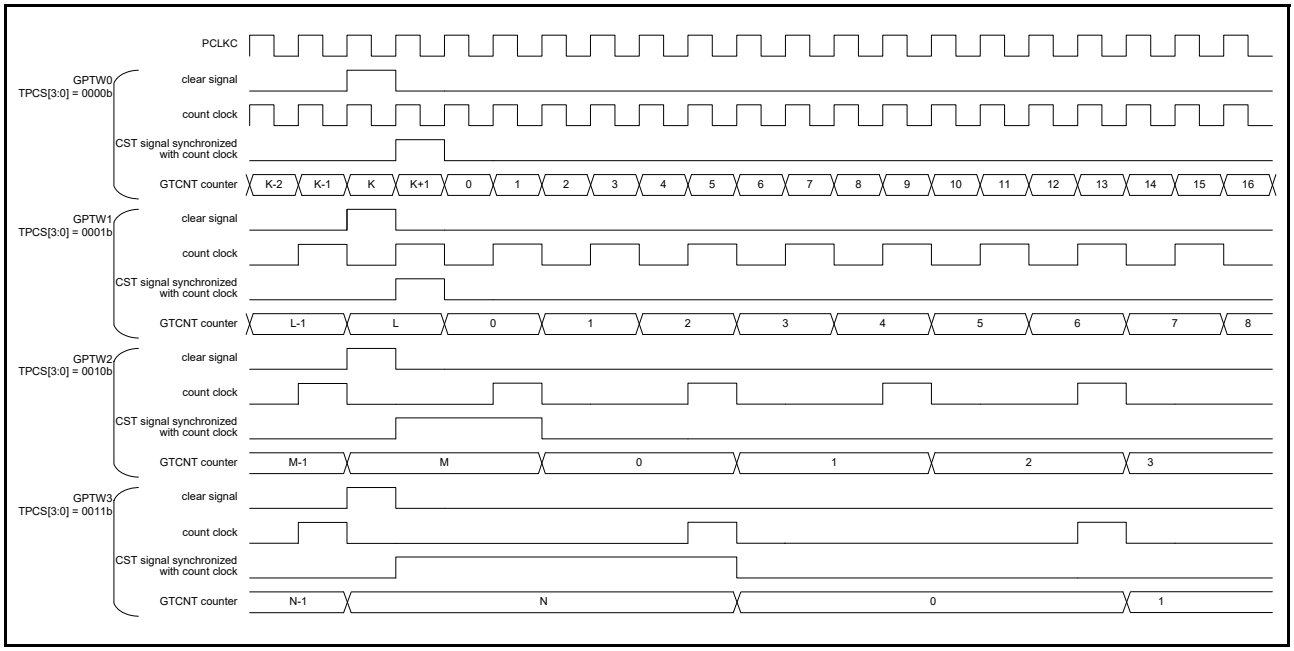


Figure 24.133 Example of Simultaneous Clearing Operation by Software (with Different Count Period)

24.3.8.2 Synchronous Operation by Hardware Source

The count operations for respective channels can be started/stopped/cleared simultaneously by following hardware sources: external trigger input, and ELC event input.

Synchronous operation by the GTIOCnA and GTIOCnB pin inputs (n = 0 to 7) can be performed by setting the ELC event by input capture as a hardware source.

Figure 24.134 shows an example of simultaneous start/stop/clearing operation of four channels by hardware source.

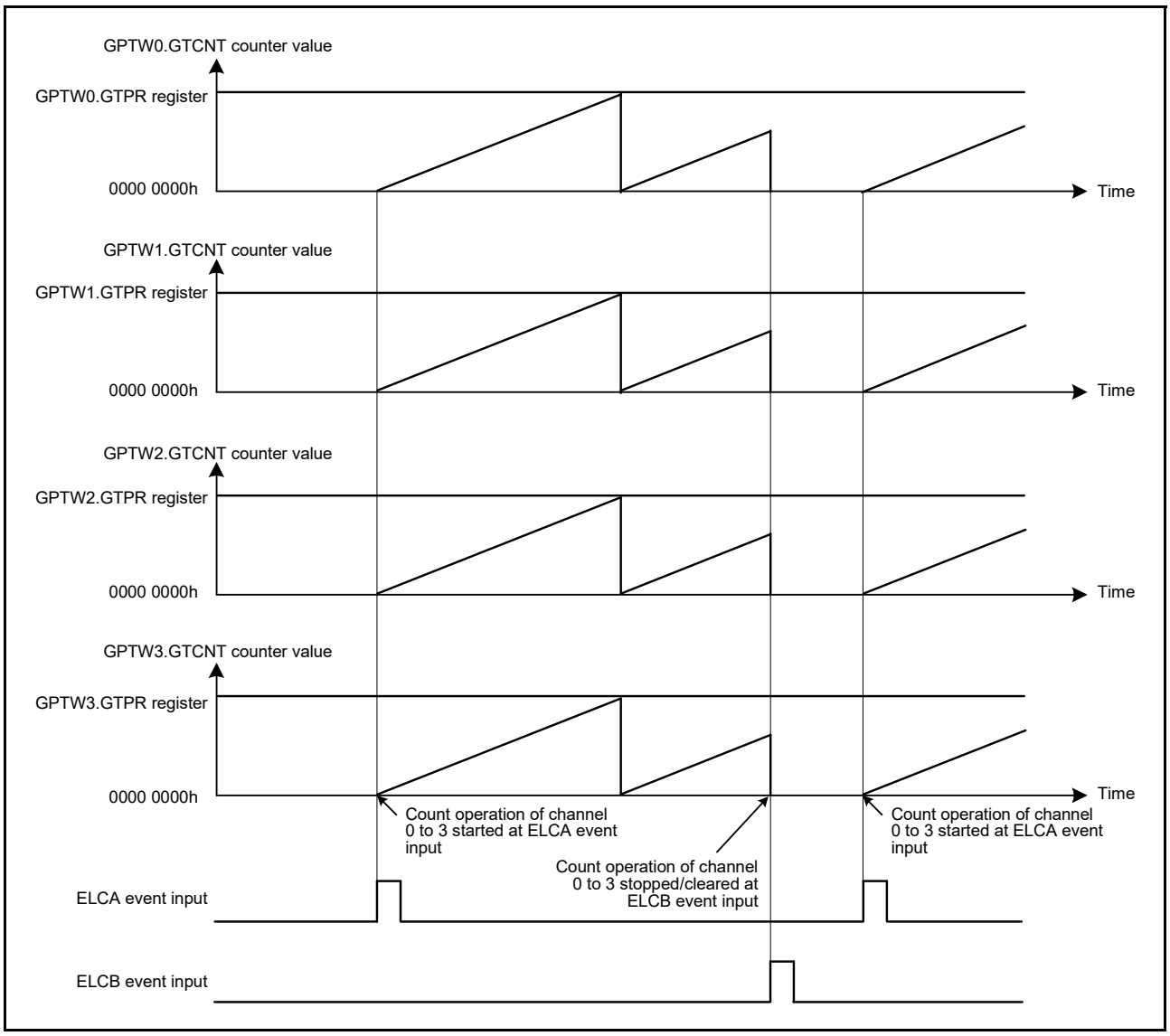


Figure 24.134 Example of Simultaneous Start/Stop/Clearing Operation by Hardware Source (with Same Count Period (GTPR Register Value))

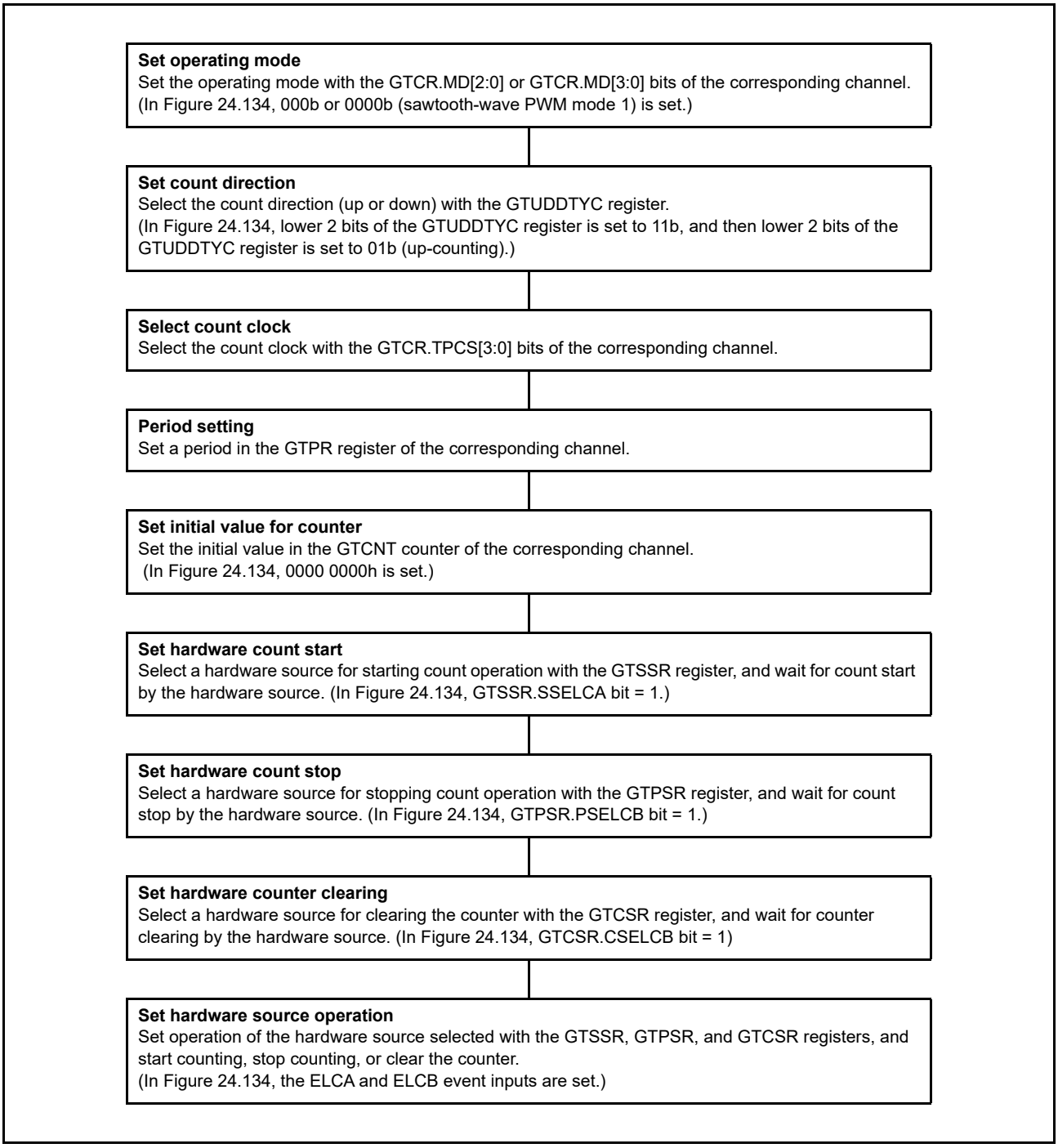


Figure 24.135 Example for Setting Simultaneous Start by Hardware Source

24.3.8.3 Synchronous Clear Operation by Inter Channel Cooperation

The counters of other channels can be cleared at the same time with the counter clear of some channel caused by the compare match, input capture, sawtooth wave up-count overflow, sawtooth wave down-count underflow, and the GTIOCnA/GTIOCnB pin input selected in the GTCSR register.

The counter clear caused by GTCLR register, external trigger input and the ELC input can perform as synchronous clear when the same counter clear factor is set to target channels of synchronous clear. Therefore, these factors are not prepared for the synchronous clear operation by inter channel cooperation.

Set the channel that generates the synchronization clear source and the channel that is synchronized clear in the same synchronous set/clear group with the GTCR.SSCGRP[1:0] bits.

Similar to the synchronous operation in section 24.3.8.1, Synchronous Operation by Software, if the count clock selected by the GTCR.TPCS[3:0] bits is different for each channel, the synchronous clear operation cannot be performed at exact same timing. Similar to the example of simultaneous clearing operation by software in Figure 24.133, if the count clock is different for each channel, the synchronous clear factor is synchronized with the count clock of each channel before counter clearing.

When either the SSCE bit or the SSCD bit of the GTSECR register is set to 1, the GTCR.SSCEN bits of the channels that selected on the GTSECSR register are set to 0 or 1 and the GTCNT synchronous set/clear function of multiple channels are enabled or disabled at the same time.

Table 24.23 lists the settings of the registers used to set sources for synchronous clearing. Figure 24.136 shows an example of synchronous clear operation by inter channel cooperation and Figure 24.137 shows an example for setting synchronous clear operation by inter channel cooperation.

Table 24.23 Settings of the Registers Used to Set Sources for Synchronous Clearing

Sources for Synchronous Clearing by Inter Channel Cooperation	Register.Bit(s)	Value
GTCCRA register compare match/input capture	GTINTAD.SCFA	1b
	GTCSR.CSCMSC[2:0]	001b
GTCCRB register compare match/input capture	GTINTAD.SCFB	1b
	GTCSR.CSCMSC[2:0]	010b
GTCCRC register compare match	GTINTAD.SCFC	1b
	GTCSR.CSCMSC[2:0]	011b
GTCCRD register compare match	GTINTAD.SCFD	1b
	GTCSR.CSCMSC[2:0]	100b
GTCCRE register compare match	GTINTAD.SCFE	1b
	GTCSR.CSCMSC[2:0]	101b
GTCCRF register compare match	GTINTAD.SCFF	1b
	GTCSR.CSCMSC[2:0]	110b
Sawtooth wave up-count overflow	GTINTAD.SCFPO	1b
Sawtooth wave down-count underflow	GTINTAD.SCFPU	1b
Clearing due to input on the GTIOCnA or GTIOCnB pin	GTCR.SCGTIOC	1b

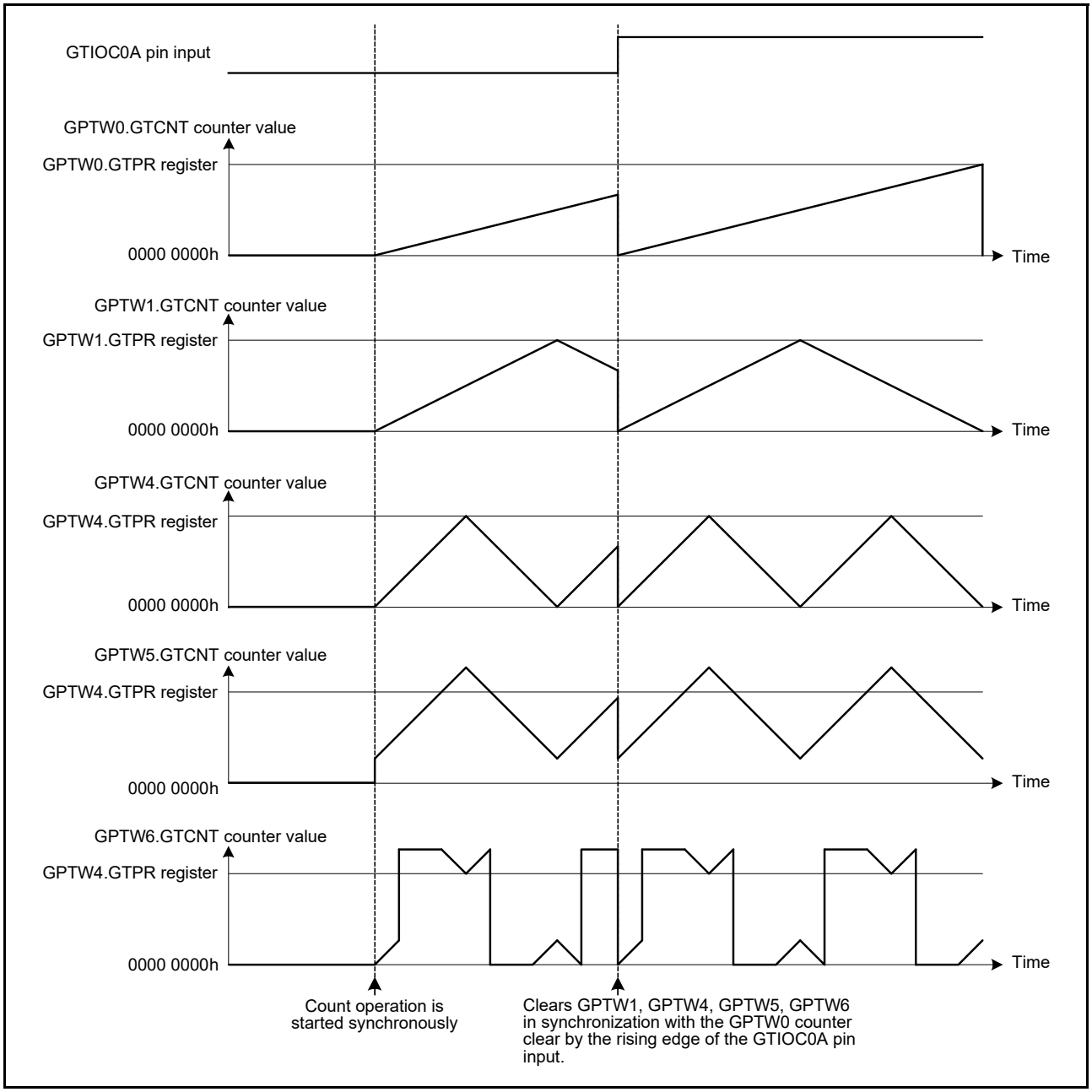


Figure 24.136 Example of Synchronous Clear Operation by Inter Channel Cooperation (GPTW0 is sawtooth wave and counter is cleared by the rising edge of the GTIOC0A, GPTW1 is triangle wave, GPTW4, GPTW5, GPTW6 are complementary PWM mode. GPTW0, GPTW1, GPTW4, GPTW5, GPTW6 are the same synchronous set/clear group)

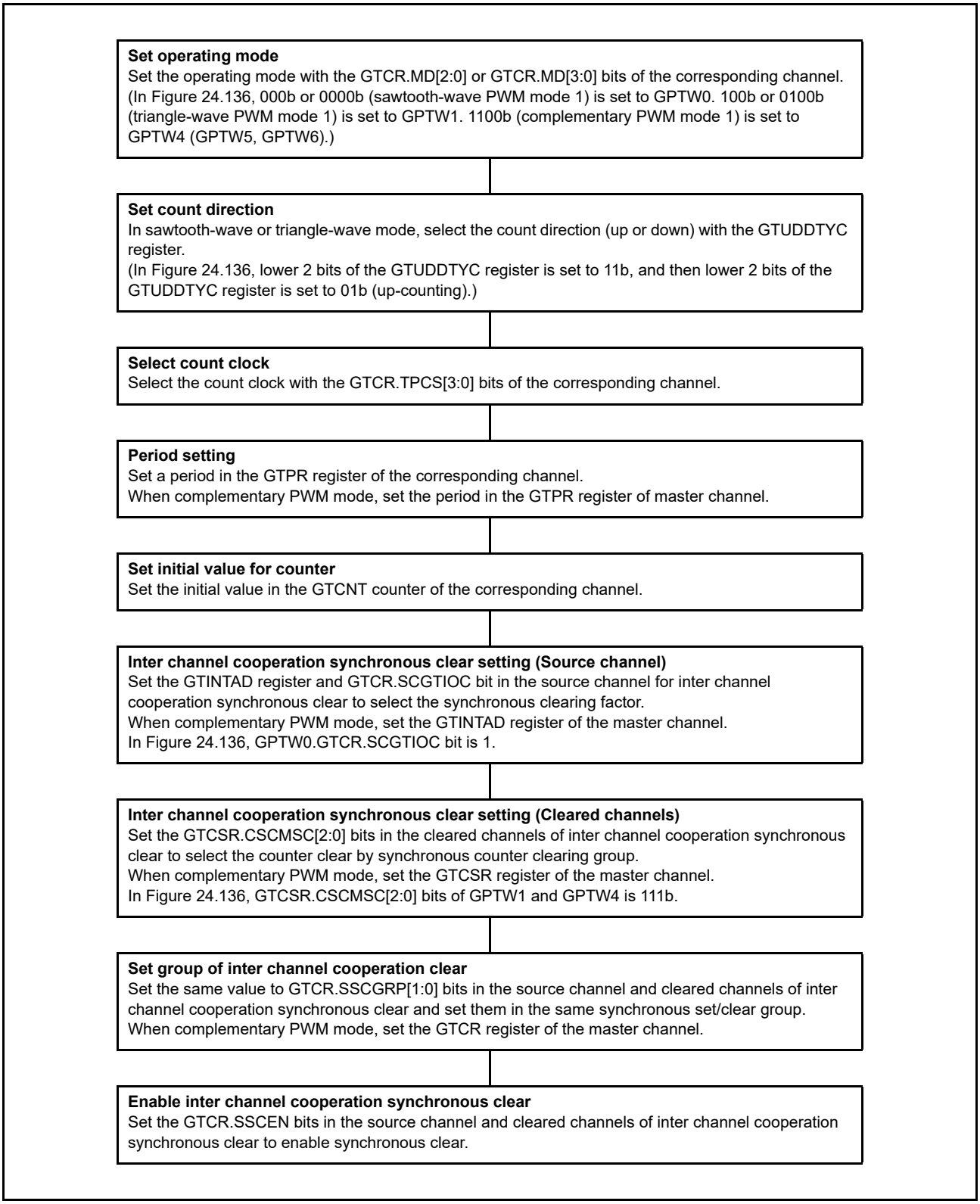


Figure 24.137 Example for Setting Synchronous Clear Operation by Inter Channel Cooperation

24.3.9 PWM Output Operation Examples

(1) Synchronous PWM Output

The GPTW can output two phases of linked PWM waveforms for one channel, or 16 phases of linked PWM waveforms for a maximum of 8 channels by synchronizing operation of the channels.

Figure 24.138 shows an example in which four channels perform synchronous operation in sawtooth-wave PWM mode and eight phases of PWM waveforms are output. The GTIOCnA pin (n = 0 to 3) is set so that it will output low as the initial output, high at a GTCCRA register compare match, and low at the end of the cycle. The GTIOCnB pin is set so that it will output low as the initial output, high at a GTCCRB register compare match, and low at the end of the cycle.

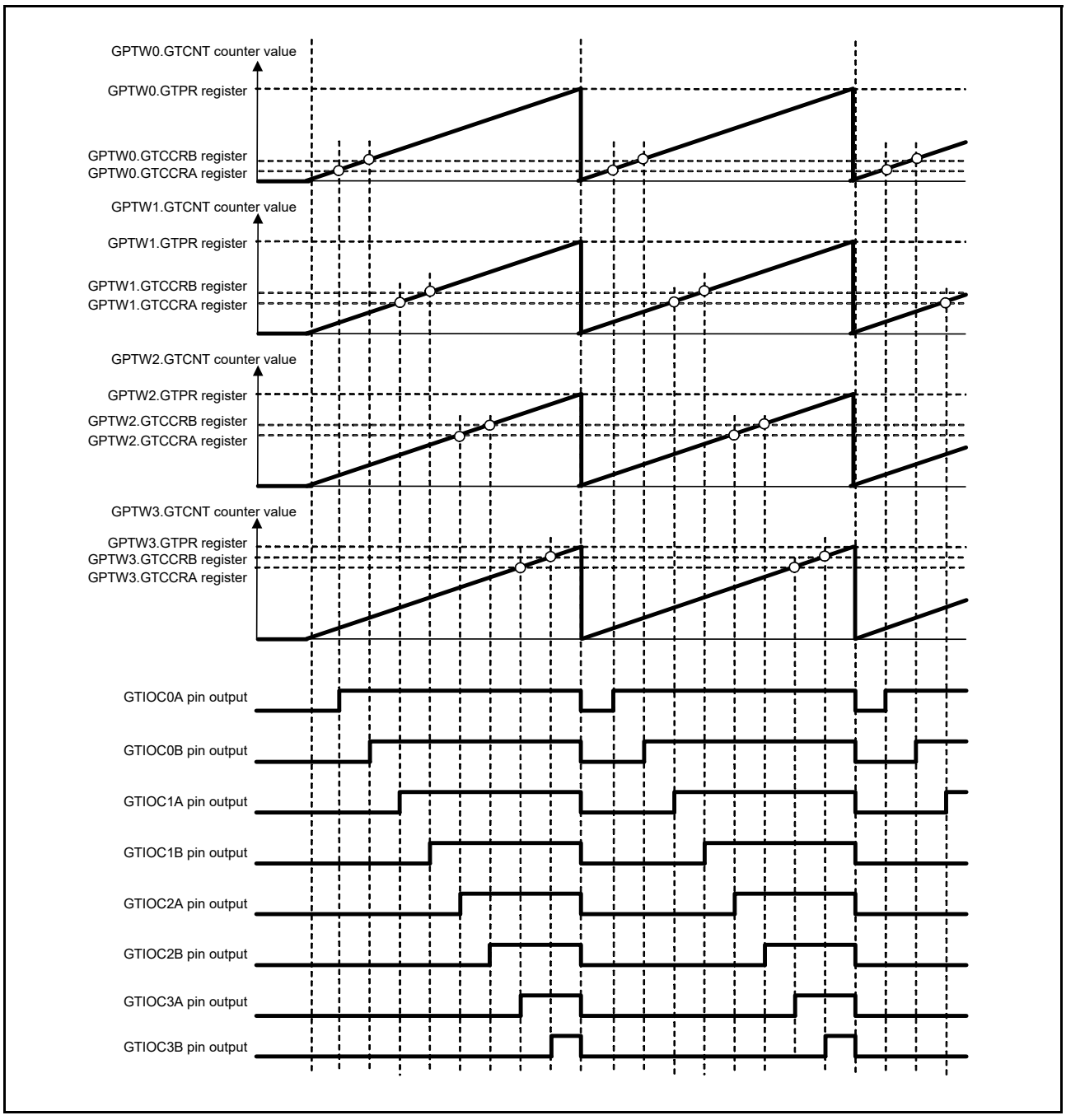


Figure 24.138 Example of Synchronous PWM Output

(2) Three-Phase Sawtooth-Wave Complementary PWM Output

Figure 24.139 shows an example in which three channels perform synchronous operation in sawtooth-wave PWM mode 1 and three-phase complementary PWM waveforms are output. The GTIOCnA pin (n = 0 to 2) is set so that it will output low as the initial output, high at a GTCCRA register compare match, and low at the end of the cycle. The GTIOCnB pin is set so that it will output high as the initial output, low at a GTCCRB register compare match, and high at the end of the cycle.

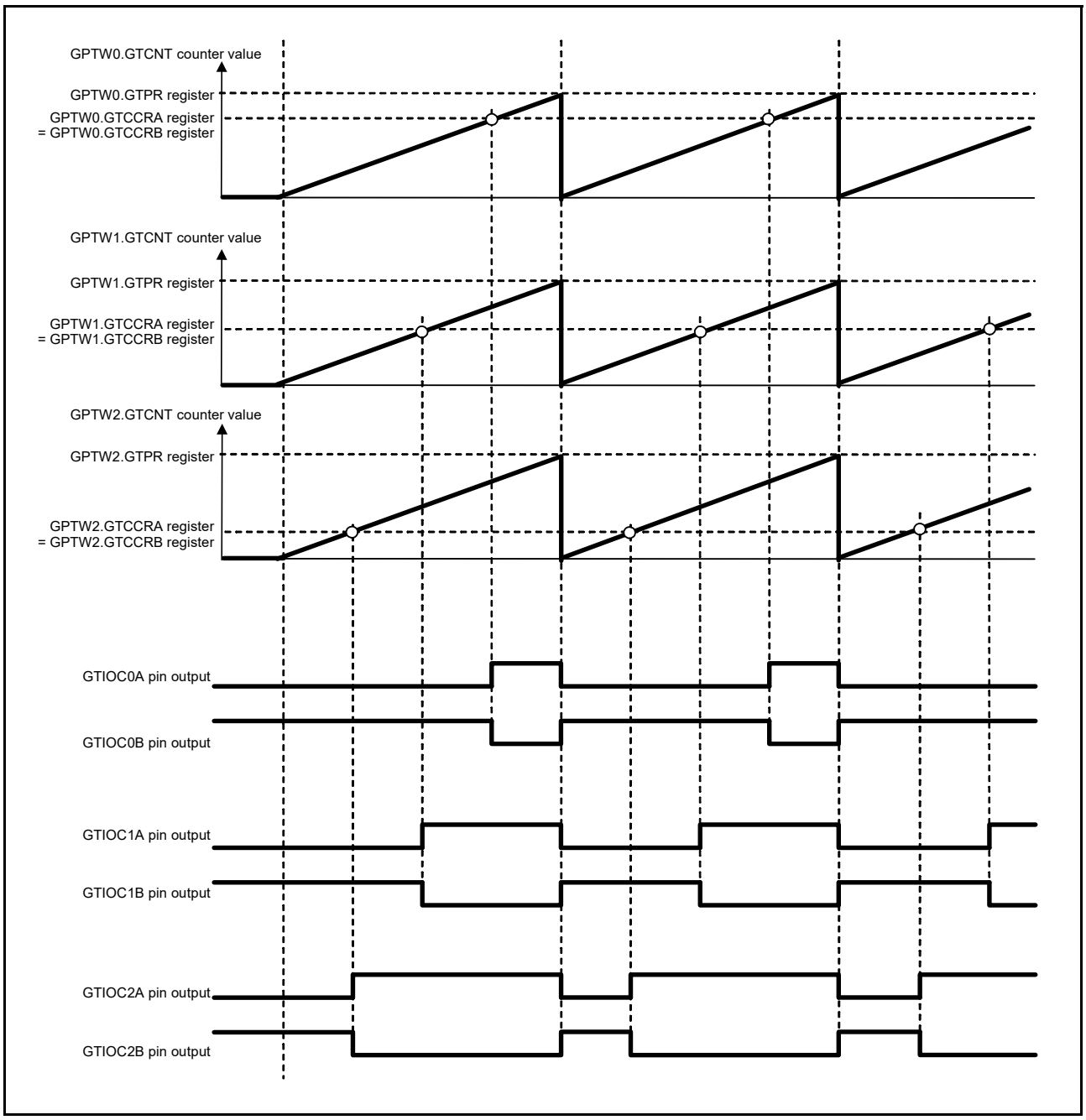


Figure 24.139 Example of Three-Phase Sawtooth-Wave Complementary PWM Output

(3) Three-Phase Sawtooth-Wave Complementary PWM Output with Automatic Dead Time Setting

Figure 24.140 shows an example in which three channels perform synchronous operation in sawtooth-wave one-shot pulse mode with automatic dead time setting and three-phase complementary PWM waveforms are output. The GTIOCnA pin (n = 0 to 2) is set so that it will output low as the initial output, toggle the output at a GTCCRA register compare match, and retain the output at the end of the cycle. The GTIOCnB pin is set so that it will output high as the initial output, toggle the output at a GTCCRB register compare match, and retain the output at the end of the cycle.

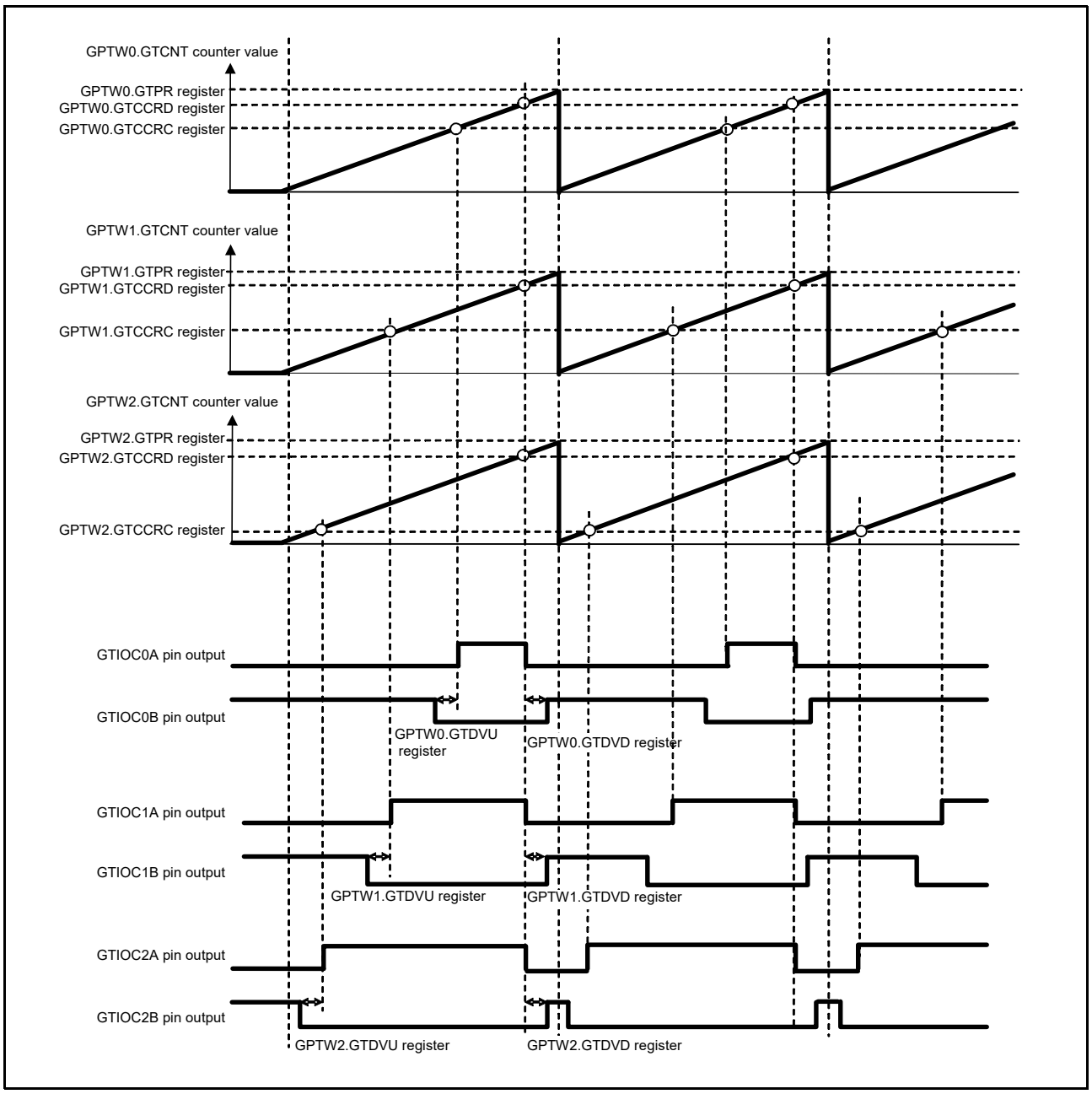


Figure 24.140 Example of Three-Phase Sawtooth-Wave Complementary PWM Output with Automatic Dead Time Setting

(4) Three-Phase Triangle-Wave Complementary PWM Output

Figure 24.141 shows an example in which three channels perform synchronous operation in triangle-wave PWM mode 1 and three-phase complementary PWM waveforms are output. The GTIOCnA pin (n = 0 to 2) is set so that it will output low as the initial output, toggle the output at a GTCCRA register compare match, and retain the output at the end of the cycle. The GTIOCnB pin is set so that it will output high as the initial output, toggle the output at a GTCCRB register compare match, and retain the output at the end of the cycle.

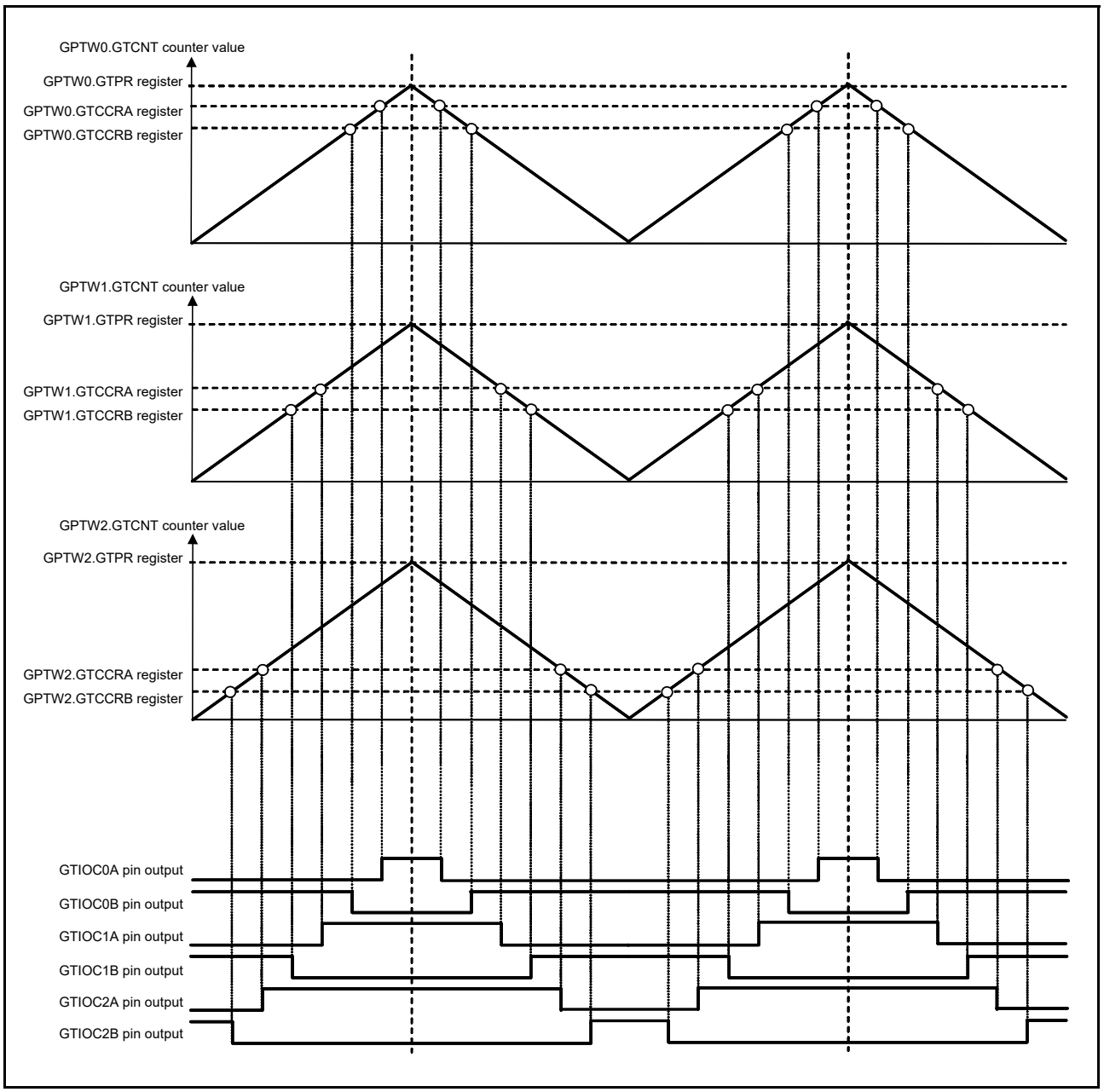


Figure 24.141 Example of Three-Phase Triangle-Wave Complementary PWM Output

(5) Three-Phase Triangle-Wave Complementary PWM Output with Automatic Dead Time Setting

Figure 24.142 shows an example in which three channels perform synchronous operation in triangle-wave PWM mode 1 with automatic dead time setting and three-phase complementary PWM waveforms are output. The GTIOCnA pin (n = 0 to 2) is set so that it will output low as the initial output, toggle the output at a GTCCRA register compare match, and retain the output at the end of the cycle. The GTIOCnB pin is set so that it will output high as the initial output, toggle the output at a GTCCRB register compare match, and retain the output at the end of the cycle.

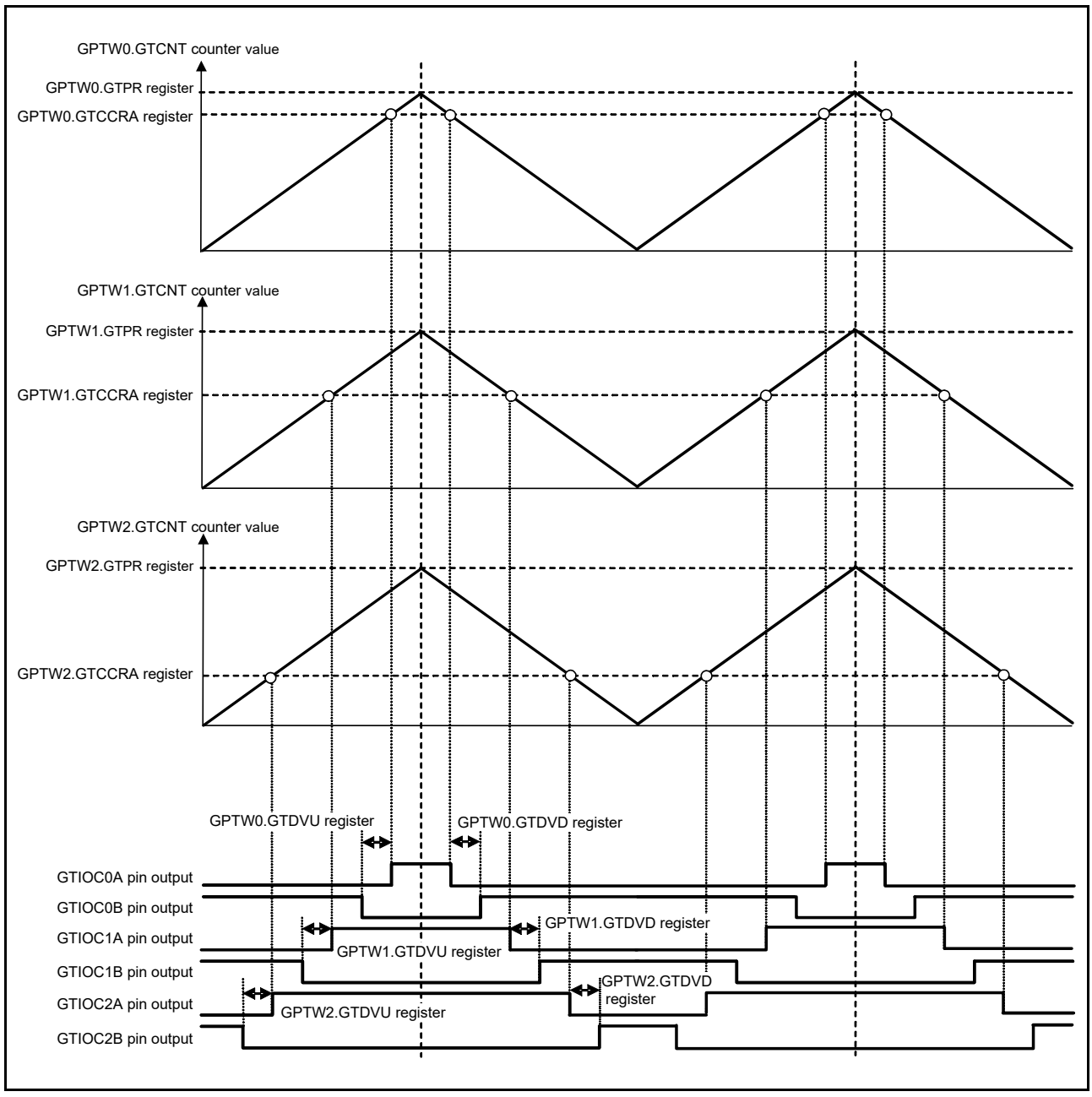


Figure 24.142 Example of Three-Phase Triangle-Wave Complementary PWM Output with Automatic Dead Time Setting

(6) Three-Phase Asymmetric Triangle-Wave Complementary PWM Output with Automatic Dead Time Setting

Figure 24.143 shows an example in which three channels perform synchronous operation in triangle-wave PWM mode 3 with automatic dead time setting and three-phase complementary PWM waveforms are output. The GTIOCnA pin (n = 0 to 2) is set so that it will output low as the initial output, toggle the output at a GTCCRA register compare match, and retain the output at the end of the cycle. The GTIOCnB pin is set so that it will output high as the initial output, toggle the output at a GTCCRB register compare match, and retain the output at the end of the cycle.

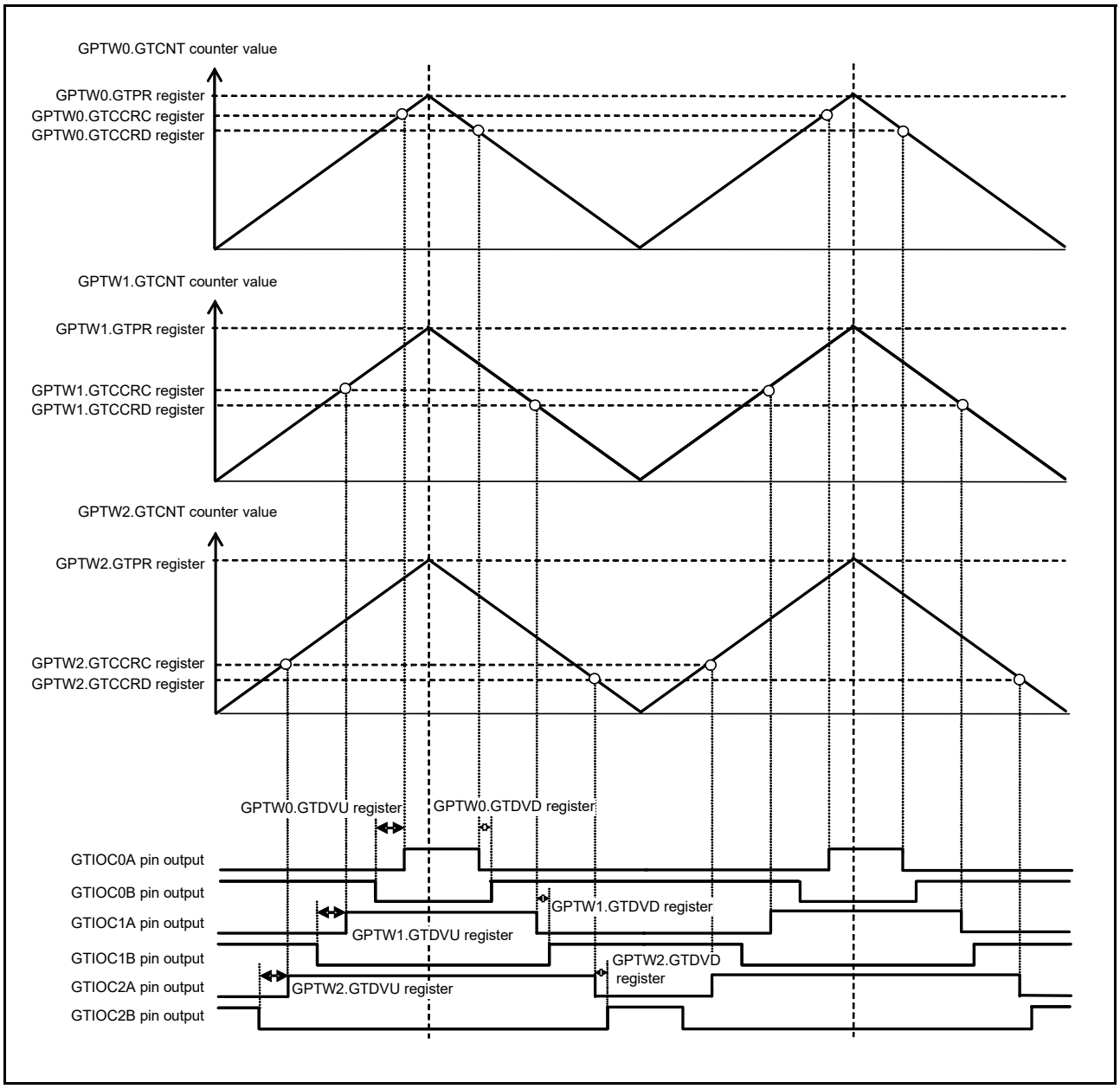


Figure 24.143 Example of Three-Phase Asymmetric Triangle-Wave Complementary PWM Output with Automatic Dead Time Setting

24.3.10 Cycle Count Function

By setting the GTPC register, the end of cycle can be counted.

The number of cycle to be counted should be set into the GTPC.PCNT counter when the GTPC.PCEN bit is 0. When the PCEN bit is 1, the PCNT counter can be read, but writing is disabled. When the PCEN bit is 1, down-counting is performed at the end of cycle. When the PCNT counter is 1 at the end of cycle, it becomes 0 and counting is stopped to finish the cycle count operation. At that time, the GTST.PCF flag is set, and a cycle count end interrupt request (GTCEIn) is generated if the setting of the GTINTAD.GTINTPC bit is 1. When the GTPC.ASTP bit is 1, the GTCNT counter is also stopped at the same time that the cycle count operation is finished.

When the GTCNT counter is stopped while cycle count function is enabled, the PCNT counter keeps its value. When the GTCNT counter restarts counting and the PCEN bit is 1, the PCNT counter restarts down-counting from the hold value. When the PCEN bit is changed from 0 to 1 while the PCNT counter is 0 and the ASTP bit is 1, the GTCNT counter is stopped at the count clock immediately after that.

When either GTSECR.SPCE bit or GTSECR.SPCD bit is set to 1, the PCEN bit in the channels set to 1 by the GTSECSR register is simultaneously set the value to enable or disable the cycle count function for multiple channels.

Figure 24.144 and Figure 24.145 show examples of PWM cycle count function.

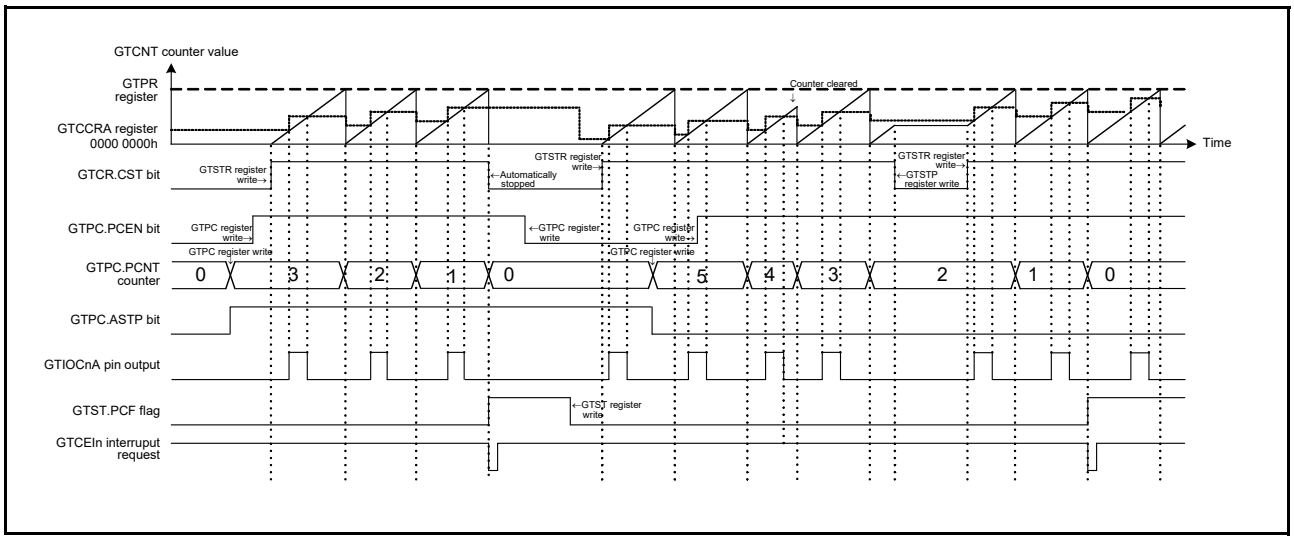


Figure 24.144 Example of PWM Cycle Count Function (Sawtooth-Wave One-Shot Pulse Mode) (n = 0 to 973)

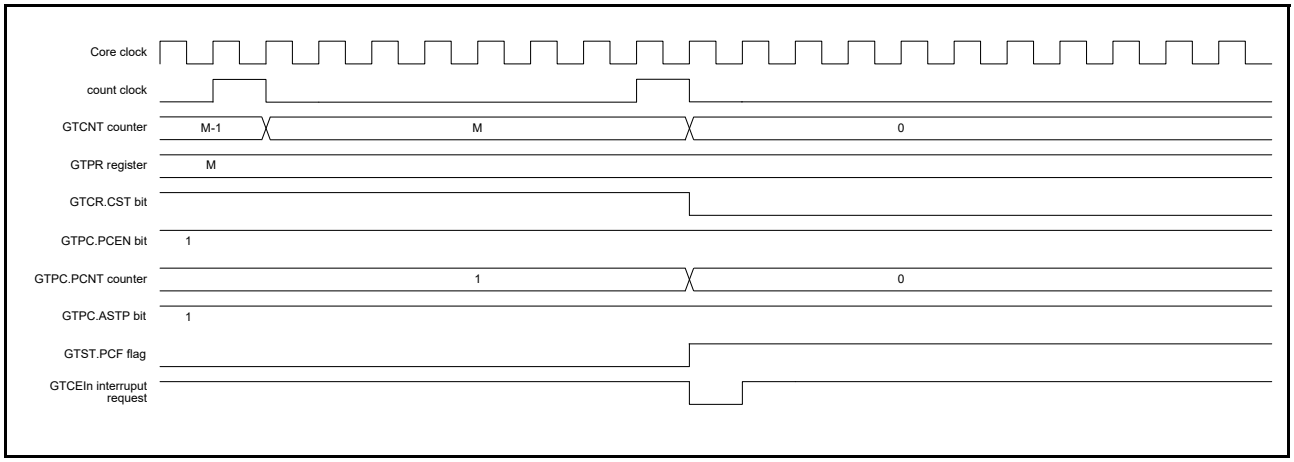


Figure 24.145 Example of the Timing of Operations for PWM Cycle Count Function (Sawtooth-Wave, Up-Counting) (n = 0 to 973)

24.3.11 Phase Counting Mode

The GTCNT counter can be up-counting or down-counting by detecting the phase difference between the GTIOCnA pin input (n = 0 to 7) and the GTIOCnB pin input.
Detection of the phase difference can be performed by setting the relationship between the edge and the level of the GTIOCnA pin input and the GTIOCnB pin input using the GTUPSR and GTDNSR registers for input combination.
Refer to section 24.3.1.1, Counter Operation for count operation.
Figure 24.146 to Figure 24.155 show the examples of setting procedure for the phase counting mode 1 to 5, and the Table 24.24 to Table 24.33 show the setting of the GTUPSR and GTDNSR registers and up-counting and down-counting conditions.

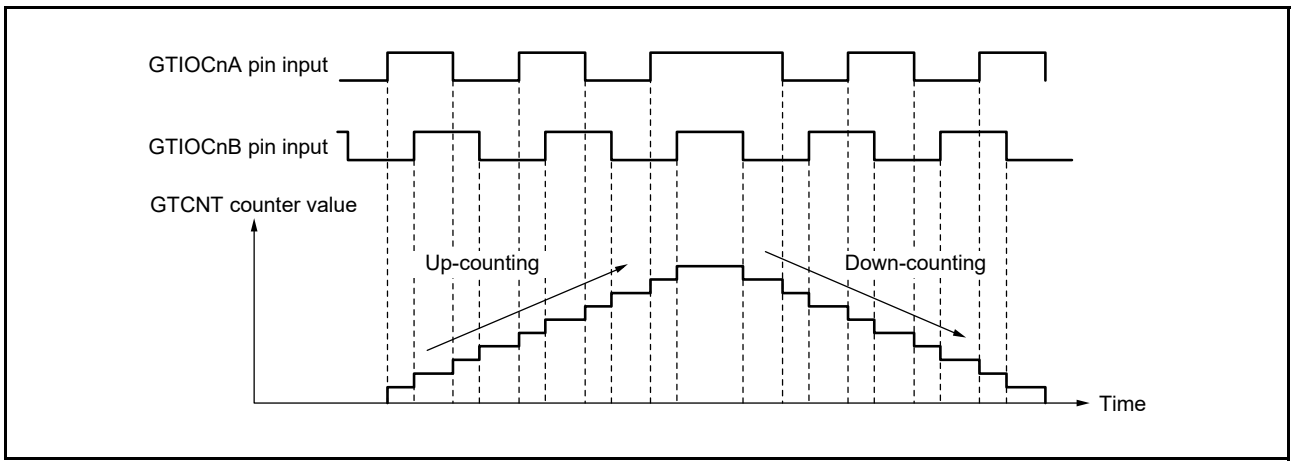


Figure 24.146 Example of setting procedure for the phase counting mode 1 (n = 0 to 7)

Table 24.24 Up-counting and down-counting conditions for the phase counting mode 1 (n = 0 to 7)

GTIOCnA pin input	GTIOCnB pin input	Operation	Setting of register
High	↑	Up-counting	GTUPSR = 0000 6900h GTDNSR = 0000 9600h
Low	↓		
↑	Low		
↓	High		
High	↓	Down-counting	
Low	↑		
↑	High		
↓	Low		

↑: Rising edge

↓: Falling edge

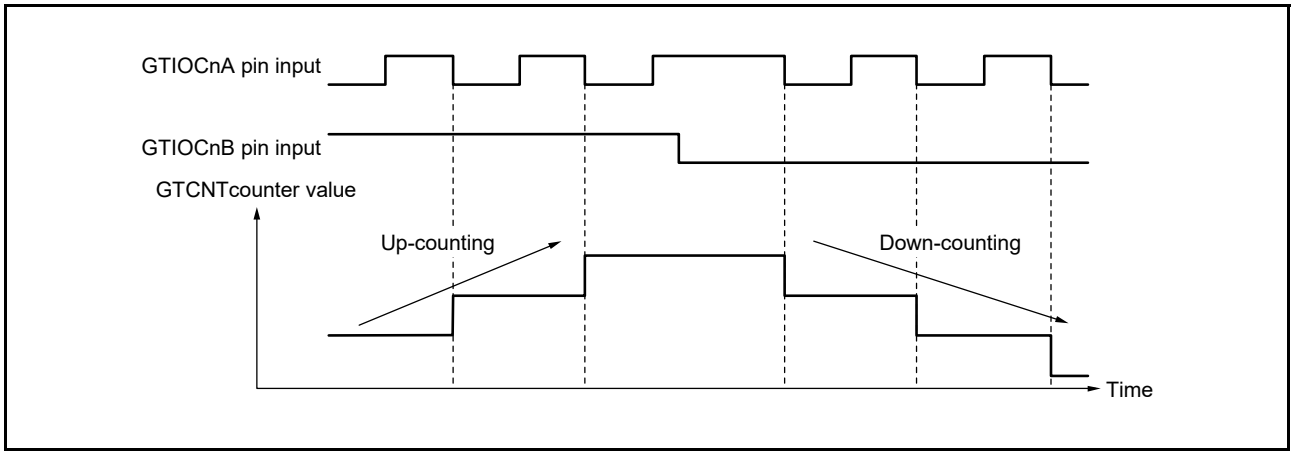


Figure 24.147 Example of setting procedure for the phase counting mode 2 (n = 0 to 7)

Table 24.25 Up-counting and down-counting conditions for the phase counting mode 2 (n = 0 to 7)

GTIOCnA pin input	GTIOCnB pin input	Operation	Setting of register
High	↑	Don't care	GTUPSR = 0000 0800h GTDNSR = 0000 0400h
Low	↓		
↑	Low		
↓	High	Up-counting	
High	↓	Don't care	
Low	↑		
↑	High		
↓	Low	Down-counting	

↑: Rising edge

↓: Falling edge

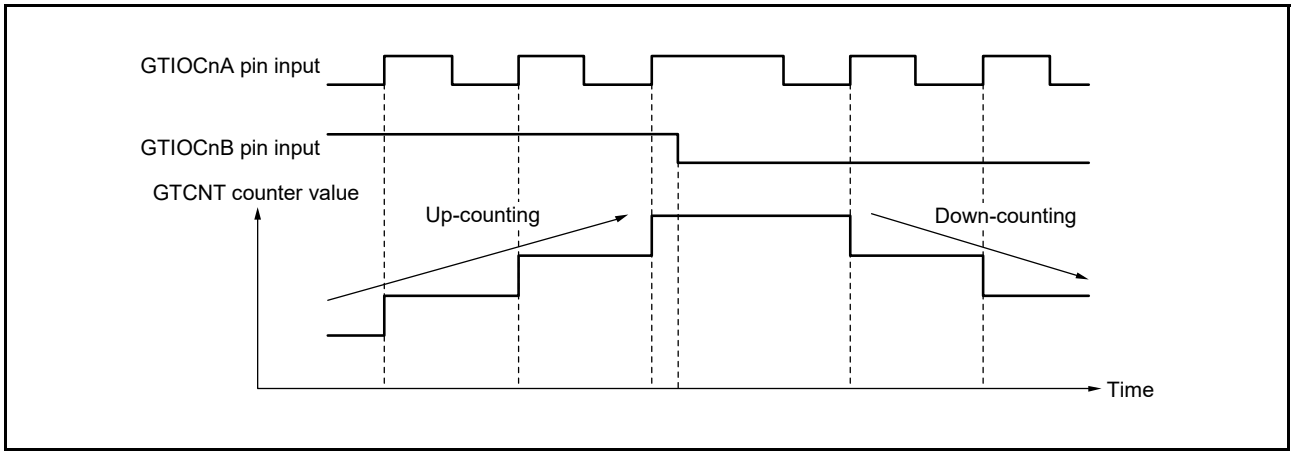


Figure 24.148 Example of setting procedure for the phase counting mode 2 (n = 0 to 7)

Table 24.26 Up-counting and down-counting conditions for the phase counting mode 2 (n = 0 to 7)

GTIOCnA pin input	GTIOCnB pin input	Operation	Setting of register
High	↑	Don't care	GTUPSR = 0000 0200h GTDNSR = 0000 0100h
Low	↓		
↑	Low	Down-counting	
↓	High	Don't care	
High	↓		
Low	↑		
↑	High	Up-counting	
↓	Low	Don't care	

↑: Rising edge

↓: Falling edge

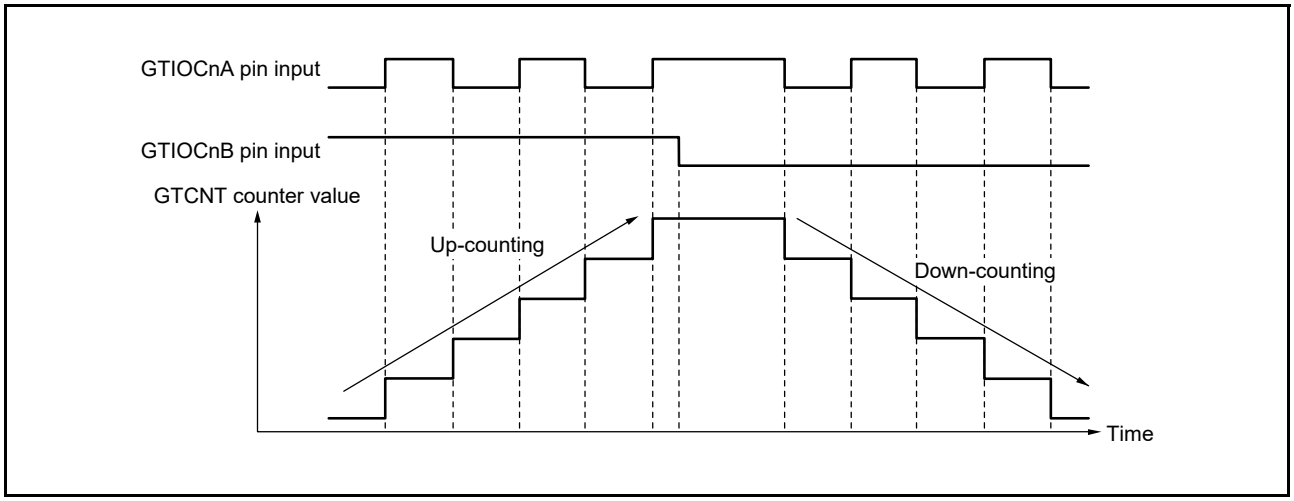


Figure 24.149 Example of setting procedure for the phase counting mode 2 (n = 0 to 7)

Table 24.27 Up-counting and down-counting conditions for the phase counting mode 2 (n = 0 to 7)

GTIOCnA pin input	GTIOCnB pin input	Operation	Setting of register
High	↑	Don't care	GTUPSR = 0000 0A00h GTDNSR = 0000 0500h
Low	↓		
↑	Low	Down-counting	
↓	High	Up-counting	
High	↓	Don't care	
Low	↑		
↑	High	Up-counting	
↓	Low	Down-counting	

↑: Rising edge

↓: Falling edge

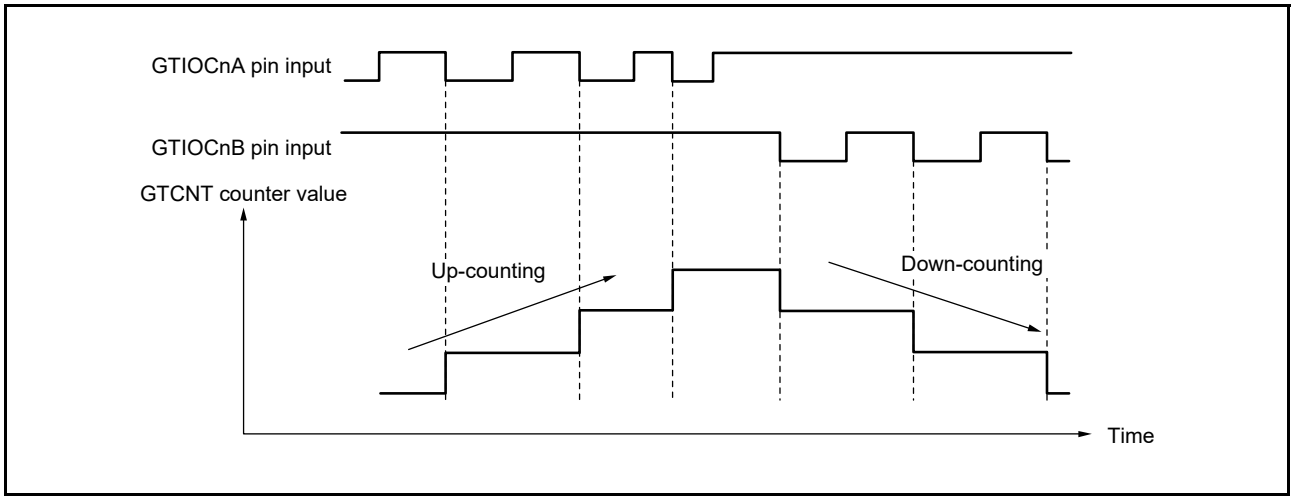


Figure 24.150 Example of setting procedure for the phase counting mode 3 (n = 0 to 7)

Table 24.28 Up-counting and down-counting conditions for the phase counting mode 3 (n = 0 to 7)

GTIOCnA pin input	GTIOCnB pin input	Operation	Setting of register
High	↑	Don't care	GTUPSR = 0000 0800h GTDNSR = 0000 8000h
Low	↓		
↑	Low		
↓	High	Up-counting	
High	↓	Down-counting	
Low	↑	Don't care	
↑	High		
↓	Low		

↑: Rising edge

↓: Falling edge

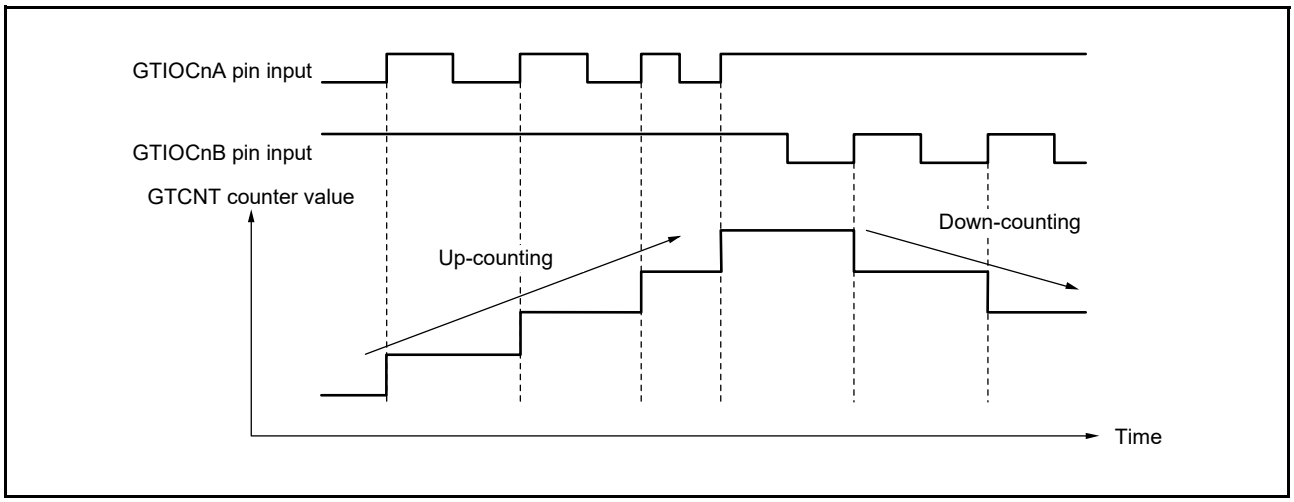


Figure 24.151 Example of setting procedure for the phase counting mode 3 (n = 0 to 7)

Table 24.29 Up-counting and down-counting conditions for the phase counting mode 3 (n = 0 to 7)

GTIOCnA pin input	GTIOCnB pin input	Operation	Setting of register
High	↑	Down-counting	GTUPSR = 0000 0200h GTDNSR = 0000 2000h
Low	↓	Don't care	
↑	Low		
↓	High		
High	↓		
Low	↑		
↑	High	Up-counting	
↓	Low	Don't care	

↑: Rising edge

↓: Falling edge

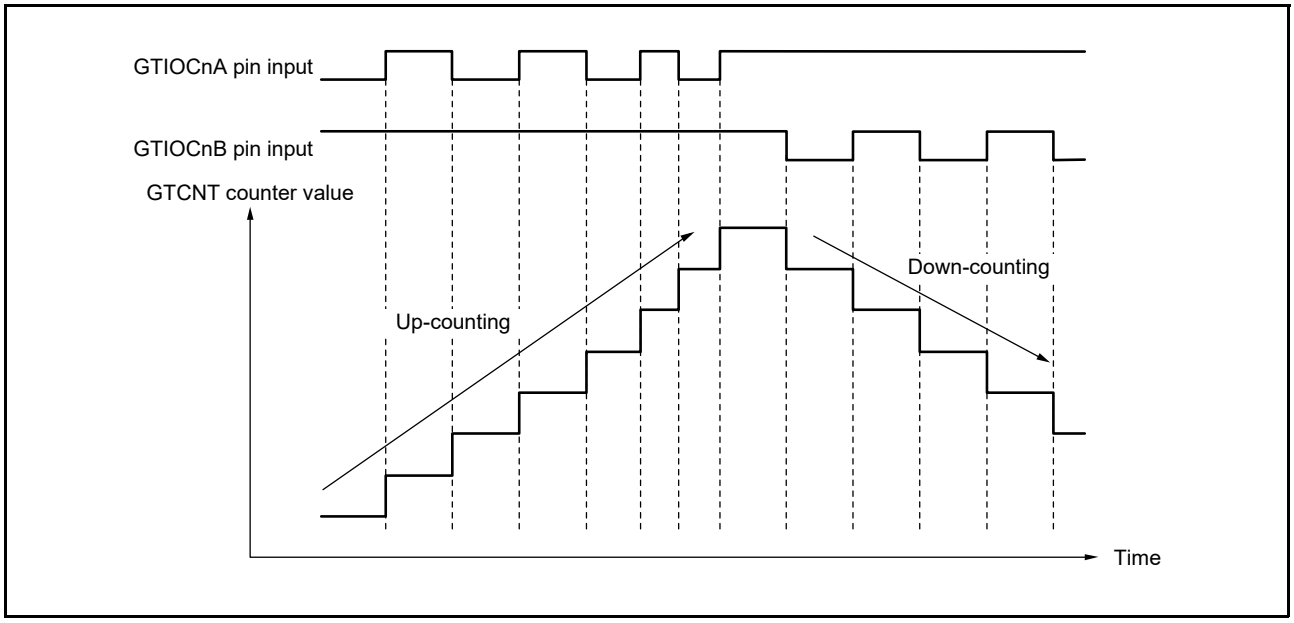


Figure 24.152 Example of setting procedure for the phase counting mode 3 (n = 0 to 7)

Table 24.30 Up-counting and down-counting conditions for the phase counting mode 3 (n = 0 to 7)

GTIOCnA pin input	GTIOCnB pin input	Operation	Setting of register
High	↑	Down-counting	GTUPSR = 0000 0A00h GTDNSR = 0000 A000h
Low	↓	Don't care	
↑	Low		
↓	High	Up-counting	
High	↓	Down-counting	
Low	↑	Don't care	
↑	High	Up-counting	
↓	Low	Don't care	

↑: Rising edge

↓: Falling edge

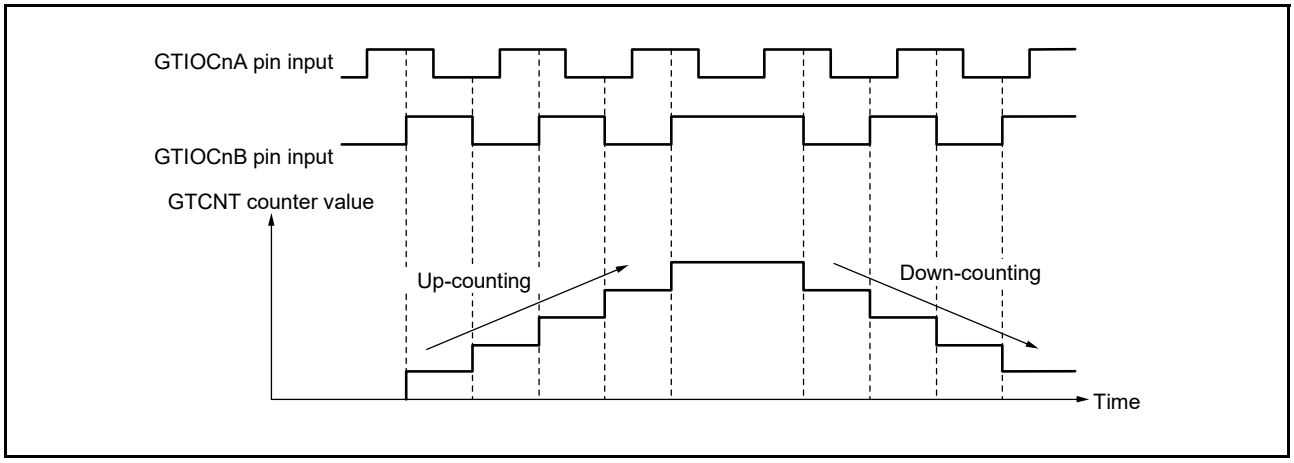


Figure 24.153 Example of setting procedure for the phase counting mode 4 (n = 0 to 7)

Table 24.31 Up-counting and down-counting conditions for the phase counting mode 4 (n = 0 to 7)

GTIOCnA pin input	GTIOCnB pin input	Operation	Setting of register
High	↑	Up-counting	GTUPSR = 0000 6000h GTDNSR = 0000 9000h
Low	↓		
↑	Low	Don't care	
↓	High		
High	↓	Down-counting	
Low	↑		
↑	High	Don't care	
↓	Low		

↑: Rising edge

↓: Falling edge

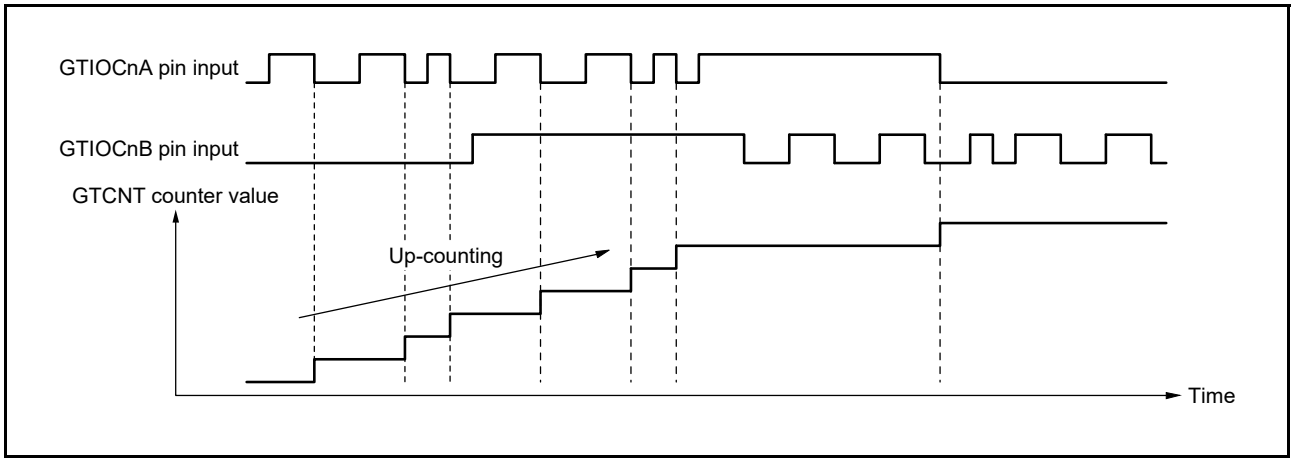


Figure 24.154 Example of setting procedure for the phase counting mode 5 (n = 0 to 7)

Table 24.32 Up-counting and down-counting conditions for the phase counting mode 5 (n = 0 to 7)

GTIOCnA pin input	GTIOCnB pin input	Operation	Setting of register
High	↑	Don't care	GTUPSR = 0000 0C00h GTDNSR = 0000 0000h
Low	↓		
↑	Low		
↓	High	Up-counting	
High	↓	Don't care	
Low	↑		
↑	High		
↓	Low	Up-counting	

↑: Rising edge

↓: Falling edge

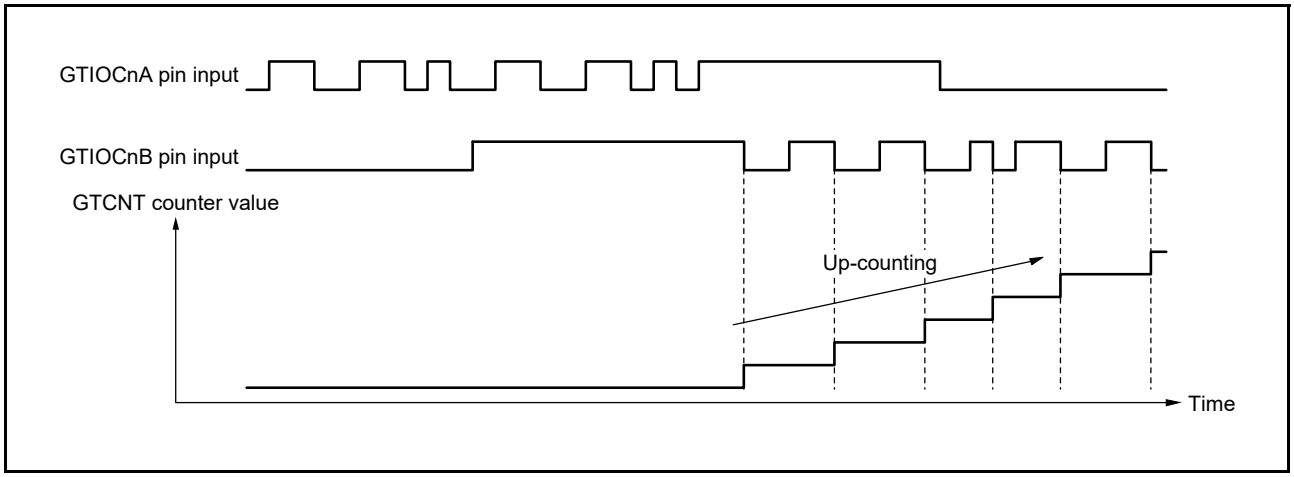


Figure 24.155 Example of setting procedure for the phase counting mode 5 (n = 0 to 7)

Table 24.33 Up-counting and down-counting conditions for the phase counting mode 5 (n = 0 to 7)

GTIOCnA pin input	GTIOCnB pin input	Operation	Setting of register
High	↑	Don't care	GTUPSR = 0000 C000h GTDNSR = 0000 0000h
Low	↓	Up-counting	
↑	Low	Don't care	
↓	High		
High	↓	Up-counting	
Low	↑	Don't care	
↑	High		
↓	Low		

↑: Rising edge

↓: Falling edge

24.3.12 External Pulse Width Measuring Function

The pulse width of GTIOCnA pin input (n = 0 to 7), GTIOCnB pin input, and GTETRGA/GTETRGB/GTETRGC/GTETRGD pin inputs can be measured.

The setting to enable or disable count-up of the GTCNT counter and the input pin and level which measured pulse width are selected by the USILVL[3:0] bits of the GTUPSR register.

The setting to enable or disable count-down of the GTCNT counter and the input pin and level which measured pulse width are selected by the DSILVL[3:0] bits of the GTDNSR register.

The setting to enable both count-up and count-down of the GTCNT counter at the same time is prohibited.

The counting operation performs cycle counting with the period of the GTPR register.

If the phase counting function and the pulse width measuring function are enabled at the same time, the pulse width measuring function does not work and the phase counting function works.

Figure 24.156 and Figure 24.157 show examples of external pulse width measuring function and Figure 24.158 shows an example for setting external pulse width measuring function.

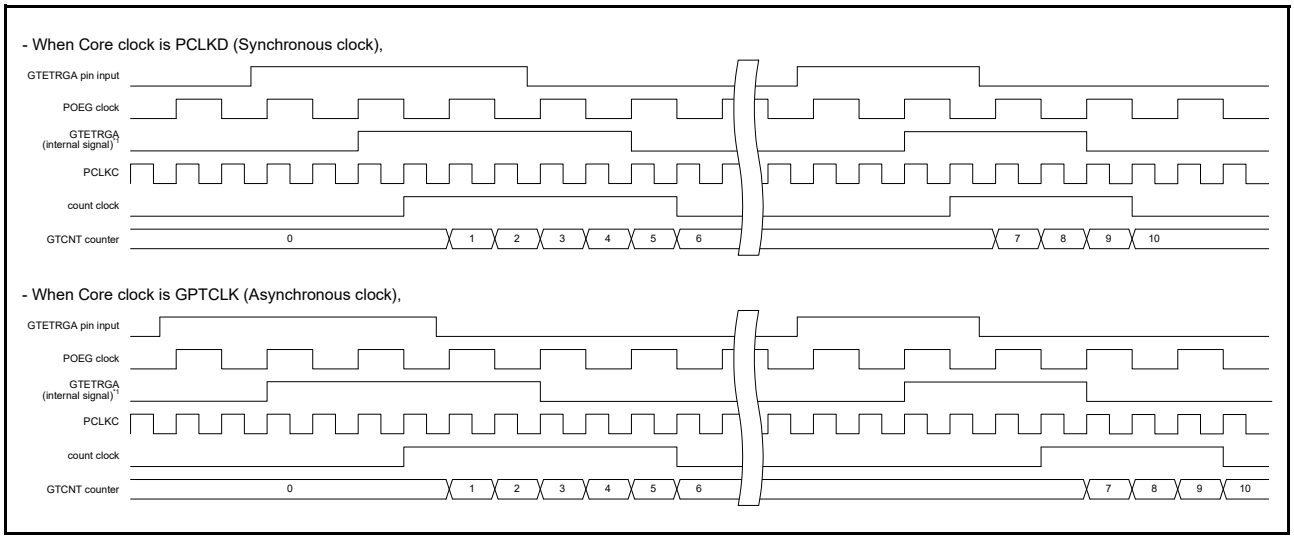


Figure 24.156 Example of External Pulse Width Measuring Function (Up-Counting)

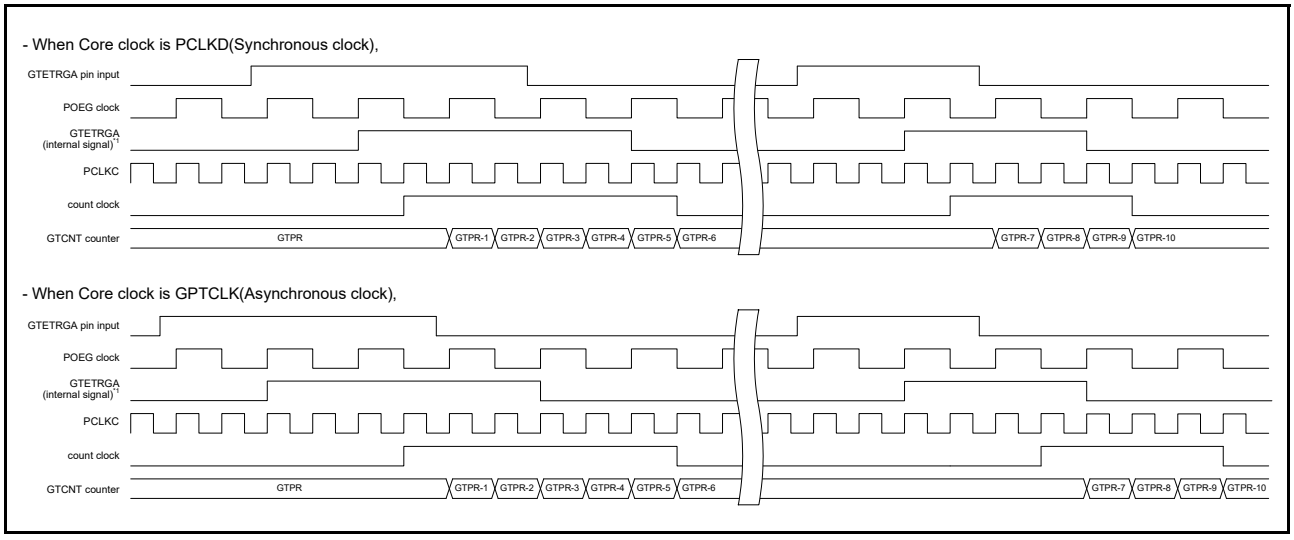


Figure 24.157 Example of External Pulse Width Measuring Function (Down-Counting)

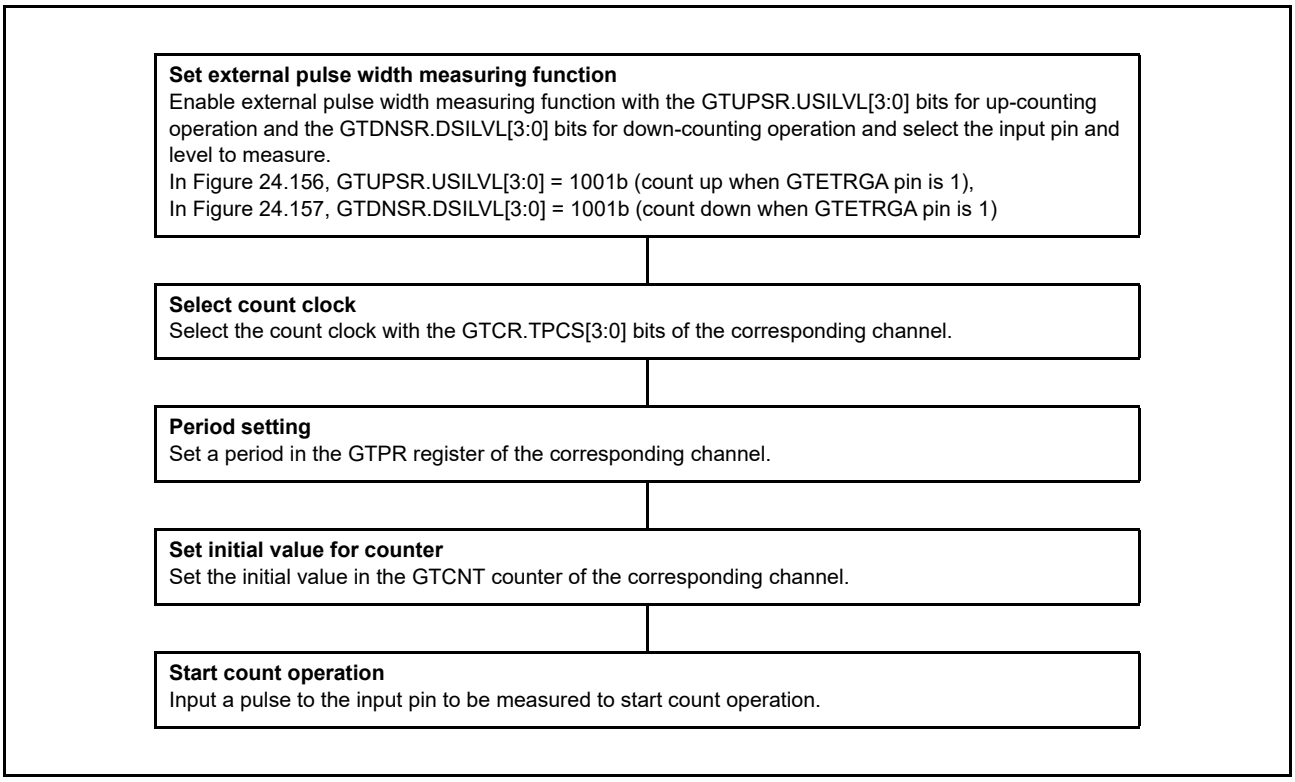


Figure 24.158 Example for Setting External Pulse Width Measuring Function

24.3.13 Output Phase Switching Control Function (OPS)

OPS can easily control Brushless DC motor using Output Phase Switching Control Register (OPSCR).
OPS uses S/W setting value (OPSCR.UF, VF, WF bits) or external signals detected by the Hall element as input signals.
OPS outputs either level signals or chopped signals by GPTW0's PWM as the 6-phase (U-positive phase/negative phase, V-positive phase/negative phase, W-positive phase/negative phase) signals to control motor.
Figure 24.159 shows the block diagram of OPS.
The GPT_UVWEDGE signal is output signal to ELC generated by detecting the edge of input signal.

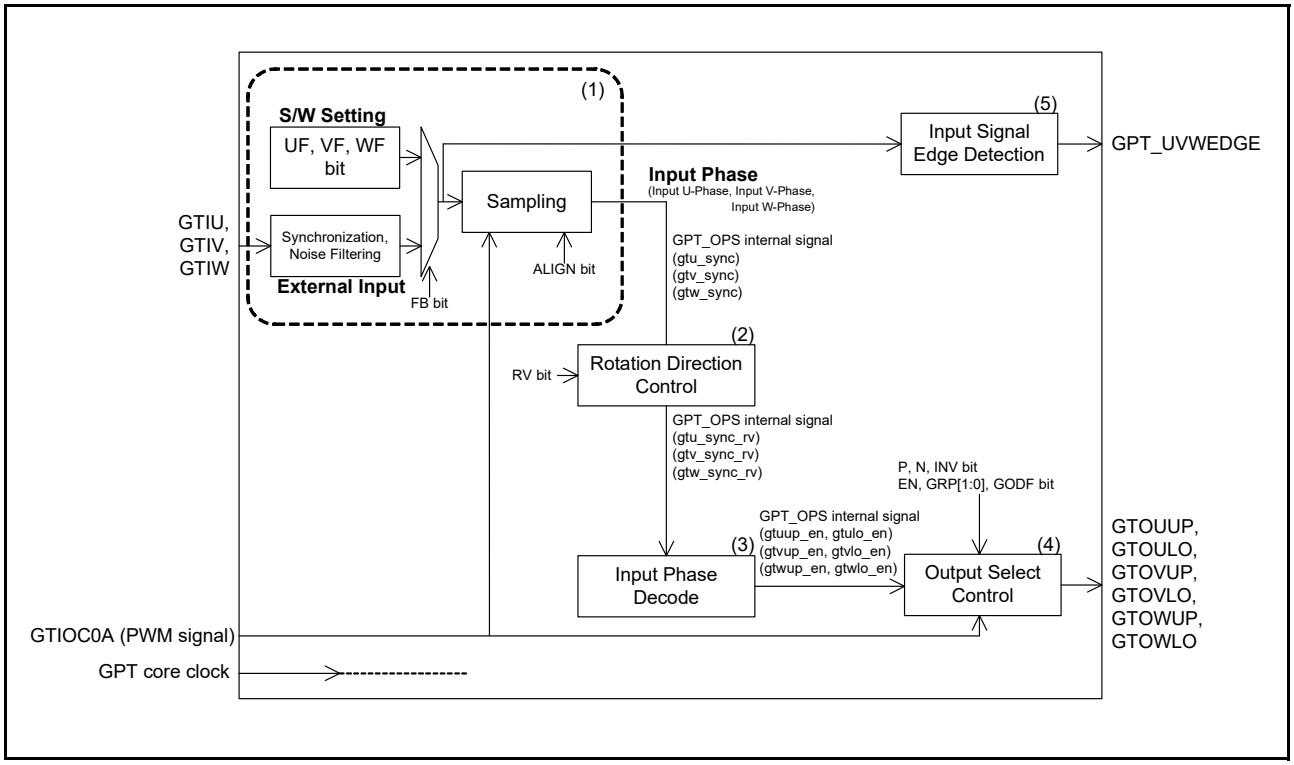


Figure 24.159 OPS Block Diagram

Figure 24.160 and Figure 24.161 show examples of OPS level output operation.

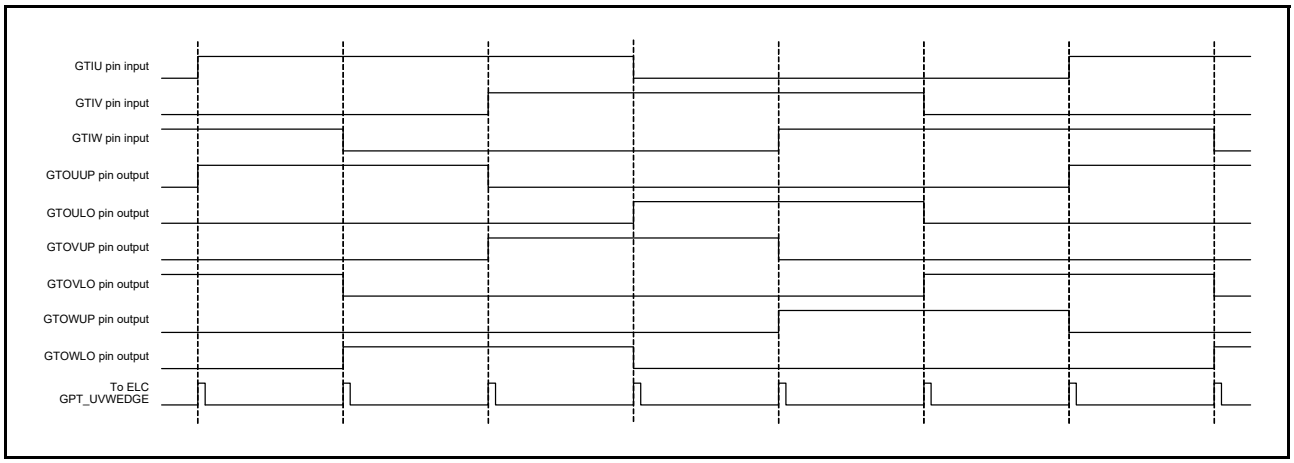


Figure 24.160 Example of OPS Level Output Operation (Forward Rotation) (FB = 0, RV = 0, P = 0, N = 0, INV = 0)

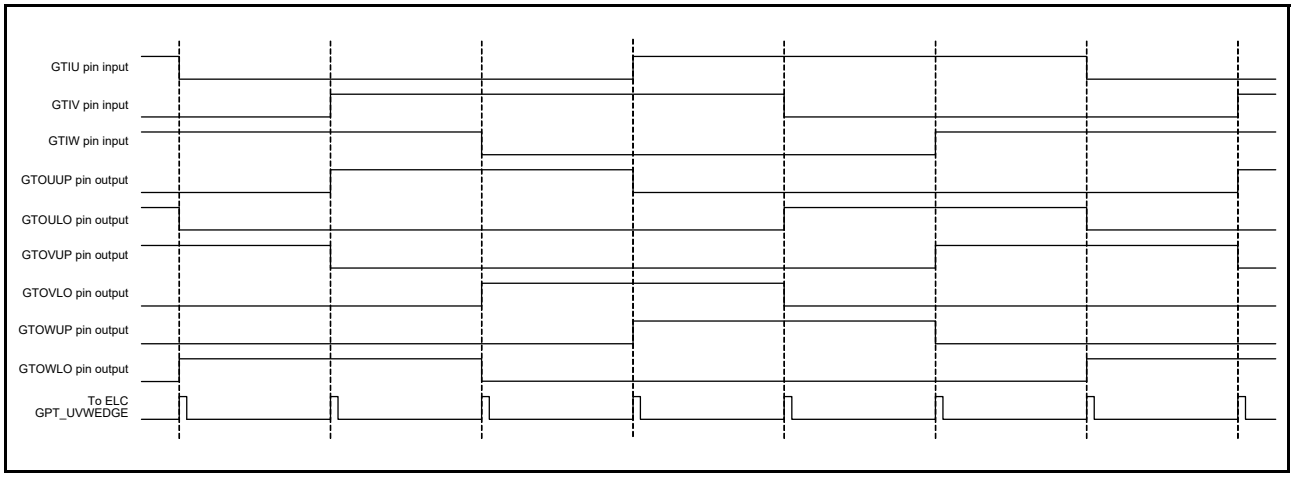


Figure 24.161 Example of OPS Level Output Operation (Reverse Rotation) (FB = 0, RV = 1, P = 0, N = 0, INV = 0)

Figure 24.162 and Figure 24.163 show examples of OPS chopped output operation.

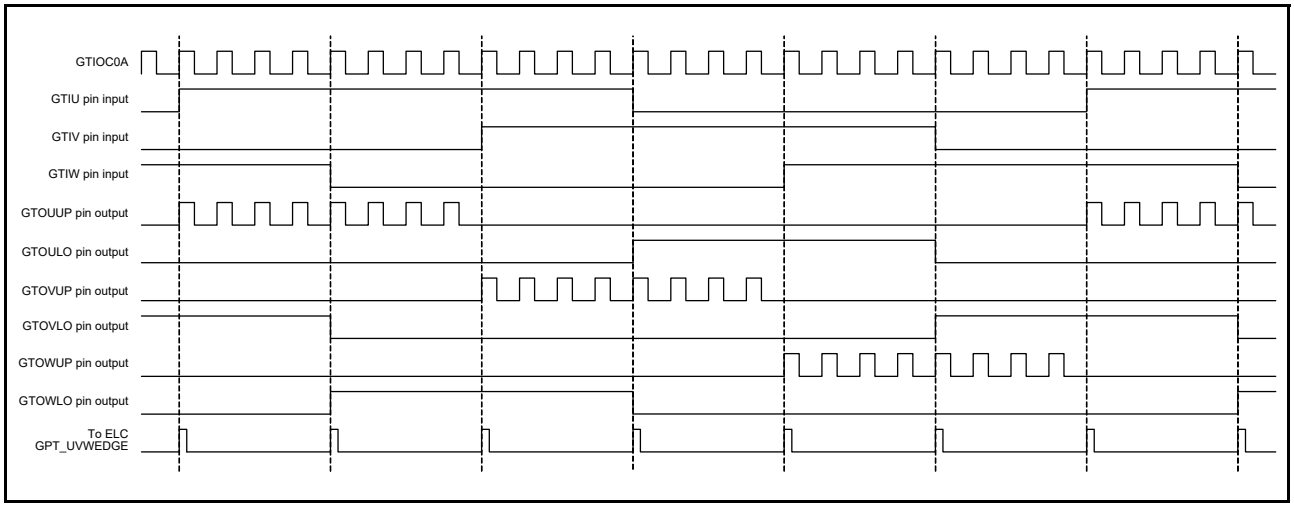


Figure 24.162 Example of OPS Chopped Output Operation (Positive Phase 120-degree) (FB = 0, RV = 0, P = 1, N = 0, INV = 0)

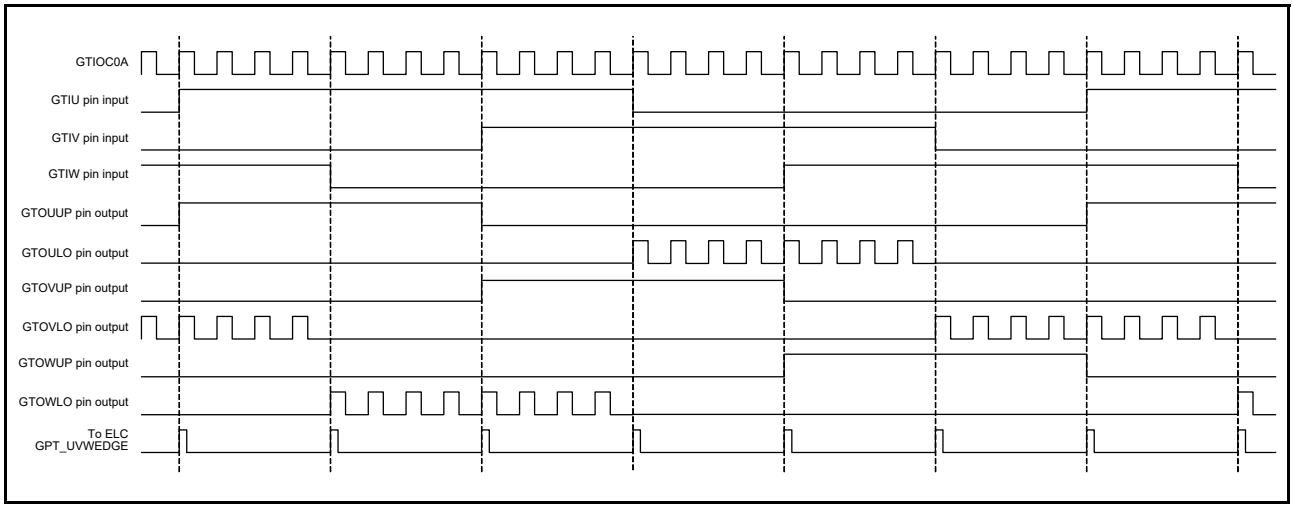


Figure 24.163 Example of OPS Chopped Output Operation (Negative Phase 120-degree) (FB = 0, RV = 0, P = 0, N = 1, INV = 0)

Figure 24.164 shows an example of OPS output disable control operation.

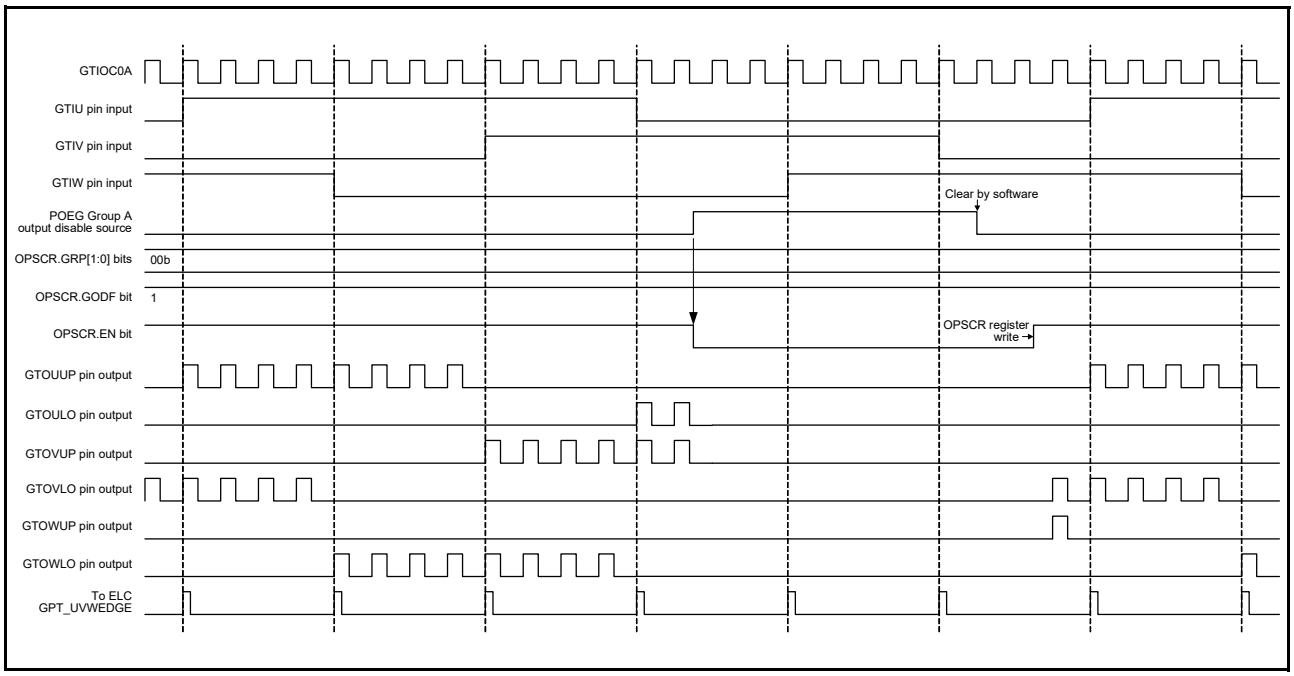


Figure 24.164 Example of OPS Output Disable Control Operation (FB = 0, RV = 0, P = 1, N = 1, INV = 0)

24.3.13.1 Input Selection and Sampling

The FB bit selects either the software setting value or external input for the input signal.
When the FB bit is 0, the GTIU, GTIV, GTIW external input are selected for the input signal to OPS after synchronization with the GPT core clock (PCLKC) and noise filtering.
When the FB bit is 1, the software setting value (UF, VF, WF bits) are selected for the input signal to OPS.
The selected input signals are sampled by the method selected by the ALIGN bit, and they are treated as input phase of OPS.
When the ALIGN bit is 0, the input signals are sampled by PCLKC.
When the ALIGN bit is 1, the input signals are sampled by the falling edge of GTIOC0A pin output.
The signals after sampling can be read by the U, V, W bits.
Table 24.34 shows the input selection by the FB bit and sampling method by the ALIGN bit.

Table 24.34 Input Selection and Sampling Method

OPSCR Register		Input Selection Sampling Method	Interrupt Phase (OPS internal signal)
FB bit	ALIGN bit		
0	0	GTIU, GTIV, GTIW external input PCLKC sampling	Input U-phase (gtu_sync) Input V-phase (gtv_sync) Input W-phase (gtw_sync)
	1	GTIU, GTIV, GTIW external input GTIOC0A falling edge sampling	
1	0	Software setting value UF, VF, WF bit PCLKC sampling	
	1	Software setting value UF, VF, WF bit GTIOC0A falling edge sampling	

24.3.13.2 Rotation Direction Control

When the rotation direction is reverse (RV = 1), the input phase is inverted.

24.3.13.3 Input Phase Decode

The 6-phase signals by decoding input phase after rotation direction control are generated.
Table 24.35 and Table 24.36 show the decode tables of input phase to rotate motor in forward (RV = 0) and reverse (RV = 1).

Table 24.35 The Decode Table of Input Phase (Forward Rotation)

Input Phase			**Input Phase after Rotation Direction Control**			**6-Phase Signals**					
U-phase	V-phase	W-phase	U-phase	V-phase	W-phase	Positive U-phase	Negative U-phase	Positive V-phase	Negative V-phase	Positive W-phase	Negative W-phase
gtu_sync	**gtv_sync**	**gtw_sync**	**gtu_sync_rv**	**gtv_sync_rv**	**gtw_sync_rv**	**gtuup_en**	**gtulo_en**	**gtvup_en**	**gtvlo_en**	**gtwup_en**	**gtwlo_en**
1	0	1	1	0	1	1	0	0	1	0	0
1	0	0	1	0	0	1	0	0	0	0	1
1	1	0	1	1	0	0	0	1	0	0	1
0	1	0	0	1	0	0	1	1	0	0	0
0	1	1	0	1	1	0	1	0	0	1	0
0	0	1	0	0	1	0	0	0	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	0	0	0	0	0	0

Table 24.36 The Decode Table of Input Phase (Reverse Rotation)

Input Phase			**Input Phase after Rotation Direction Control**			**6-Phase Signals**					
U-phase	V-phase	W-phase	U-phase	V-phase	W-phase	Positive U-phase	Negative U-phase	Positive V-phase	Negative V-phase	Positive W-phase	Negative W-phase
gtu_sync	**gtv_sync**	**gtw_sync**	**gtu_sync_rv**	**gtv_sync_rv**	**gtw_sync_rv**	**gtuup_en**	**gtulo_en**	**gtvup_en**	**gtvlo_en**	**gtwup_en**	**gtwlo_en**
1	0	1	0	1	0	0	1	1	0	0	0
1	0	0	0	1	1	0	1	0	0	1	0
1	1	0	0	0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	0	0	1	0	0
0	1	1	1	0	0	1	0	0	0	0	1
0	0	1	1	1	0	0	0	1	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	0	0	0	0	0	0

24.3.13.4 Output Selection Control

The EN, P, N, INV bits select the output wave.
The EN bit enables output of the 6-phase output. When the EN bit is 1, output of the 6-phase output is enabled. When the EN bit is 0, the external pin output is Hi-Z.
The P, N bits select whether chopping positive and negative phase are performed or not. When P, N bits are 1, chopping is performed by GTIOC0A pin output.
When the chopping is performed, there are cases where the PWM width of output is shorter than the width of the PWM used to chop just before or after switching of output phase, depending on the phase difference between the phase output switch timing and the phase of PWM.
The INV bit selects the polarity (either positive logic or negative logic) of phase output.
Table 24.37 and Table 24.38 show the output selection control method for positive and negative phase output.

Table 24.37 Output Selection Control Method (Positive Phase)

EN bit	P bit	INV bit	GTOnUP
0	—	—	0 (External pin output Hi-z)
1	0	0	Positive logic level output (gtmup_en)
1	0	1	Negative logic level output (~gtmup_en)
1	1	0	Positive logic chopped output (GTIOC0A & gumup_en)
1	1	1	Negative logic chopped output (~(GTIOC0A & gumup_en))

n = U, V, W
m = u, v, w

Table 24.38 Output Selection Control Method (Negative Phase)

EN bit	P bit	INV bit	GTOnLO
0	—	—	0 (External pin output Hi-z)
1	0	0	Positive logic level output (gtmlo_en)
1	0	1	Negative logic level output (~gtmlo_en)
1	1	0	Positive logic chopped output (GTIOC0A & gumlo_en)
1	1	1	Negative logic chopped output (~(GTIOC0A & gumlo_en))

n = U, V, W
m = u, v, w

24.3.13.5 Output Selection Control (Group Output Disable Function)

When the GODF bit is 1 and signal value selected by the GRP[1:0] bits is High (Output Disable Request), OPS's output pins are changed to Hi-Z asynchronously and the OPSCR.EN bit is cleared by the output disable request signal synchronized with PCLKC.
For the return, after clearing the Output Disable Request by software, set the EN bit to 1.
The timing of EN bit cleared is 3 PCLKC cycles after generating the output disable request. In order to perform the output disable control surely, the output disable request flag in POEG should be cleared in the timing that terminating the output disable request is at least 4 PCLKC cycles after generating the output disable request.
The example of the operation of the group output disable control, see the above-mentioned Figure 24.164.

24.3.13.6 Event Link Controller (ELC) Output

The logical sum of the pulse detected by rising and falling edge of U, V, W phase input is output to the event link controller (ELC). When the high level period of input phase is short, there are cases that the detected edge is not transmitted to the ELC correctly because of the logical sum.

24.3.13.7 OPS Start Operation Setting Flow

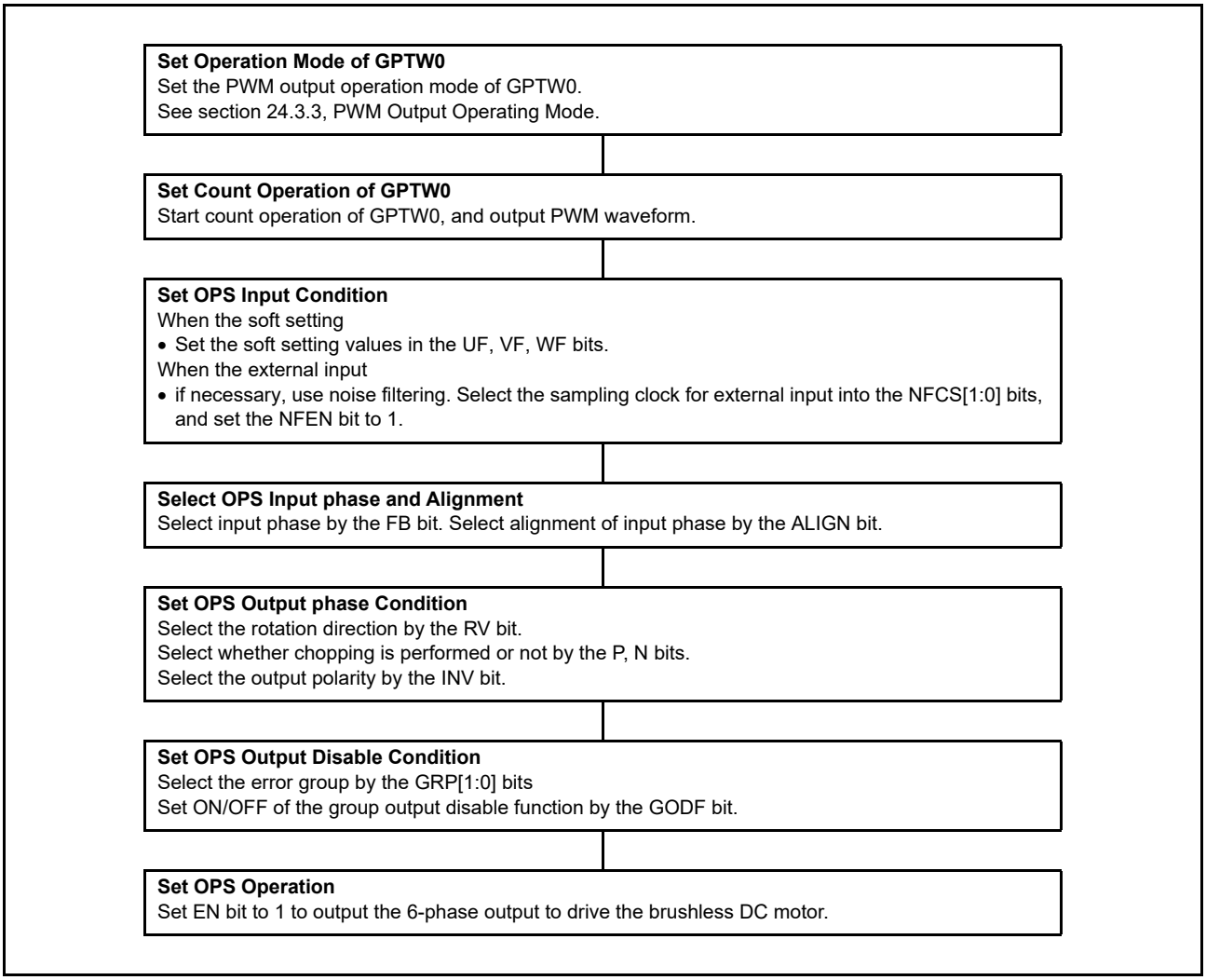


Figure 24.165 Example for Setting OPS Start Operation

24.3.14 Inter Channel Logical Operation Function

The logical operation function between compare match outputs can be performed.
Figure 24.166 shows the block diagram of inter channel logical operation.
To prevent hazard to the this function output, the signal after logical operation is latched with PCLKC. After latching, the output disable control is performed.
When the logical operation function which causes the delay of 1 PCLKC is selected, the output enable signal is also delayed with 1 PCLKC and input to the output disable control. The execution of a logical operation does not delay the interrupts, A/D conversion start requests, and event signal output to the ELC.
When the same signal (C = A or D = B) to operate logical function AND, OR, EXOR and NOR is selected, C or D is treated as 1. In the case of GTIOCnA pin output, when A of same channel is selected for C, the result of AND is A, the result of OR is 1, the result of EXOR is NOT A, and the result of NOR is 0.

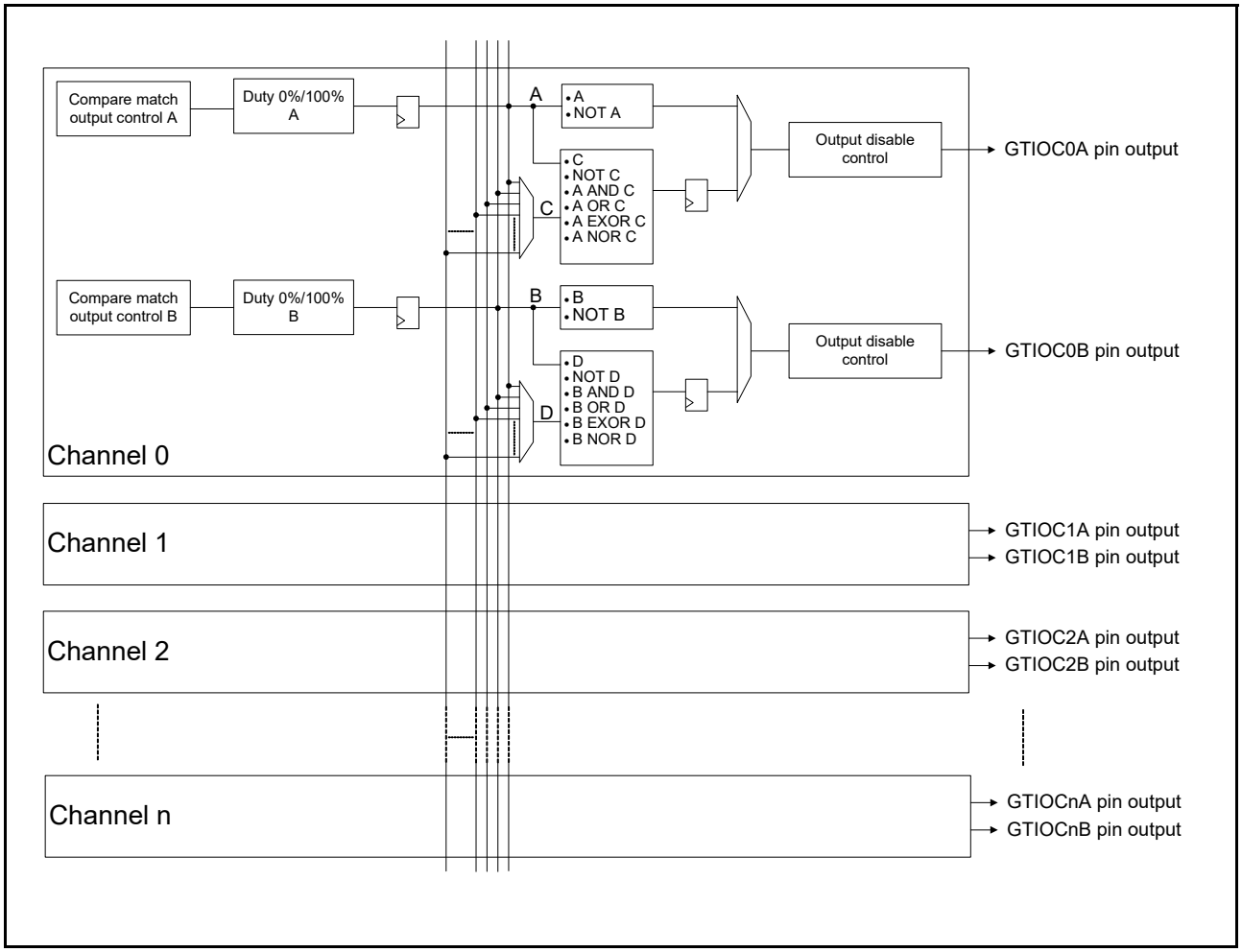


Figure 24.166 Block Diagram of Inter Channel Logical Operation

Figure 24.167 shows an example of inter channel logical operation.

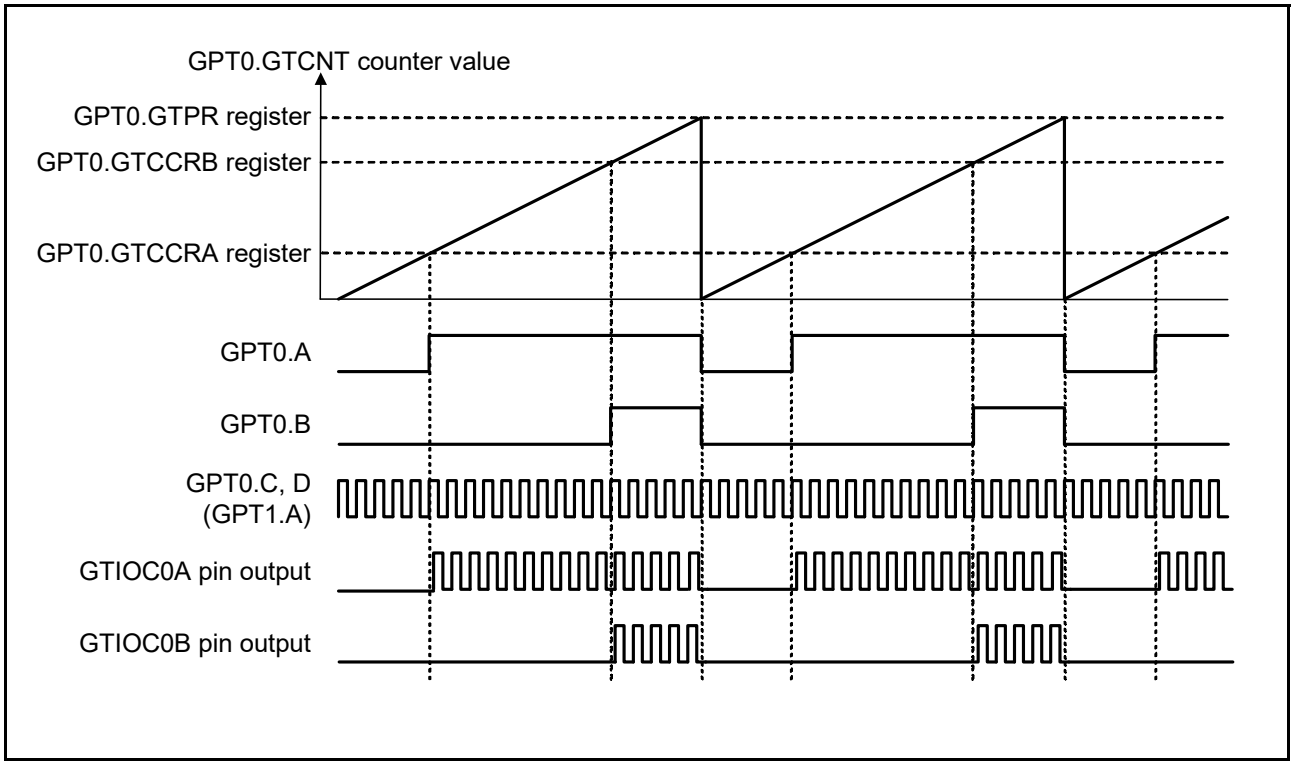


Figure 24.167 Example of Inter Channel Logical Operation (When GPTW0.GTICLF.ICLFA[2:0] = 100b, ICLFB[2:0] = 100b, GPTW0.GTICLF.ICLFSELC[5:0] = 000010b, ICLFSELD[5:0] = 000010b)

24.4 Interrupt Sources

24.4.1 Interrupt Sources and Priorities

There are three kinds of interrupt sources; GTCCRm register (m = A to F) input capture/compare match, GTCNT counter overflow (GTPR register compare match)/underflow, and dead time error.
Each interrupt source has its own status flag and control bit for generating interrupt request signals, where the generation of interrupt request signals can be enabled or disabled individually.
When an interrupt source condition is satisfied, an interrupt request is generated if the corresponding interrupt request enable or disable bit is 1.
For details, refer to section 14, Interrupt Controller (ICUG). Table 24.39 shows a list of GPTW interrupt sources.

Table 24.39 GPTW Interrupt Sources (n = 0 to 7)

Channel	Name	Interrupt Source	Interrupt Flag
GPTWn	GTCIAn	GTCCRA register input capture/compare match	GPTWn.GTST.TCFA
	GTCIBn	GTCCRB register input capture/compare match	GPTWn.GTST.TCFB
	GTCICn	GTCCRC register compare match	GPTWn.GTST.TCFC
	GTCIDn	GTCCRD register compare match	GPTWn.GTST.TCFD
	GDTEn	Dead time error	GPTWn.GTST.DTEF
	GTCEIn	Cycle count end	GPTWn.GTST.PCF
	GTCIEn	GTCCRE register compare match	GPTWn.GTST.TCFE
	GTCIFn	GTCCRF register compare match	GPTWn.GTST.TCFF
	GTCIVn	GTCNT counter overflow (GTPR register compare match)	GPTWn.GTST.TCFPO
	GTCIUn	GTCNT counter underflow	GPTWn.GTST.TCFPU

(1) GTCIAn Interrupt (n = 0 to 7)

When the GTINTAD.GTINTA bit is 1, an interrupt request is generated under the following conditions.

- When the GTCCRA register functions as a compare match register, the GTCNT counter value (in complementary PWM mode, the GTCNT counter value of master channel) matches with the GTCCRA register.
- When the GTCCRA register functions as an input capture register, the input-capture signal has caused transfer of the GTCNT counter value to the GTCCRA register. In complementary PWM mode, GTCCRA register does not function as an input capture register.

(2) GTCIBn Interrupt (n = 0 to 7)

When the GTINTAD.GTINTB bit is 1, an interrupt request is generated under the following conditions.

- When the GTCCRB register functions as a compare match register, the GTCNT counter value matches with the GTCCRB register. In complementary PWM mode, GTCCRB register does not function as a compare match register.
- When the GTCCRB register functions as an input capture register, the input-capture signal has caused transfer of the GTCNT counter value to the GTCCRB register. In complementary PWM mode, GTCCRB register does not function as an input capture register.

(3) GTCICn Interrupt (n = 0 to 7)

When the GTINTAD.GTINTC bit is 1, an interrupt request is generated under the following condition.

- When the GTCCRC register functions as a compare match register, the GTCNT counter value (in complementary PWM mode, the GTCNT counter value of master channel) matches with the GTCCRC register.

A compare match is not performed and thus interrupt is not requested under the following conditions.

- GTCR.MD[2:0] bits = 001b, GTCR.MD[3:0] bits = 0001b (sawtooth-wave one-shot pulse mode)
- GTCR.MD[2:0] bits = 110b, GTCR.MD[3:0] bits = 0110b (triangle-wave PWM mode 3)
- GTBER.CCRA[1:0] bits = 01b, 10b, 11b (buffer operation with the GTCCRC register)

(4) GTCIDn Interrupt (n = 0 to 7)

When the GTINTAD.GTINTD bit is 1, an interrupt request is generated under the following condition.

- When the GTCCRD register functions as a compare match register, the GTCNT counter value (in complementary PWM mode, the GTCNT counter value of master channel) matches with the GTCCRD register.

A compare match is not performed and thus interrupt is not requested under the following conditions.

- GTCR.MD[2:0] bits = 001b, GTCR.MD[3:0] bits = 0001b (sawtooth-wave one-shot pulse mode)
- GTCR.MD[2:0] bits = 110b, GTCR.MD[3:0] bits = 0110b (triangle-wave PWM mode 3)
- GTBER.CCRA[1:0] bits = 10b, 11b (buffer operation with the GTCCRD register)

(5) GTCIEn Interrupt (n = 0 to 7)

When the GTINTAD.GTINTE bit is 1, an interrupt request is generated under the following condition.

- When the GTCCRE register functions as a compare match register, the GTCNT counter value (in complementary PWM mode, the GTCNT counter value of master channel) matches with the GTCCRE register.

A compare match is not performed and thus interrupt is not requested under the following conditions.

- GTCR.MD[2:0] bits = 001b, GTCR.MD[3:0] bits = 0001b (sawtooth-wave one-shot pulse mode)
- GTCR.MD[2:0] bits = 110b, GTCR.MD[3:0] bits = 0110b (triangle-wave PWM mode 3)
- GTBER.CCRB[1:0] bits = 01b, 10b, 11b (buffer operation with the GTCCRE register)

(6) GTCIFn Interrupt (n = 0 to 7)

When the GTINTAD.GTINTF bit is 1, an interrupt request is generated under the following condition.

- When the GTCCRF register functions as a compare match register, the GTCNT counter value (in complementary PWM mode, the GTCNT counter value of master channel) matches with the GTCCRF register.

A compare match is not performed and thus interrupt is not requested under the following conditions.

- GTCR.MD[2:0] bits = 001b, GTCR.MD[3:0] bits = 0001b (sawtooth-wave one-shot pulse mode)
- GTCR.MD[2:0] bits = 110b, GTCR.MD[3:0] bits = 0110b (triangle-wave PWM mode 3)
- GTBER.CCRB[1:0] bits = 10b, 11b (buffer operation with the GTCCRF register)

(7) GTCIVn Interrupt (n = 0 to 7)

When the GTINTAD.GTINTPR[0] bit is 1, an interrupt request is generated under the following conditions.

- In sawtooth-wave mode, interrupt requests are enabled at overflows (the GTCNT counter value changes from the GTPR register value to 0 during up-counting).
- In sawtooth-wave PWM mode 1 and sawtooth-wave one-shot pulse mode, interrupt requests are enabled at overflows (when the GTCNT counter value changes from the GTPR register value to 0 during up-counting).
- In sawtooth-wave PWM mode 2, interrupt requests are enabled at overflows (the GTCNT counter value changes from GTCCRm (m = A to F) register value selected with GTCSR.CSCMSC[2:0] bits to 0000 0000h) or the time when GTCNT counter value matches GTPR register value.
- In triangle-wave mode, interrupt requests are enabled at crests (the GTCNT counter value changes from the GTPR register value to the GTPR register value minus 1).

- In complementary PWM mode, interrupt requests are enabled at crests (the GTCNT counter value of master channel changes from the GTPR register value to the GTPR register value minus 1).
- In counting driven by a hardware source, including counting for the external pulse width measuring function, interrupt requests are enabled on overflows (that is, when the GTCNT counter value changes from the GTPR register value to 0 during up-counting).

(8) GTCIUn Interrupt (n = 0 to 7)

When the GTINTAD.GTINTPR[1] bit is 1, an interrupt request is generated under the following conditions.

- In sawtooth-wave mode, interrupt requests are enabled at underflows (the GTCNT counter value changes from 0 to the GTPR register value during down-counting).
- In sawtooth-wave PWM mode 1 and sawtooth-wave one-shot pulse mode, interrupt requests are enabled at overflows (when the GTCNT counter value changes from the GTPR register to 0 during down-counting).
- In triangle-wave mode, interrupt requests are enabled at troughs (the GTCNT counter value changes from 0 to 1).
- In complementary PWM mode, interrupt requests are enabled at troughs (the GTCNT counter value of master channel changes from 0 to 1).
- In count operation by a hardware source, interrupt requests are enabled at underflows (the GTCNT counter value changes from 0 to the GTPR register value during down-counting).

(9) GDTEn Interrupt (n = 0 to 7)

When automatic dead time setting has been made, the GTST.DTEF flag becomes 1 when the timer output toggle point with dead time added exceeds the count period. If the GTINTAD.GRPDTE bit is 1 at this time, a dead time error interrupt request (GDTE) is generated.
In addition, when the timer output toggle point with dead time added is back within the count period, the GTST.DTEF flag changes from 1 to 0.

(10) GTCEIn Interrupt (n = 0 to 7)

When the GTPC.PCEN bit is 1 and the GTPC.PCNT counter is 1, and the end of the cycle is reached, if the GTINTAD.GTINTPC bit is 1, a cycle count end interrupt (GTCEIn) request is generated.

Table 24.40 Relationship between Interrupt Signals and Interrupt Enable Bits (n = 0 to 7)

Interrupt Signal	Interrupt Enable Bit	Status Flag
GTCIAn	GTINTAD.GTINTA bit	GTST.TCFA Flag
GTCIBn	GTINTAD.GTINTB bit	GTST.TCFB Flag
GTCICn	GTINTAD.GTINTC bit	GTST.TCFC Flag
GTCIDn	GTINTAD.GTINTD bit	GTST.TCFD Flag
GDTEn	GTINTAD.GRPDTE	GTST.DTEF Flag
GTCEIn	GTINTAD.GTINTPC	GTST.PCF Flag
GTCIEn	GTINTAD.GTINTE bit	GTST.TCFE Flag
GTCIFn	GTINTAD.GTINTF bit	GTST.TCFF Flag
GTCIVn	GTINTAD.GTINTPR[1:0] bits	GTST.TCFPO Flag
GTCIUn		GTST.TCFPU Flag

24.4.2 DMAC/DTC Activation

The DMAC and DTC can be triggered by the interrupt request in each channel. For details, see section 14, Interrupt Controller (ICUG), section 17, DMA Controller (DMACAa), section 18, Data Transfer Controller (DTCb).

24.4.3 Interrupt and A/D Conversion Start Request Skipping Function

24.4.3.1 Interrupt Skipping Function by GTITC Register

By setting the GTITC register, the GTCNT counter overflow (GTPR register compare match) interrupt (GTCIV) and underflow interrupt (GTCIU) can be skipped. Other interrupts and A/D conversion start request signals can be skipped in coordination with the GTCIV/GTCIU skipping function. However, the dead time error interrupts cannot be linked with the GTCIV/GTCIU skipping function.

When the interrupt is skipped, the updating of relevant status flag is also skipped. The interrupt skipping is continued even if the status flag is set to 1.

The interrupt skipping function is related only to the GTITC register setting, and is not related to setting of the GTINTAD register interrupt enable bit. The interrupt enable bit is used only to control the output of the interrupt signal after thinning out.

When both troughs and crests are counted and skipped in triangle-wave mode, if the number of times of skipping is odd, GTCIV/GTCIU interrupt requests cannot be generated at troughs only or at crests only depending on the skipping counter start timing. Therefore, in order to count both troughs and crests and generate the GTCIV/GTCIU interrupts at troughs only or crests only in triangle-wave mode, the number of times of skipping should be even.

Similarly, in sawtooth-wave mode, when both overflows and underflows are counted and skipped with the count direction changed, GTCIV/GTCIU interrupt requests cannot be generated at overflows only or at underflows only. Therefore, in order to count both overflows and underflows with the count direction changed and generate the GTCIV/GTCIU interrupts at overflows only or underflows only in sawtooth-wave mode, the skipping state should be carefully checked before using.

When changing the skipping count, be sure to release the skipping count setting (GTITC.IVTC[1:0] bits = 00b).

Figure 24.168 to Figure 24.173 show examples of skipping function operation.

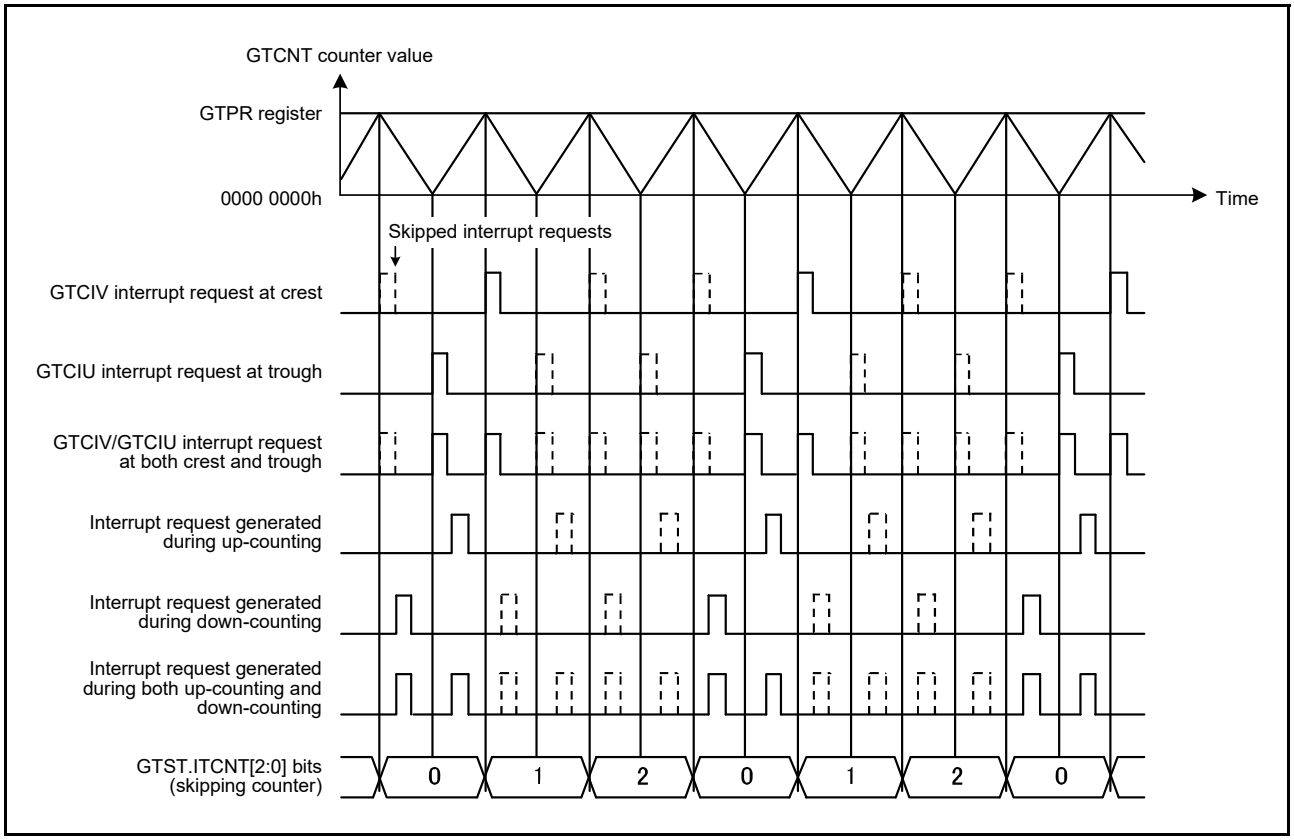


Figure 24.168 Example of Interrupt Skipping Function Operation (Triangle Waves, Counting and Skipping Crests, Skipping Count: 2)

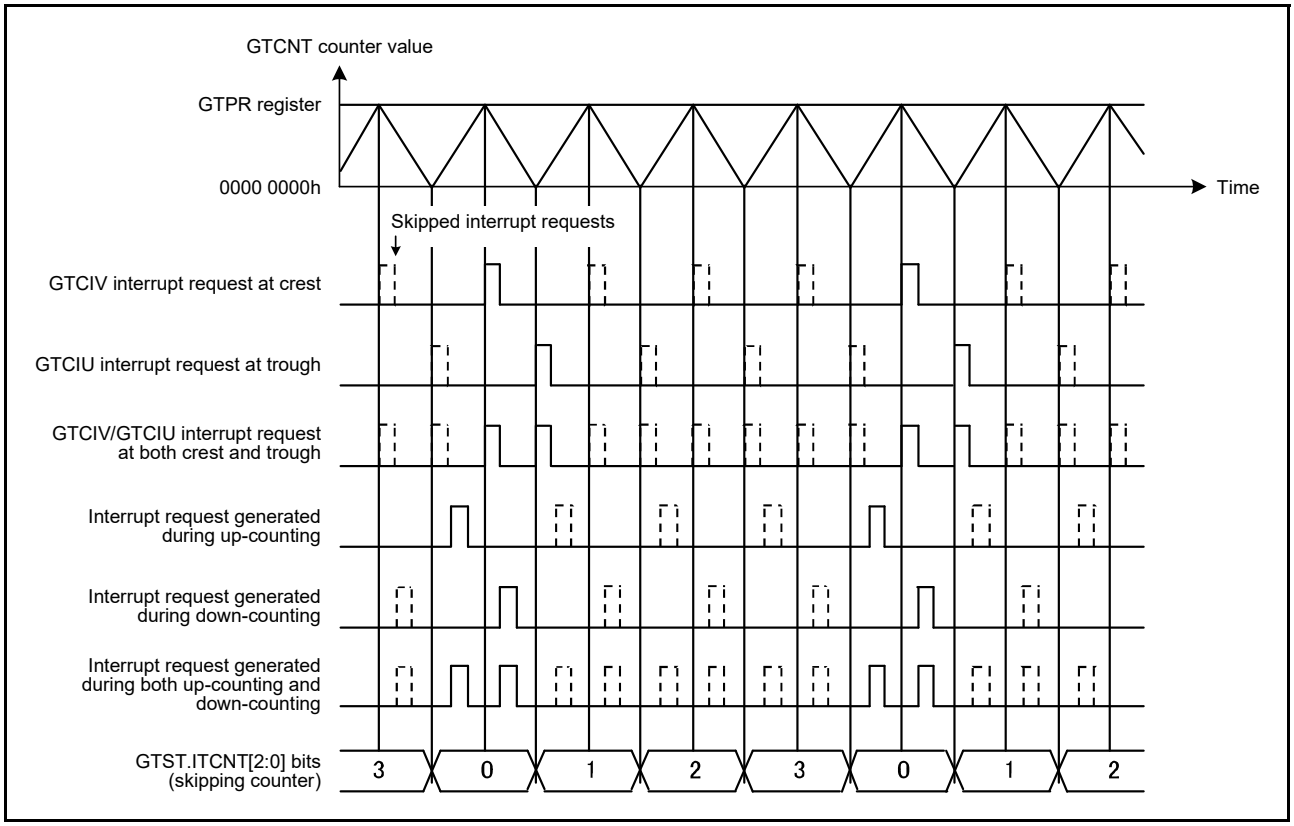


Figure 24.169 Example of Interrupt Skipping Function Operation (Triangle Waves, Counting and Skipping Troughs, Skipping Count: 3)

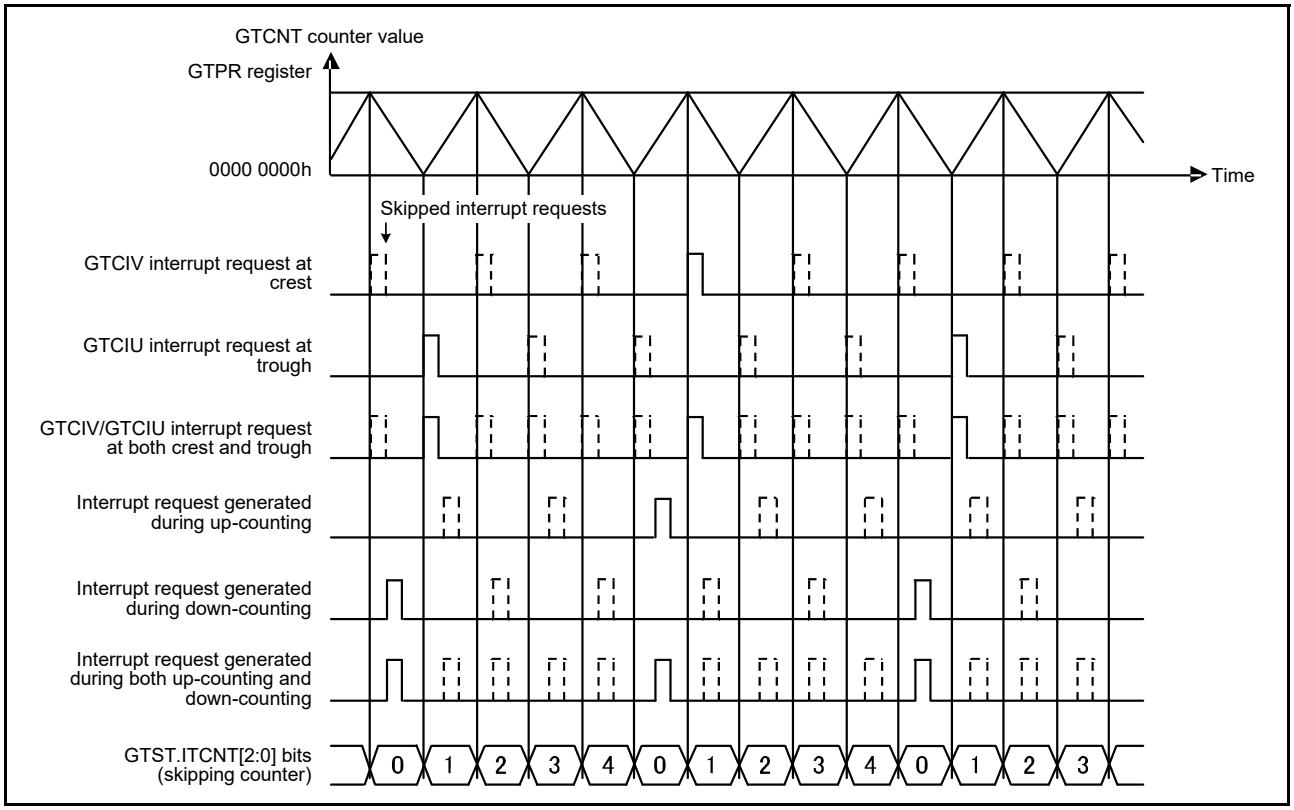


Figure 24.170 Example of Interrupt Skipping Function Operation (Triangle Waves, Counting and Skipping Both Troughs and Crests, Skipping Count: 4)

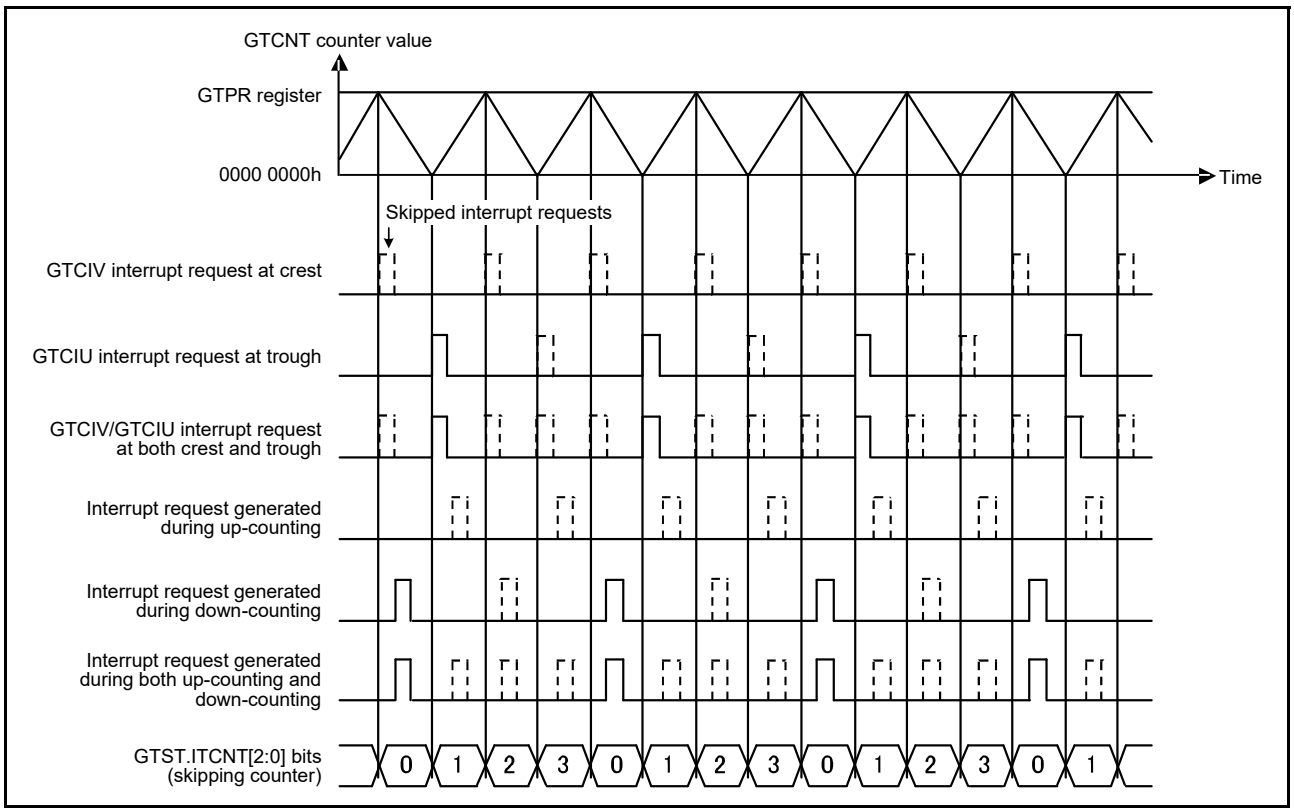


Figure 24.171 Example of Interrupt Skipping Function Operation (Triangle Waves, Counting and Skipping Both Troughs and Crests, Skipping Count: 3, Skipping Started at Up-Counting)

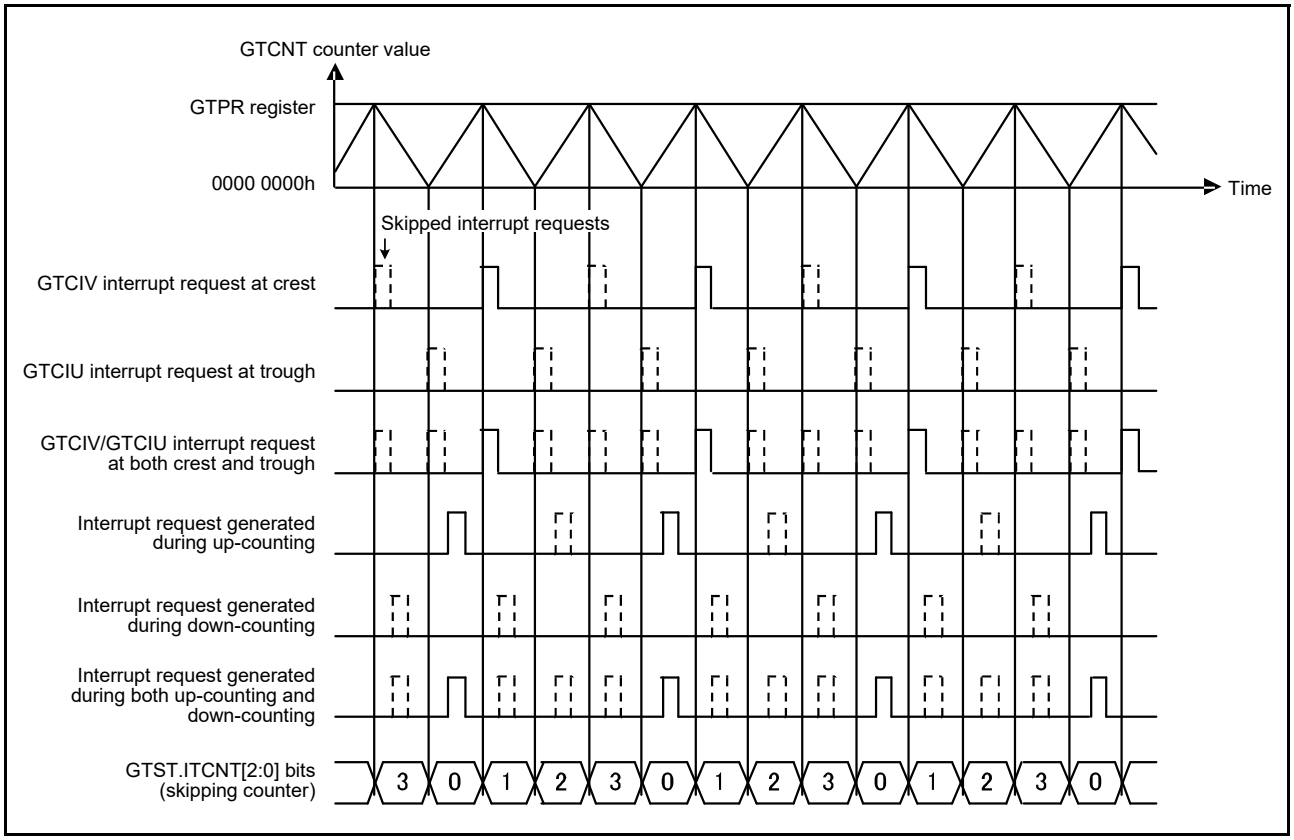


Figure 24.172 Example of Interrupt Skipping Function Operation (Triangle Waves, Counting and Skipping Both Troughs and Crests, Skipping Count: 3, Skipping Started at Down-Counting)

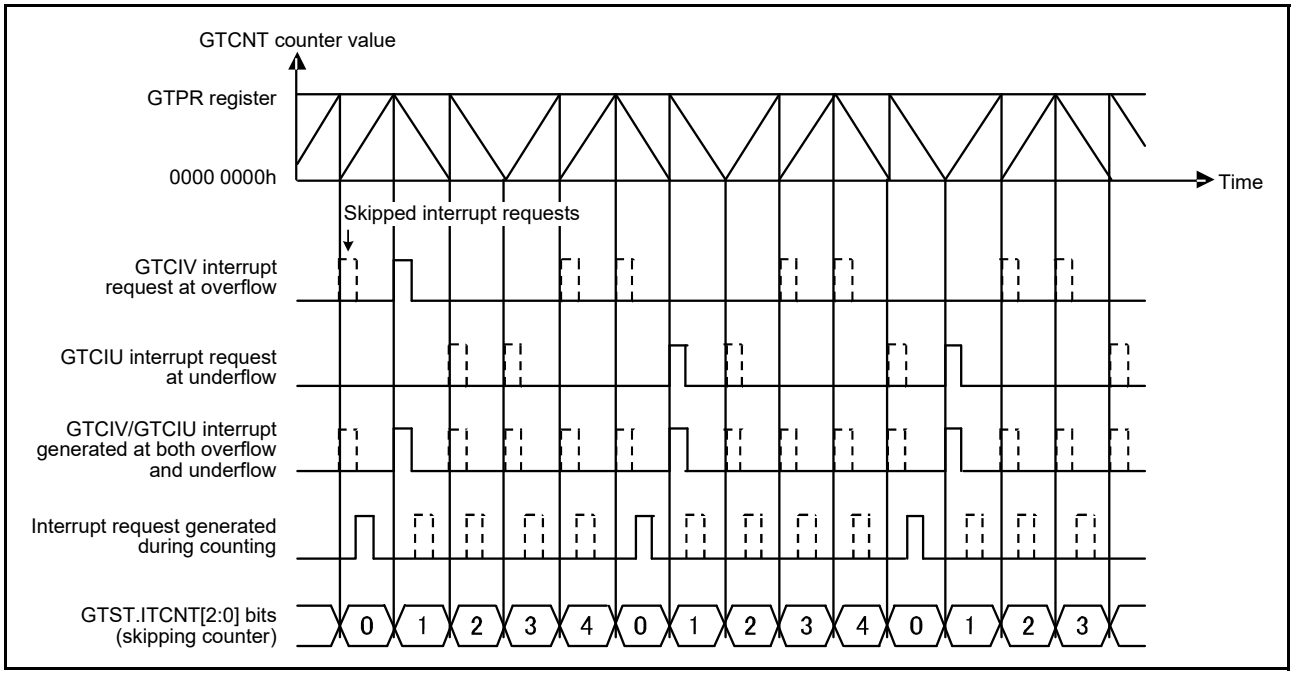


Figure 24.173 Example of Interrupt Skipping Function Operation (Sawtooth Waves, Operation with Count Direction Changed, Counting and Skipping Both Overflows and Underflows, Skipping Count: 4)

24.4.3.2 Extended Interrupt Skipping Function

Overflow/underflow interrupt, compare match/input capture interrupt, A/D conversion start request, and buffer transfer can be skipped by counting the GTCNT counter overflow or underflow based on settings of the GTEITC, GTEITLI1, GTEITLI2 and GTEITLB registers. The dead time error interrupt cannot be skipped.

Overflow/underflow interrupt, compare match/input capture interrupt, A/D conversion start request, and buffer transfer can individually be set for skipping and the skipping period using the GTEITLI1, GTEITLI2, and GTEITLB registers. The skipping period is set, related to the operation of the two independent extended interrupt skipping counters (EITCNT1[3:0] and EITCNT2[3:0] bits in the GTEITC registers), as a period that either of the counter value for such skipping counters is other than 0 or other than the skipping count. The skipping period can also be set as a period that the both skipping counters are set as other than 0 or other than the skipping count for the counter value.

Figure 24.174 shows the counter operation for interrupt skipping by the GTITC register and extended interrupt skipping.

The counter operation for extended interrupt skipping is set by the GTEITC register.

The EITCNT1[3:0] bits count the count source (in Figure 24.174, a crest is selected) selected by the extended interrupt counter 1 count source select bit (EIVTC1[1:0] bits) when the initial value is set for 0, and repeat the count operation which returns to 0 when the skipping count (in Figure 24.174, count is 2) set by the extended interrupt skipping 1 skipping count setting bit (EIVTT1[3:0] bits) is achieved.

The EITCNT2[3:0] bits can set the initial value, count the count source (in Figure 24.174, a trough is selected) selected by the extended interrupt counter 2 count source select bit (EIVTC2[1:0] bits), and repeat the count operation which return to 0 when the skipping count (in Figure 24.174, count is 2) set by the EIVTT2[3:0] bits is achieved. The initial value is set when the GTEITC register is written by the access of upper 16 bits or 32 bits and when the written value to the EIVTC2[1:0] bits is other than 00b, while the setting for the extended interrupt skipping counter 2 is not in counting (EIVTC2[1:0] bits are 00b).

When the initial value is set, the written value to the extended interrupt skipping counter 2 initial value bit (EITCNT2IV[3:0] bits) is set as the initial value for the EITCNT2[3:0] bits.

The extended interrupt skipping counter starts up-counting at the first count clock after the setting is modified from not counting to counting.

The interrupt skipping count counter (GTST.ITCNT[2:0] bits) by the GTITC register is reset to 000b when the GTCNT counter operation is stopped: however, the EITCNT1[3:0] and EITCNT2[3:0] bits which are an extended interrupt skipping function retain the value even after the GTCNT counter operation is stopped, and the counting can be resumed from the value before the counter is stopped. When values for the EITCNT1[3:0] and EITCNT2[3:0] bits are to be reset (0000b), set the EIVTC1[1:0] and EIVTC2[1:0] bits to the setting for not counting (not skipping) (00b).

When the skipping count is to be changed, change the count after stopping the skipping counter operation (set either of the EIVTC1[1:0] bit or the EIVTC2[1:0] bit to 00b).

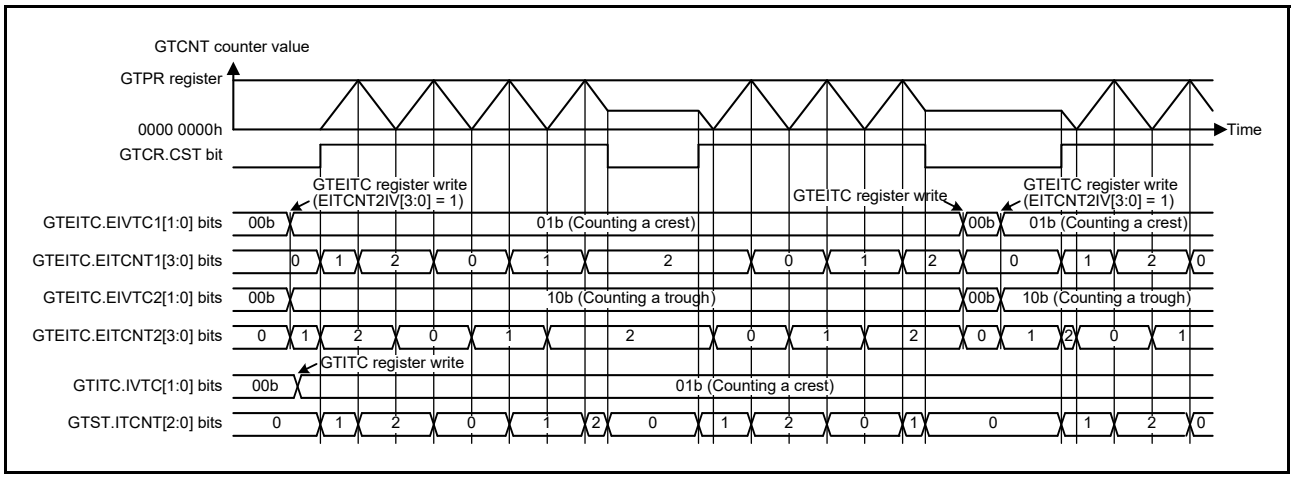


Figure 24.174 Counter Operation for Interrupt Skipping

Interrupt skipping by the GTEITLI1 register and A/D conversion start request skipping by the GTEITLI2 register can be performed simultaneously with interrupt skipping by the GTITC register or GTADCMSC register. A skipping period at this time is represented by ORed skipping periods of respective registers.
Figure 24.175 shows corresponding interrupt skipping operations by different registers are performed simultaneously.

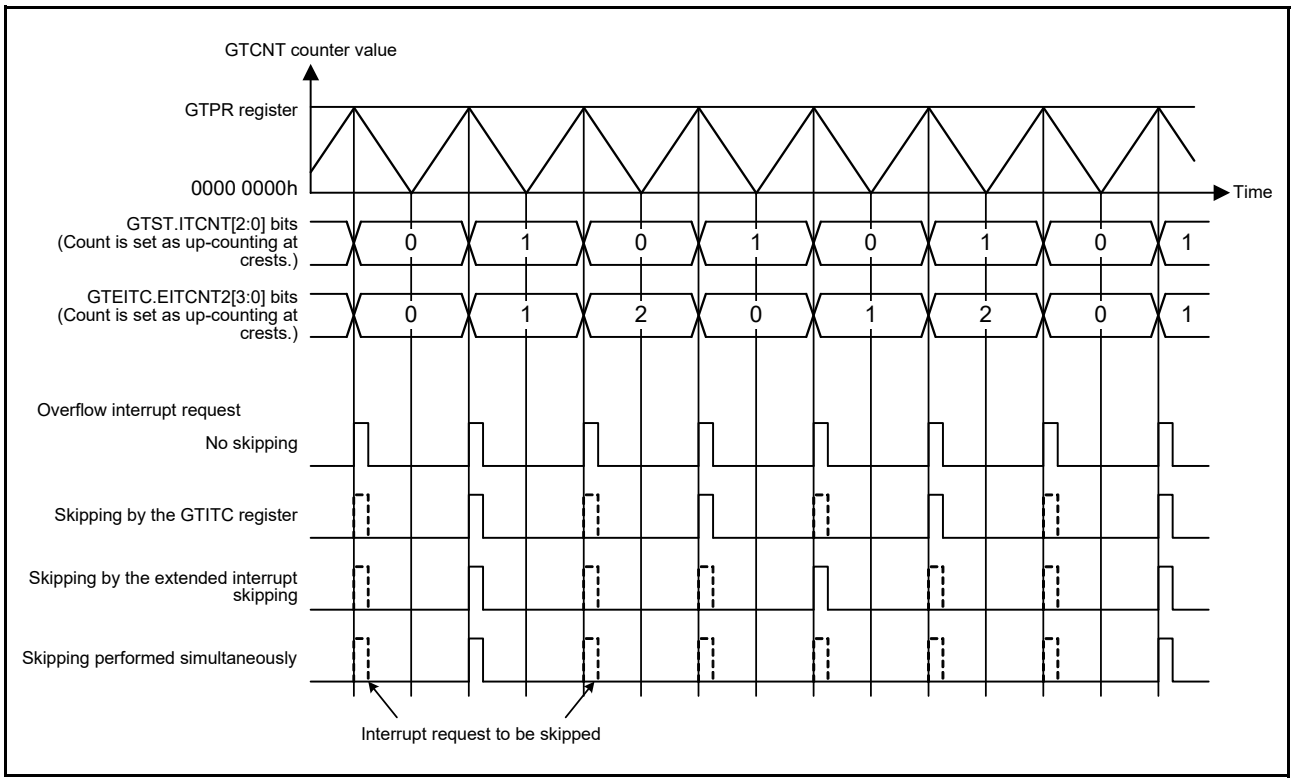


Figure 24.175 Example of Interrupt Skipping Operation (Skipping by the GTITC Register: Counting Crests, Extended Interrupt Skipping: EIVTC1[1:0] bits = 00b, EIVTC2[1:0] bits = 01b, EITLV[2:0] bits = 010b)

When the extended interrupt skipping selected by GTEITLI1 is performed, the relevant status flag is also skipped. The skipping function is continued even if the status flag is set to 1.
Skipping of updating of status flag, interrupt and ELC event source outputs corresponding to interrupts for which skipping can be set up in the GTEITLI1 register is only based on the settings of the GTITC register and the extended interrupt skipping register, and has no connection with the setting of the interrupt enable bit in the GTINTAD register. The interrupt enable bit is used only for the interrupt signal output after skipping.
When the A/D conversion start request skipping selected by GTEITLI2 is performed, the relevant status flag is also skipped. The skipping function is continued even if the status flag is set to 1.
Skipping of updating of status flag, interrupt and ELC event source outputs corresponding to A/D conversion start request for which skipping can be set up in the GTEITLI2 register is only based on the settings of the GTITC register and the extended interrupt skipping register, and has no connection with the setting of A/D conversion start request enable bit in the GTINTAD register. The A/D conversion start request enable bit is used only for A/D conversion start request output (interrupt and ELC event source output) after skipping.
A buffer transfer skipping by the GTEITLB register is performed in all of buffer operation which is enabled in the GTBER, GTDTCR, and GTBER2 registers, or all buffer operations performed by sawtooth-wave one-shot pulse mode, triangle-wave PWM mode 3, or complementary PWM mode (excludes buffer transfer from GTCCRC, GTCCRE to GTCCRA).
An interrupt skipping and a buffer transfer skipping are operated individually. An interrupt output without performing a buffer transferred performing a buffer transfer without an interrupt output can also be operated.
Figure 24.176 to Figure 24.183 show examples of skipping function operation.

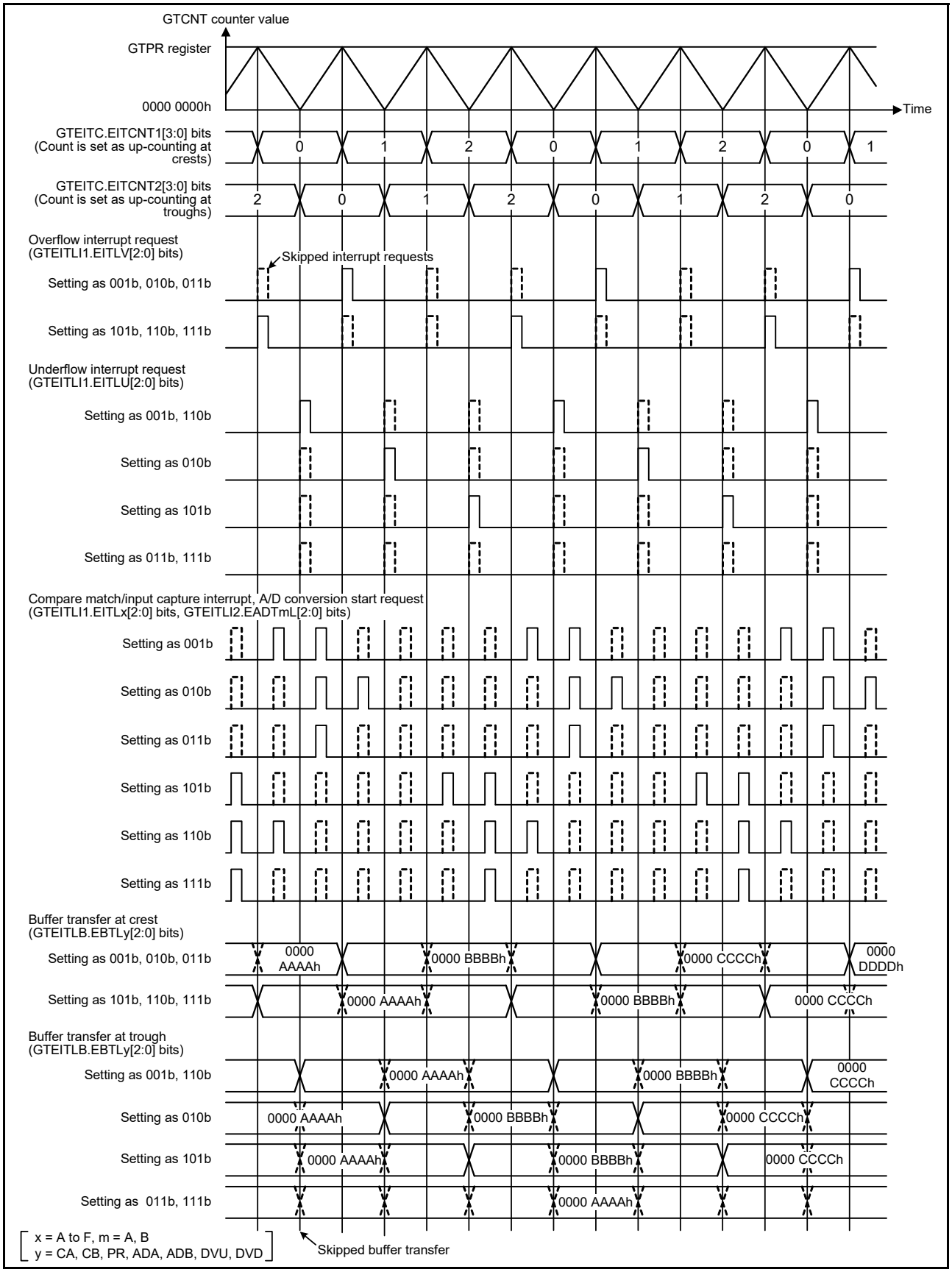


Figure 24.176 Example of Extended Interrupt Skipping Function (Triangle Waves, Counting Crests, Extended Interrupt Skipping 1 Skipping Count: 2, Counting Troughs, Extended Interrupt Skipping 2 Skipping Count: 2, Extended Interrupt Skipping Counter 2 Initial Value: 0)

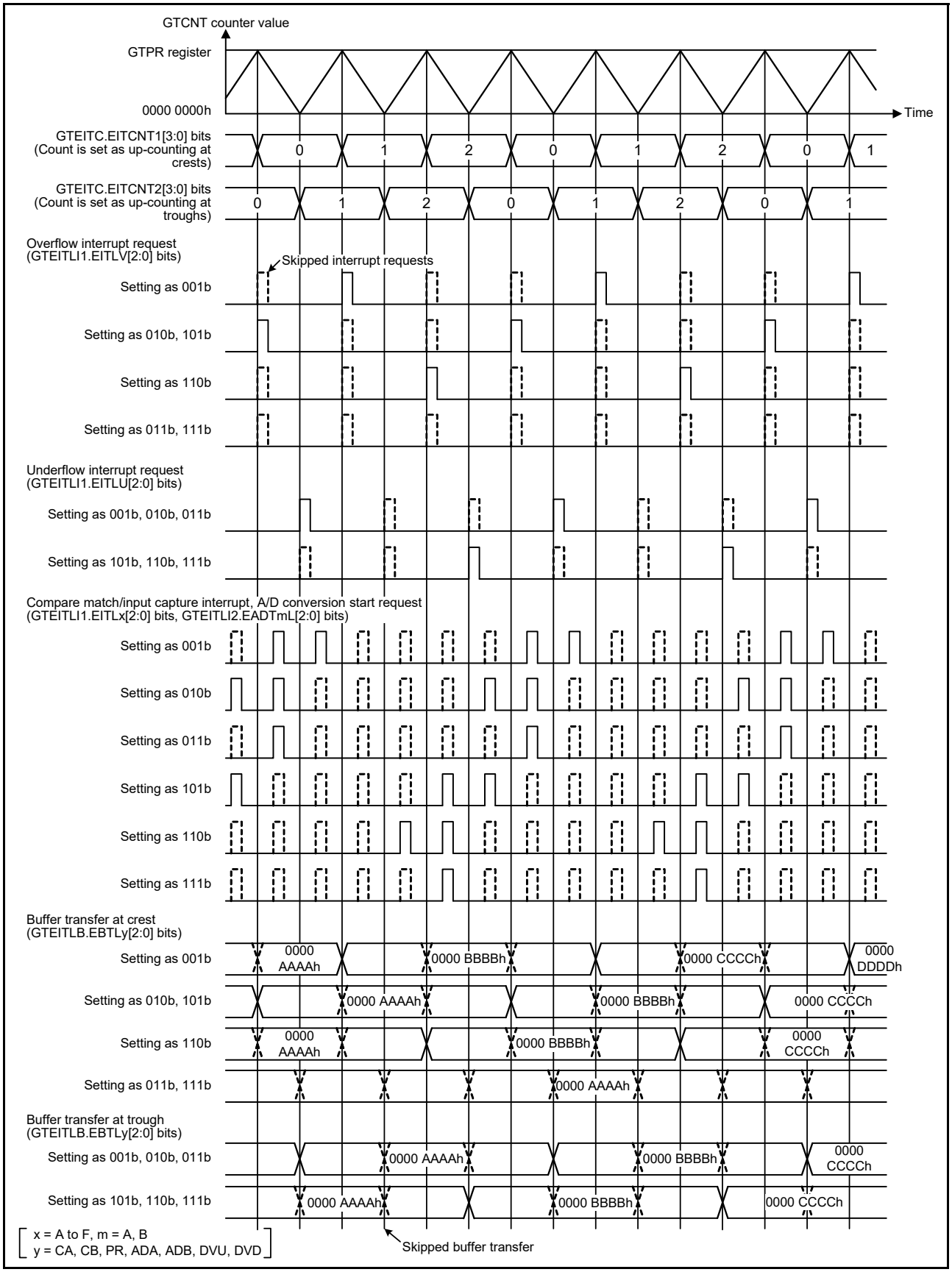


Figure 24.177 Example of Extended Interrupt Skipping Function (Triangle Waves, Count Crests by the Extended Interrupt Skipping 1 with Skipping Count: 2, Count Troughs by the Extended Interrupt Skipping 2 with Skipping Count: 2, Extended Interrupt Skipping Counter 2 Initial Value: 1)

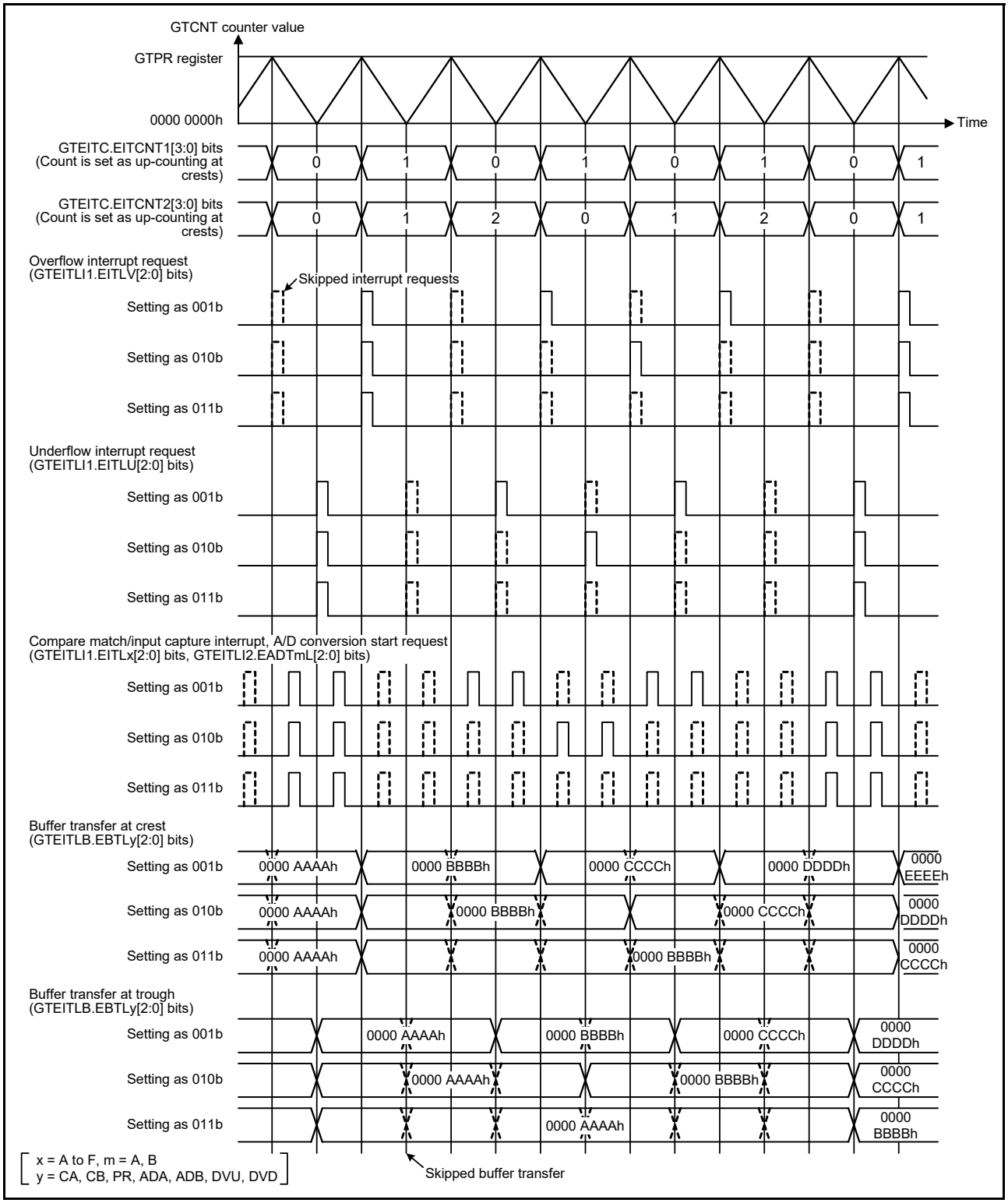


Figure 24.178 Example of Extended Interrupt Skipping Function Operation (Triangle Waves, Counting Crests, Extended Interrupt Skipping 1 Skipping Count: 1, Counting Crests, Extended Interrupt Skipping 2 Skipping Count: 2, Extended Interrupt Skipping Counter 2 Initial Value: 0, Skipped in the Period other than GTEITC.EITCNTk bits (k = 1, 2) as 0)

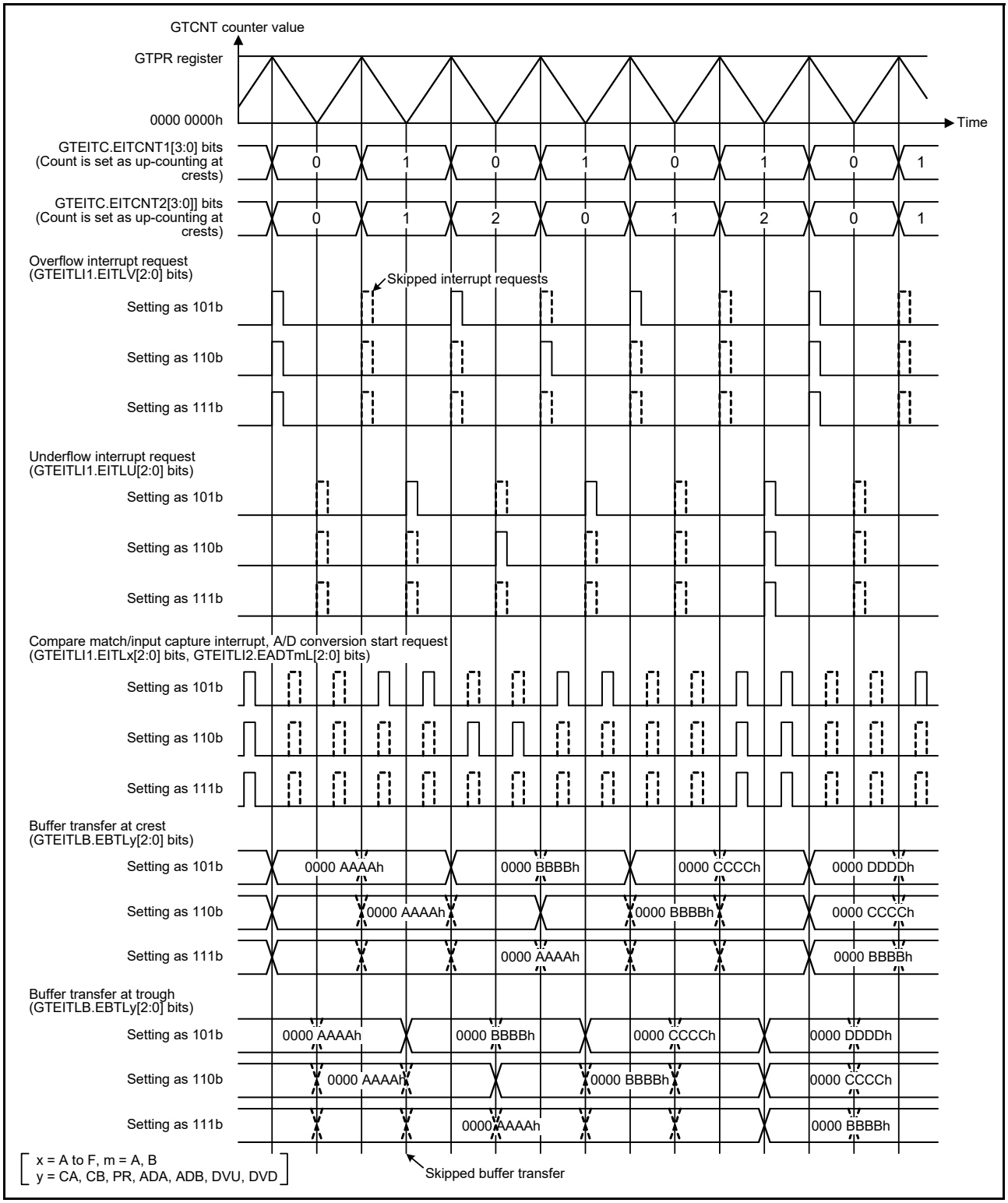


Figure 24.179 Example of Extended Interrupt Skipping Function Operation (Triangle Waves, Counting Crests, Extended Interrupt Skipping 1 Skipping Count: 1, Counting Crests, Extended Interrupt Skipping 2 Skipping Count: 2, Extended Interrupt Skipping Counter 2 Initial Value: 0, Skipped in the Period other than GTEITC.EITCNTk bits (k = 1, 2) as GTEITC.EIVTTk bit.)

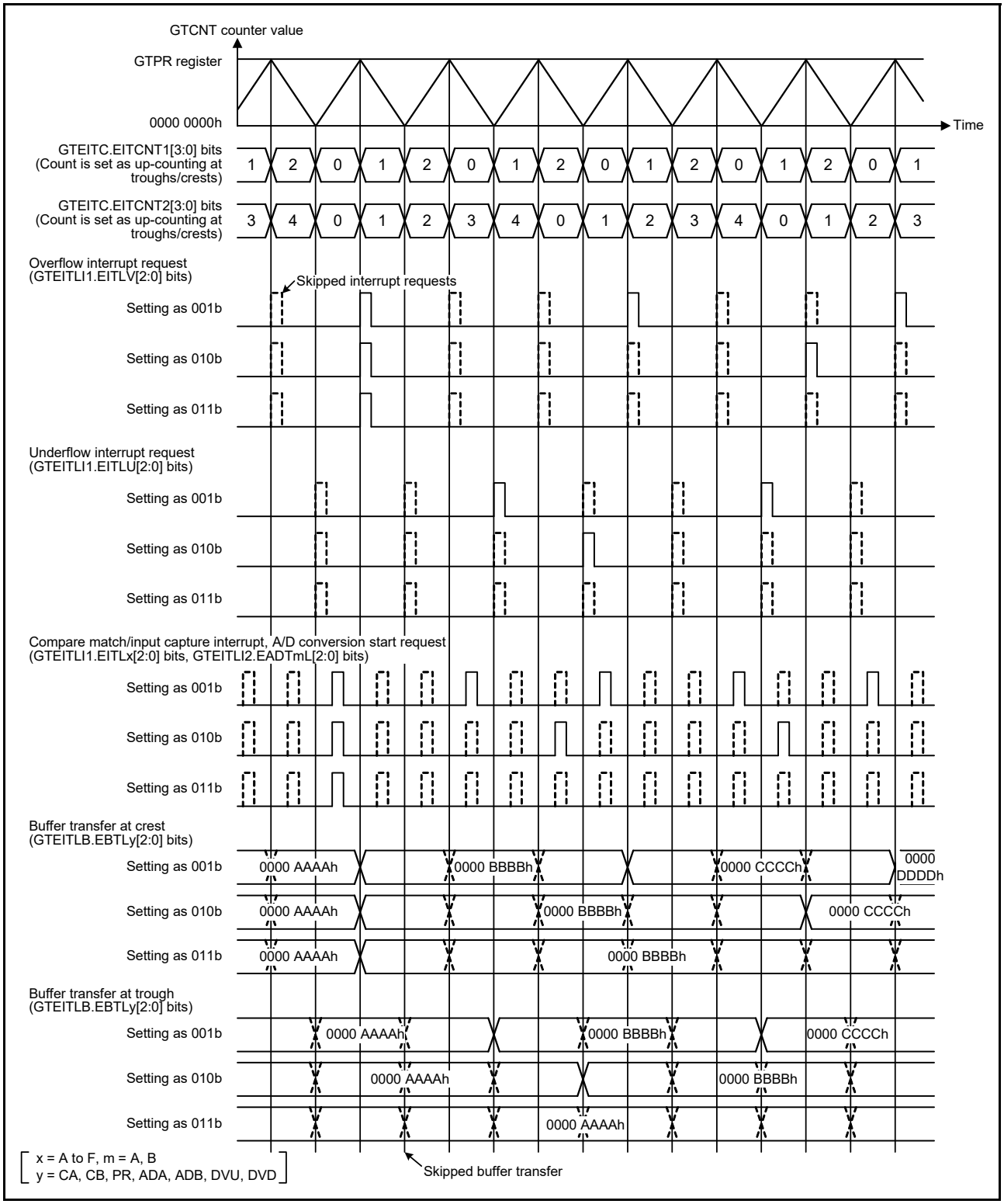


Figure 24.180 Example of Extended Interrupt Skipping Function Operation (Triangle Waves, Counting both Crests and Troughs, Extended Interrupt Skipping 1 Skipping Count: 2, Counting both Crests and Troughs, Extended Interrupt Skipping 2 Skipping Count: 4, Extended Interrupt Skipping Counter 2 Initial Value: 0, Skipped in the Period other than GTEITC.EITCNTk bits (k = 1, 2) as 0)

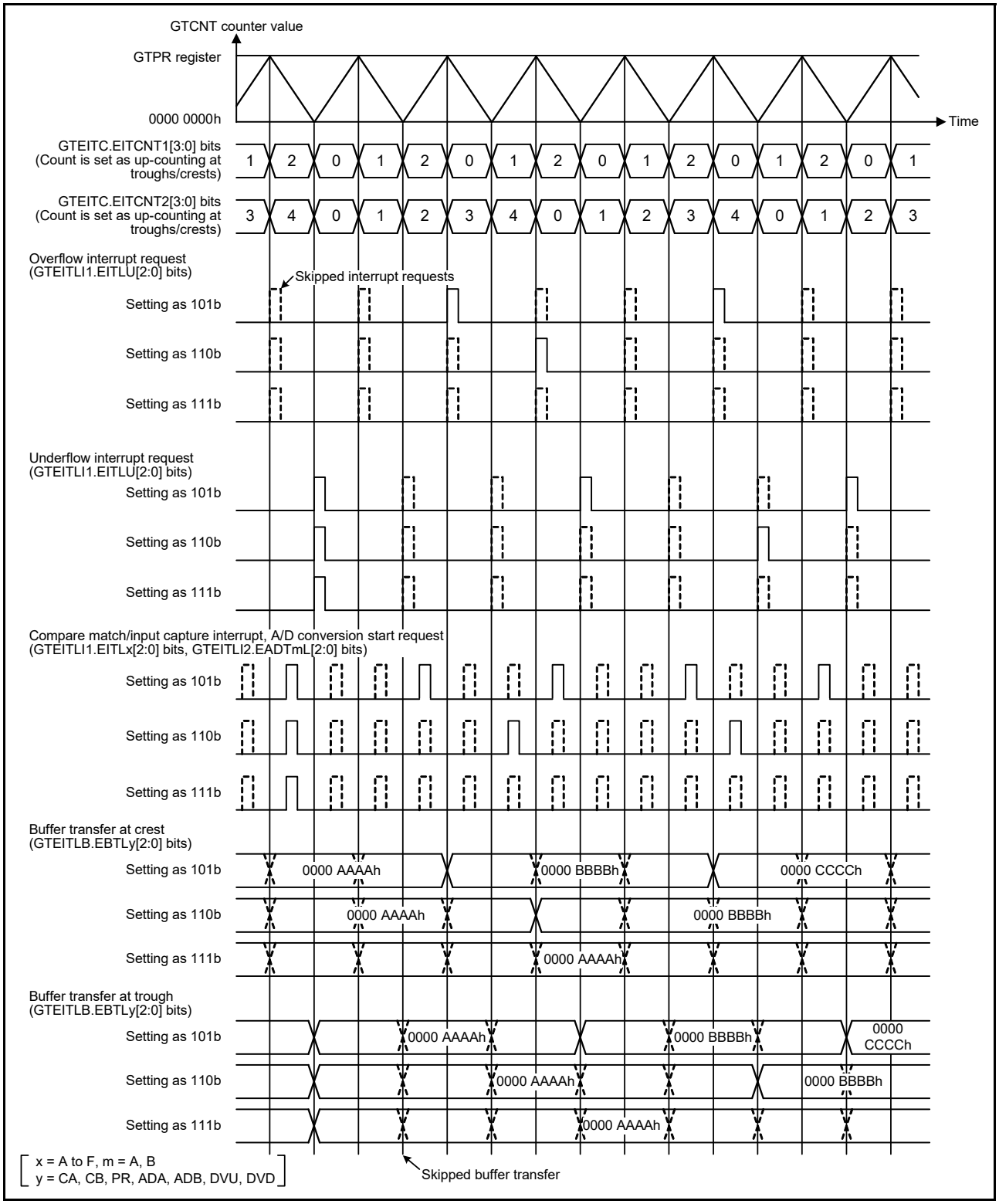


Figure 24.181 Example of Extended Interrupt Skipping Function Operation (Triangle Waves, Counting both Crests and Troughs, Extended Interrupt Skipping 1 Skipping Count: 2, Counting both Crests and Troughs, Extended Interrupt Skipping 2 Skipping Count: 4, Extended Interrupt Skipping Counter 2 Initial Value: 0, Skipped in the Period other than GTEITC.EITCNTk bits (k = 1, 2) as GTEITC.EIVTTk bit.)

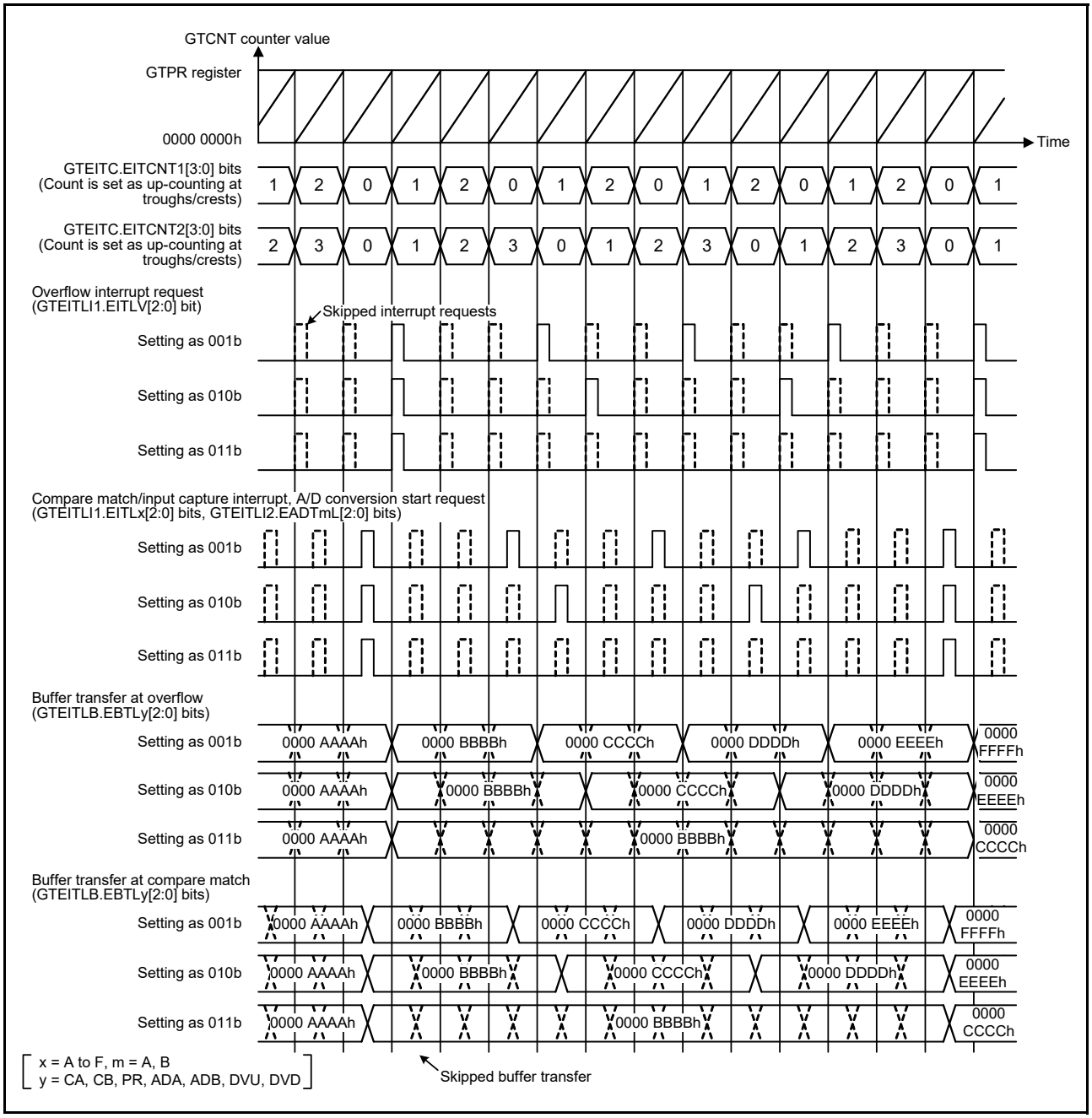


Figure 24.182 Example of Extended Interrupt Skipping Function Operation (Up-Counting in Sawtooth Waves, Counting Overflow, Extended Interrupt Skipping 1 Skipping Count: 2, Counting Overflow, Extended Interrupt Skipping 2 Skipping Count: 3, Extended Interrupt Skipping Counter 2 Initial Value: 0, Skipped in the Period other than GTEITC.EITCNTk bits (k = 1, 2) as 0)

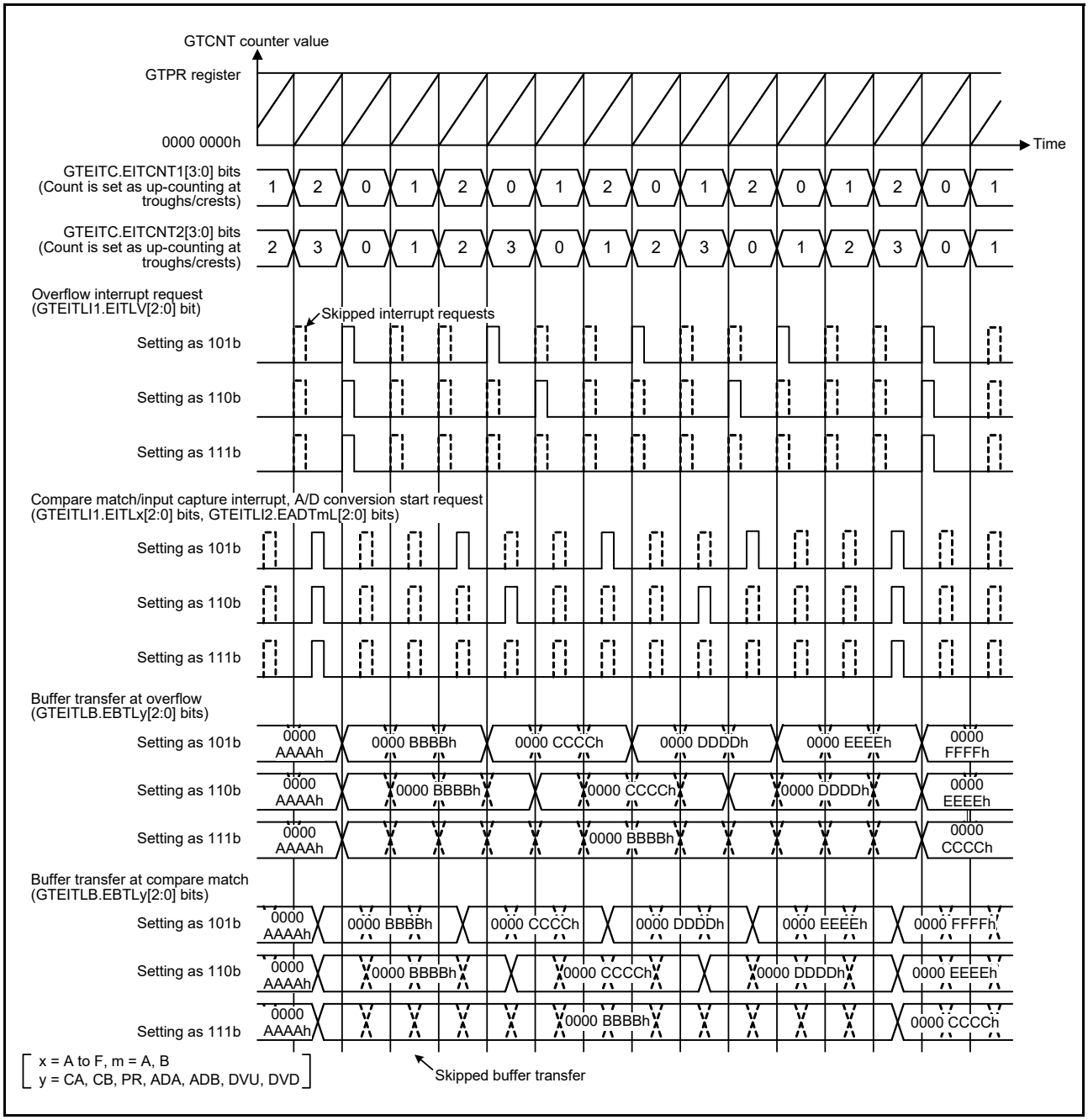


Figure 24.183 Example of Extended Interrupt Skipping Function Operation (Up-Counting Sawtooth Waves, Counting Overflow, Extended Interrupt Skipping 1 Skipping Count: 2, Counting Overflow, Extended Interrupt Skipping 2 Skipping Count: 3, Extended Interrupt Skipping Counter 2 Initial Value: 0, Skipped in the Period other than GTEITC.EITCNTk bits (k = 1, 2) as GTEITC.EIVTTk bit.)

Figure 24.184 shows an example of the extended interrupt skipping operation on the input capture. When the setting is for the input capture operation setting (GTCR.ICDS bit = 0) at count stop of the GTCNT counter, the interrupt by the input capture and the extended skipping on buffer transfer is enabled even at count stop of the GTCNT counter.
When the input capture is generated at count stop of the GTCNT counter by setting the ICDS bit as 0, an interrupt and a buffer transfer are skipped if the skipping counter value is the same with the skipping period set in a corresponding interrupt function select bit.

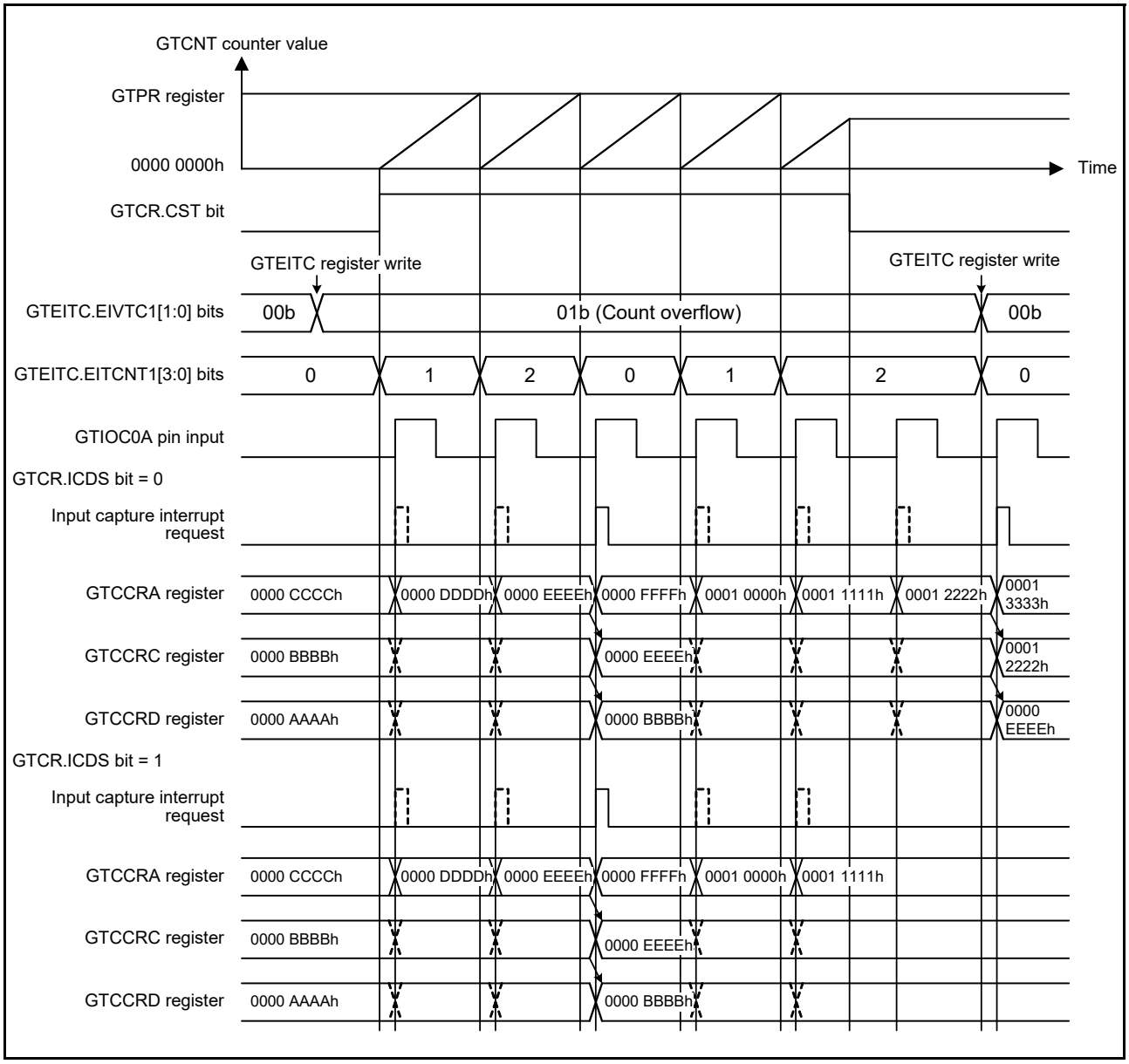


Figure 24.184 Example of Input Capture Operation in the Extended Interrupt Skipping Function Operation (Up-Counting Sawtooth Waves, Counting Overflow, Extended Interrupt Skipping 1 Skipping Count: 2, Skipped in the Period other than the EITCNT1 as 0, Input Capture at the Input Rising)

Figure 24.185 shows an example for setting the extended interrupt skipping.

The extended interrupt skipping counter 2 initial value is set by the written value to the ETICNT2IV[3:0] bits which is applied to change the setting from not counting the extended interrupt skipping counter 2 count source (GTEITC.EIVTC2[1:0] bits = 00b) to counting (EIVTC2[1:0] bits = other than 00b). Writing to the extended interrupt skipping counter 2 initial value bit (ETICNT2IV[3:0] bits) is performed only when the setting of the above mentioned extended interrupt skipping counter 2 initial value is written.

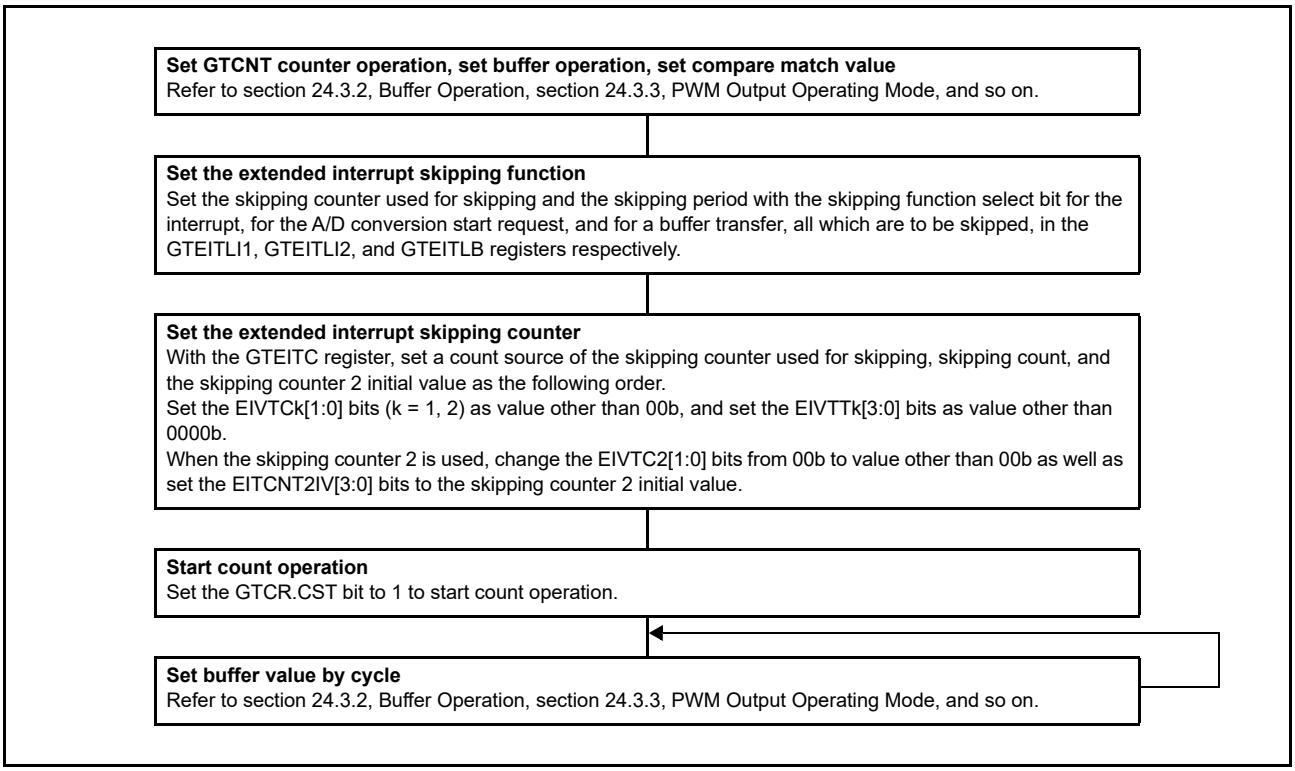


Figure 24.185 Example for Setting the Extended Interrupt Skipping

24.4.3.3 A/D conversion Start Request Compare Match Skipping Function

The A/D conversion start request and GTADTR register buffer transfer can be skipped by counting the compare matches of GTADTRA register and GTADTRB register based on settings of the GTADCMSC register and GTADCMSS register. The skipping period is set, related to the operation of the two independent A/D conversion start request compare match skipping counters (ADCMSCNT1[3:0] and ADCMSCNT2[3:0] bits in the GTADCMSC register), as a period that either of the counter value for such skipping counters is other than 0 or other than the skipping count. The skipping period can also be set as a period that the both skipping counters are set as other than 0 or other than the skipping count for the counter value.

Figure 24.186 shows the counter operation for A/D conversion start request compare match skipping.

The counter operation for A/D conversion start request compare match skipping is set by the GTADCMSC register.

The ADCMSCNT1[3:0] bits count the count source (in Figure 24.186, a crest is selected) selected by the A/D conversion start request compare match skipping counter 1 count source select bit (GTADCMSC.ADCMSC1[1:0] bits), and repeat the count operation which returns to 0 when the skipping count (in Figure 24.186, count is 2) set by the A/D conversion start request compare match skipping 1 skipping count setting bit (ADCMST1[3:0] bits) is achieved.

The ADCMSCNT2[3:0] bits count the count source (in Figure 24.186, a trough is selected) selected by the A/D conversion start request compare match skipping counter 2 count source select bit (GTADCMSC.ADCMSC2[1:0] bits), and repeat the count operation which returns to 0 when the skipping count (in Figure 24.186, count is 2) set by the A/D conversion start request compare match skipping 2 skipping count setting bit (ADCMST2[3:0] bits) is achieved.

The ADCMSCNTk[3:0] bits (k = 1, 2) can be set for the initial value. The initial value is set when the GTADCMSC register is written by the access of 16 bits or 32 bits while the setting for the A/D conversion start request compare match skipping counter k is not counted (ADCMSCk[1:0] bits are 00b) and the write value to the ADCMSCk[1:0] bits is other than 00b.

When the initial value is set, the write value to the A/D conversion start request compare match skipping counter k initial value bits (ADCMSCNTkIV[3:0] bits) are set as the initial value for the ADCMSCNT2[3:0] bits.

The A/D conversion start request compare match skipping counter starts up-counting at the first count clock after the setting is modified from not counting to counting.

The ADCMSCNT1[3:0] and ADCMSCNT2[3:0] bits for the A/D conversion start request compare match skipping function retain the value even after the GTCNT counter operation is stopped, and the counting can be resumed from the value before the counter is stopped.

When values for the ADCMSCNT1[3:0] and ADCMSCNT2[3:0] bits are to be reset (0000b), set the ADCMSC1[1:0] and ADCMSC2[1:0] bits to the setting for not counting (not skipping) (00b).

When the skipping count is to be changed, change the count after stopping the skipping counter operation (set either of the ADCMSC1[1:0] bit or the ADCMSC2[1:0] bit to 00b).

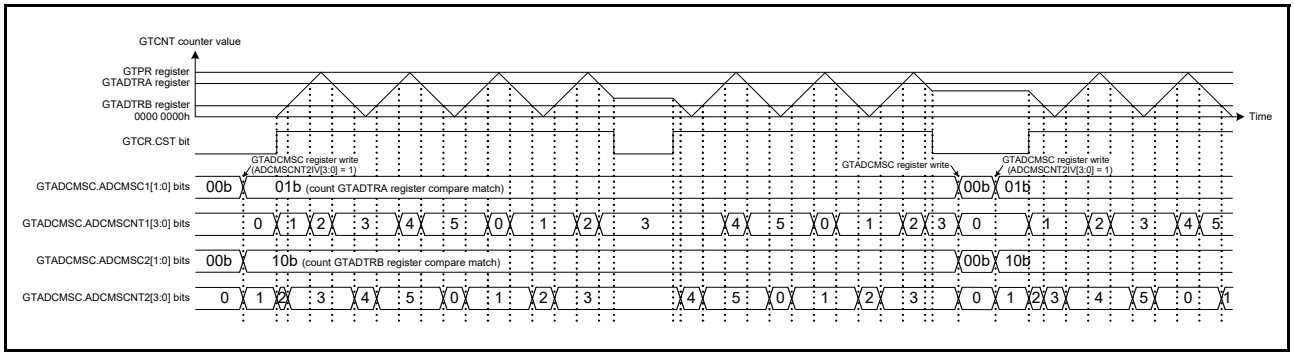


Figure 24.186 Example of A/D Conversion Start Request Compare Match Skipping Function Operation

A/D conversion start request skipping by the ADCMSC register can be performed simultaneously with skipping by the GTITC register or GTEITC register. A skipping period in this case is represented by OR-ed skipping periods of respective registers.

Figure 24.187 shows corresponding interrupt skipping operations and A/D conversion start request skipping operation are performed simultaneously.

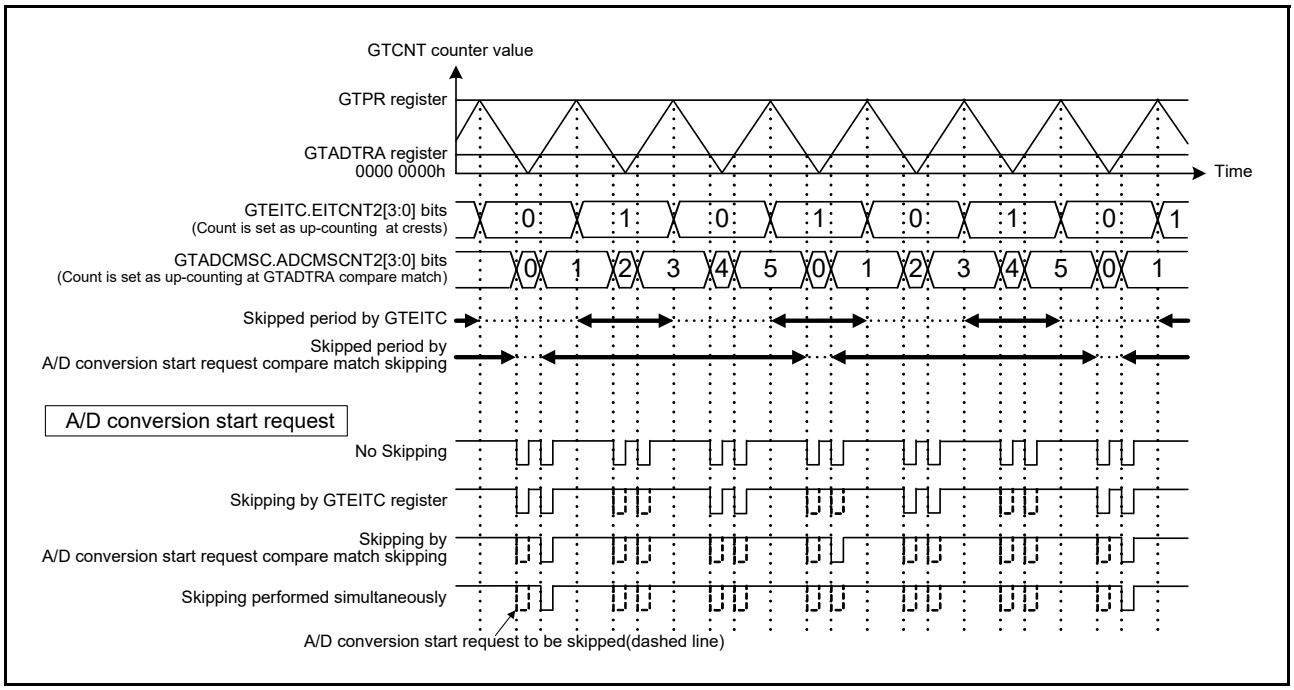


Figure 24.187 Example of A/D Conversion Start Request Compare Match Skipping Function Operation (Extended Interrupt Skipping: EIVTC2[1:0] = 01b, EADTAL[2:0] = 010b, A/D Conversion Start Request Compare Match Skipping: ADCMSC2[1:0] = 01b, ADCMSAL[2:0] = 010b)

When A/D conversion start request skipping which can be set by the GTADCMSS register is performed, updating of status flag and the ELC event output depend on the A/D conversion start request enable bit in the GTINTAD register. All the operations by A/D conversion start request which are set as disable by the GTINTAD register is not performed.
A buffer transfer skipping by the GTADCMSS register is performed in all of buffer operation which is enabled in the GTBER and GTBER2 register, or all buffer operations performed by sawtooth-wave one-shot pulse mode or triangle-wave PWM mode 3 or complementary PWM mode (excludes buffer transfer from GTCCRC, GTCCRE to GTCCRA).
An A/D conversion start request skipping and a buffer transfer skipping are operated individually.
Figure 24.188 and Figure 24.189 show examples of extended skipping function operation.

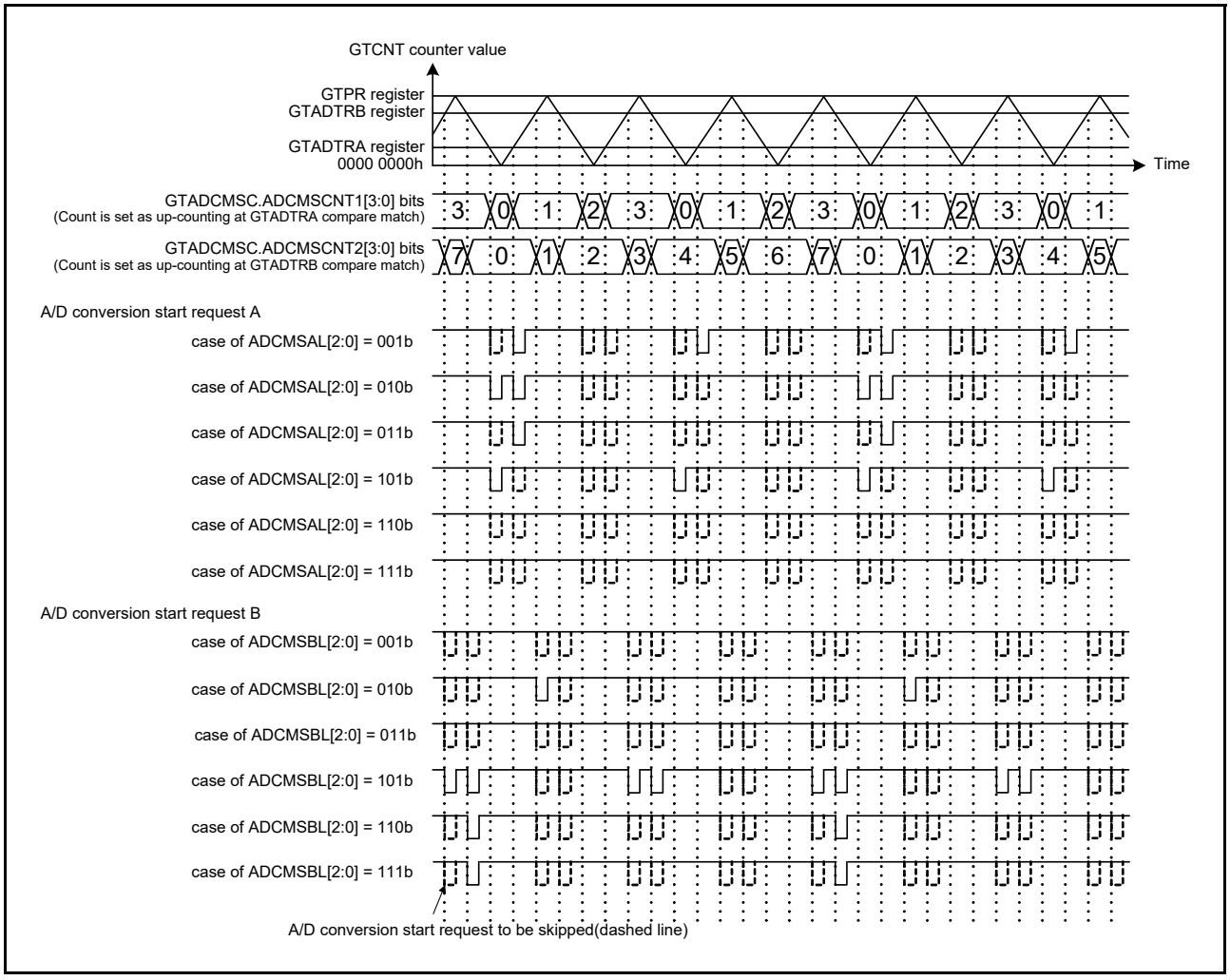


Figure 24.188 Example of A/D Conversion Start Request Compare Match Skipping Function Operation (Triangle wave, A/D Conversion Start Request Compare Match Skipping 1 Skipping Count: 3 Counting GTADTRA compare match, A/D Conversion Start Request Compare Match Skipping 2 Skipping Count: 7 Counting GTADTRB compare match)

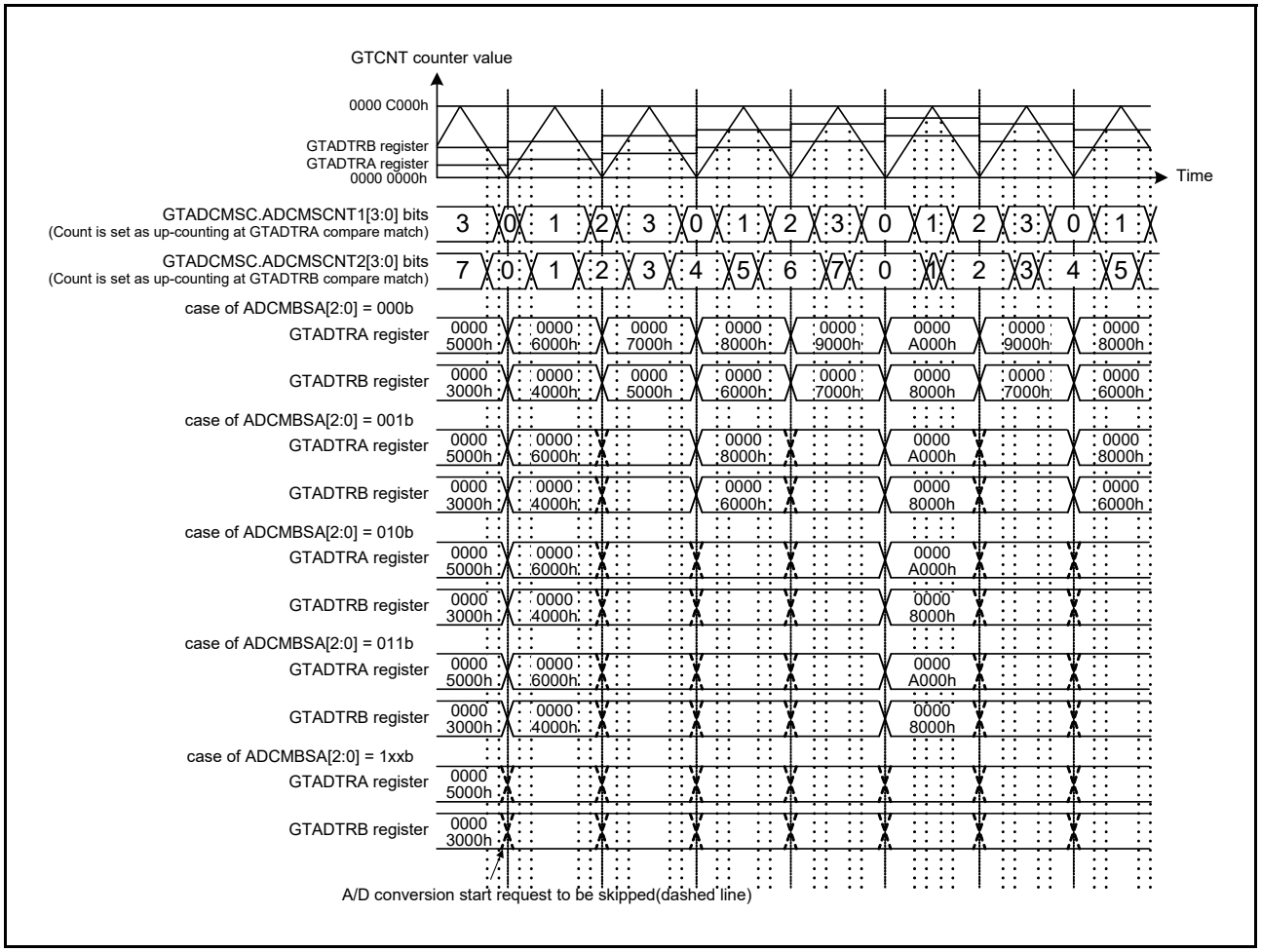


Figure 24.189 Example of A/D Conversion Start Request Compare Match Skipping Function Operation (Triangle wave, A/D Conversion Start Request Compare Match Skipping 1 Skipping Count: 3 Counting GTADTRA compare match, A/D Conversion Start Request Compare Match Skipping 2 Skipping Count: 7 Counting GTADTRB compare match, Buffer transfer of GTADTRA and GTADTRB at trough)

Figure 24.190 shows an example of setting the A/D conversion start request compare match skipping.
The A/D conversion start request compare match skipping counter 2 initial value is set by the written value to the ADCMSCNT2IV[3:0] bits which is applied to change the setting from not counting the extended interrupt skipping counter 2 count source (GTADCMSC.ADCMSC2[1:0] bits = 00b) to counting (ADCMSC2[1:0] bits = other than 00b). Writing to the A/D conversion start request compare match skipping counter 2 initial value bit (ADCMSCNT2IV[3:0] bits) is performed only when the setting of the above mentioned A/D conversion start request compare match skipping counter 2 initial value is written.

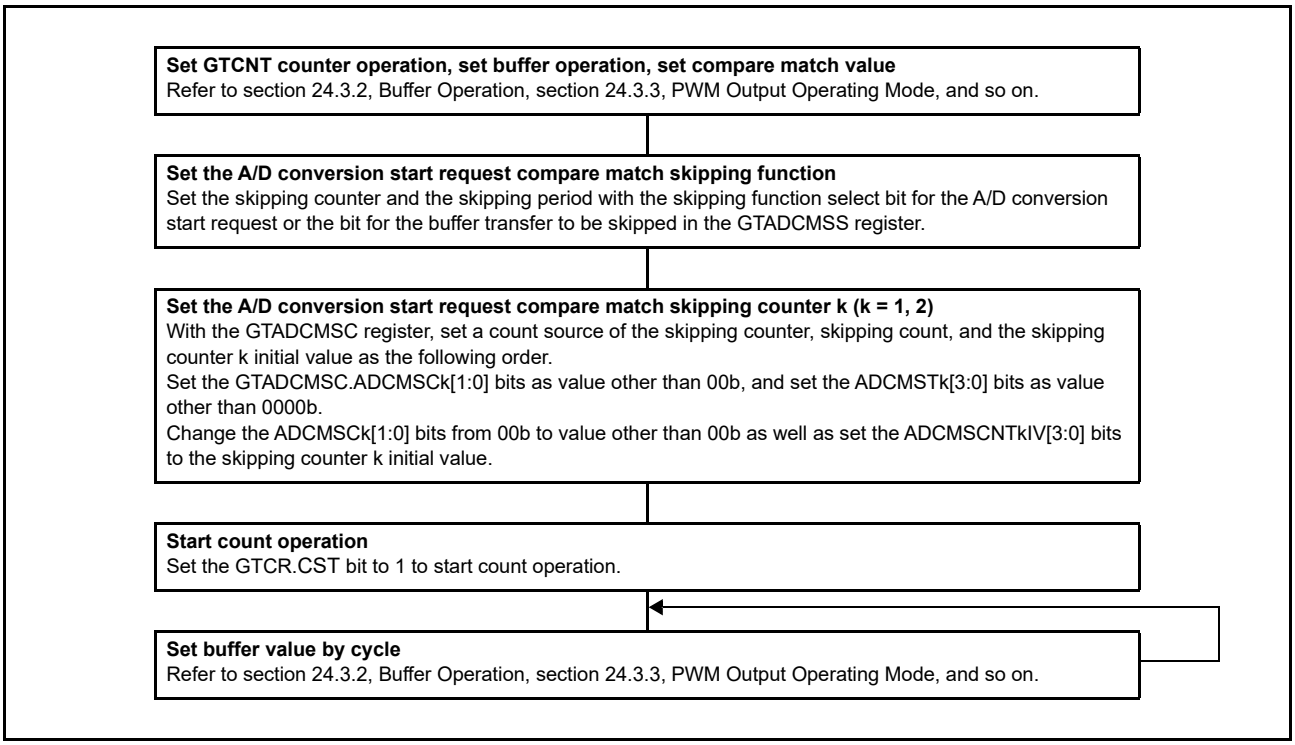


Figure 24.190 Example for Setting the A/D Conversion Start Request Compare Match Skipping

24.5 A/D Conversion Start Request

An A/D conversion start request can be issued at a compare match between the GTCNT counter and the GTADTRA or GTADTRB register, up-counting only, down-counting only, or both up-counting and down-counting can be specified by setting the GTINTAD register.

In complementary PWM mode, the A/D conversion start request can be issued at a compare match with the GTCNT counter of master channel.

During event count operation, the A/D conversion start request cannot be generated.

The A/D conversion start request is not directly output to the A/D converter, but an interrupt, and an event signal to the ELC are output.

The GTADTRA and GTADTRB registers each has two buffer registers. Buffer operation with the GTADTRA register used together with the GTADTBRA and GTADTDBRA registers, and buffer operation with the GTADTRB register used together with the GTADTBRB and GTADTDBRB registers can be performed.

The timing of the generation of requests to start A/D conversion can be monitored by an external pin. When the A/D conversion start request signal to be monitored is selected in the GTADSMR.ADSMSk bit (k = 0, 1) and when the output is enabled in the ADSMENk bit, a signal is output synchronized with a cycle frame of the timer used to generate the A/D conversion start request signal, of which the output is driven high at the generation of the A/D conversion start request signal by the GTADSMk pin, or at the end of the cycle of which the output is driven low. When a signal to request the start of A/D conversion is generated at the end of the cycle, the generation of this signal has priority in terms of monitoring output and the output remains at the high level till the end of the next cycle. The registers (GTADTRA and GTADTRB) that are sources of generating the A/D conversion start request signals and their counting directions can be checked by the A/D conversion start request flags (ADTRAUF, ADTRADF, ADTRBUF, and ADTRBDF) in the GTST register. When the output of the same A/D conversion start request signal monitoring output is enabled for multiple channels, ORed signals will be output from the GPTW.

Figure 24.191 shows an example of A/D conversion start request operation, Figure 24.192 shows an example for setting A/D conversion start request operation, and Figure 24.193 shows an example for A/D conversion start request timing operation.

Figure 24.193 shows an example of the output of A/D conversion start request A by the ELC as start source 0 for the A/D converter. The A/D conversion start request A signal is output by the ELC in response to a match in comparison with the GTADTRA register. The A/D conversion start request A signal is synchronized with PCLKC. The ELC receives the signal in synchronization with PCLKB, and then outputs the A/D conversion start source 0 signal after 1 cycle of the PCLKB has elapsed.

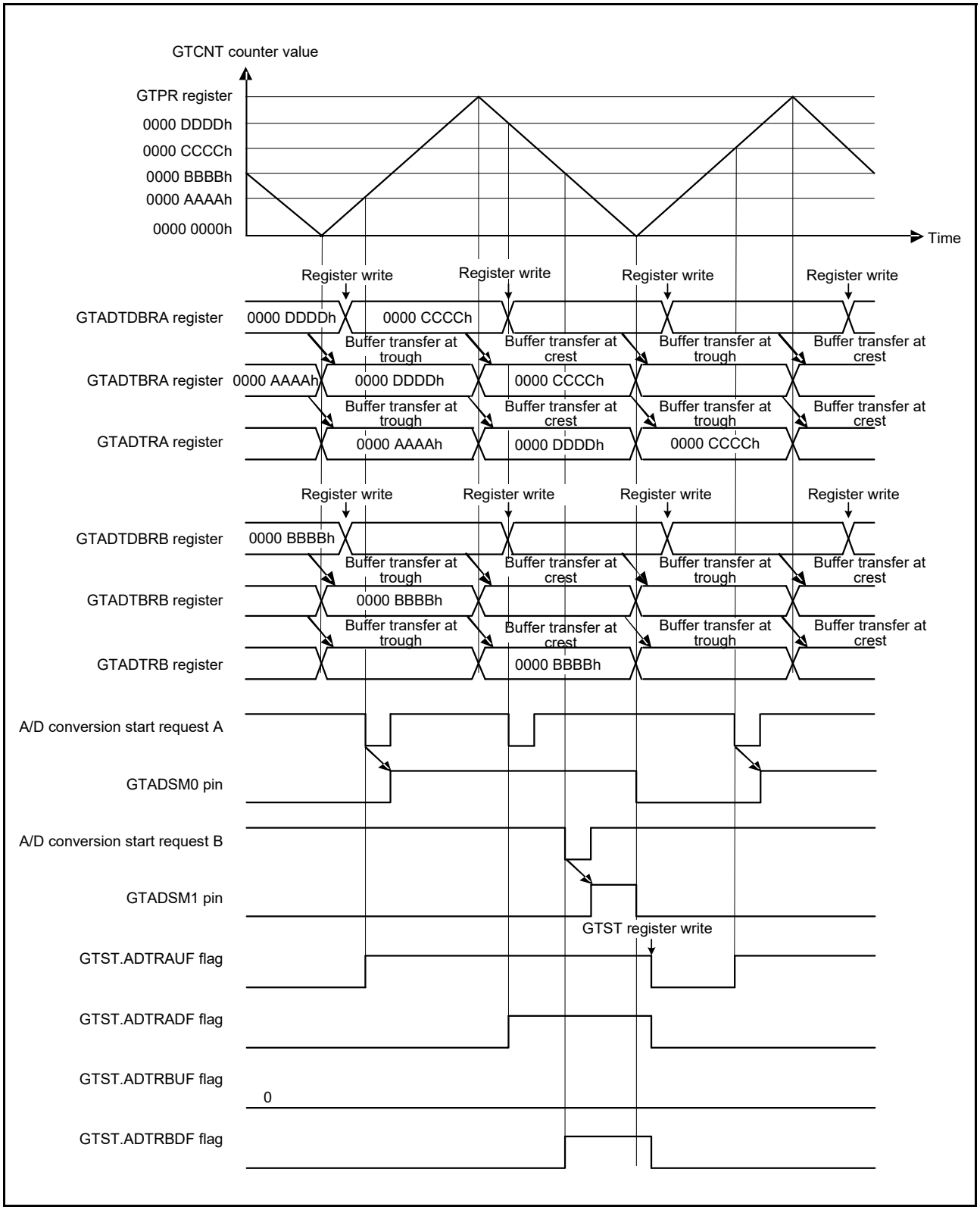


Figure 24.191 Example of A/D Conversion Start Request Timing Operation (Triangle Waves, Double Buffer Operation, Buffer Transfer at Both Troughs and Crests, A/D Conversion Start Requested by GTADTRA Register at Both Up-Counting and Down-Counting, A/D Conversion Start Requested by GTADTRB Register at Down-Counting, Monitoring of the GTADTRA Register Up-Counting by the GTADSM0 Pin, Monitoring of the GTADTRB Register Down-Counting by the GTADSM1 Pin)

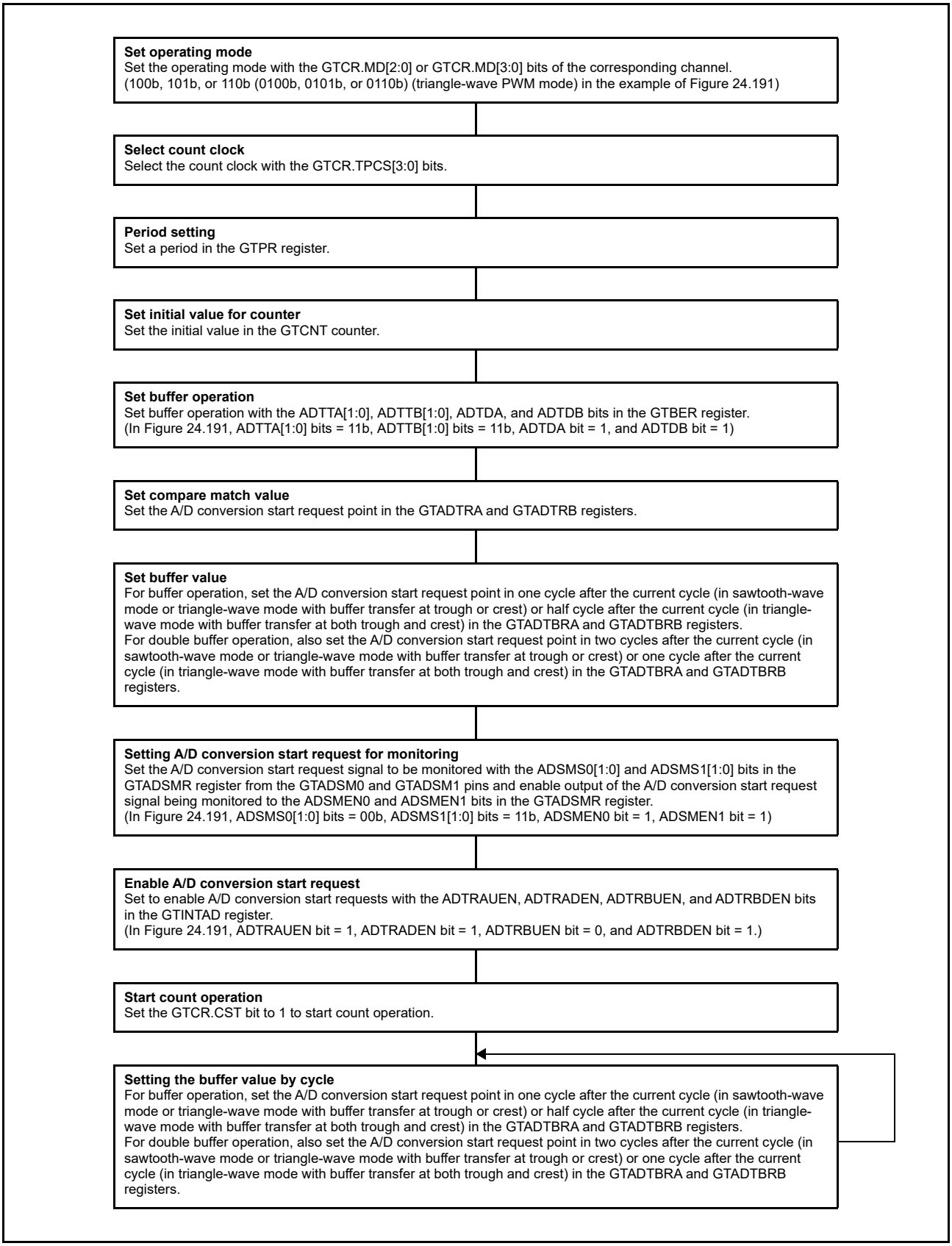


Figure 24.192 Example for Setting A/D Conversion Start Request Operation

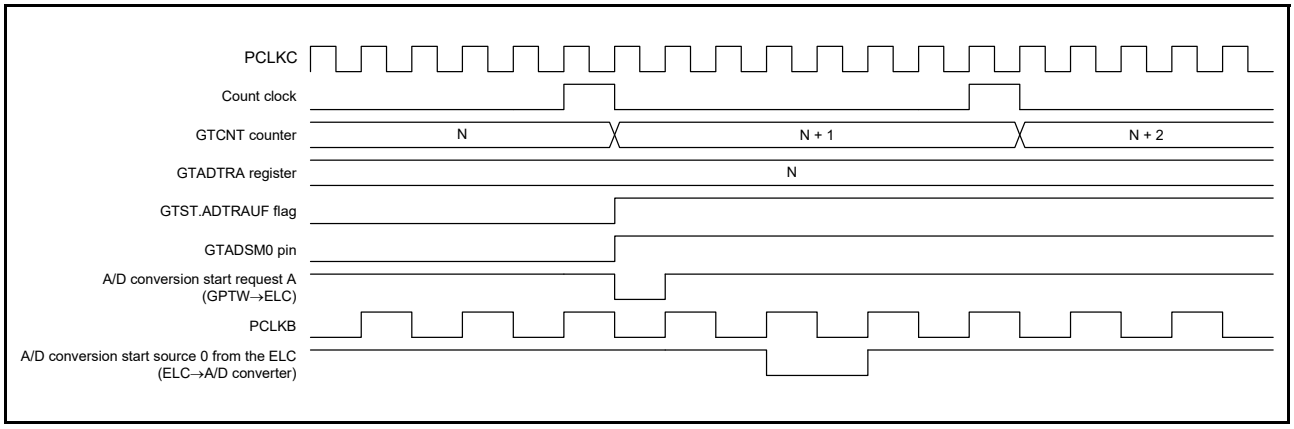


Figure 24.193 Example of A/D Conversion Start Request Timing Operation

24.6 Operations Linked by the ELC

24.6.1 Event Signal Output to the ELC

GPTW is capable of operation linked with another module set in advance when its interrupt request signal is used as an event signal by the event link controller (ELC).
The event signal can be output regardless of the setting of the corresponding interrupt request enable bits except the A/D conversion start request.
The A/D conversion start request during up-counting/down-counting can be enabled/disabled individually with the A/D conversion start request enable bit to output to the ELC.
The following is a list of the event signals output by the GPTW to the ELC. Each GPTW channel has the first 10 event signals. The last of the listed event signals is common to all channels. A dead time error interrupt does not have a corresponding event signal.

- Generation of compare match A interrupt
- Generation of compare match B interrupt
- Generation of compare match C interrupt
- Generation of compare match D interrupt
- Generation of compare match E interrupt
- Generation of compare match F interrupt
- Generation of overflow interrupt
- Generation of underflow interrupt
- Generation of A/D conversion start request A
- Generation of A/D conversion start request B
- GPTW (OPS) U-/V-/W-phase input edge detected

24.6.2 GPTW Operations in Response to Receiving Event Signals from the ELC

The GPTW can perform the following operations by the signals for sources A to H which are output from the ELC. Each signal is provided to all channels. Select an event source for a desired operation by a relevant source select register in a channel.

- Operation of count start/count stop/counter clearing
- Operation of up-counting/down-counting
- Operation of input capture A and B

Refer to operation by hardware sources in section 24.3.1.1, Counter Operation for respective operations.

24.7 Noise Filter Function

Each pin for use in input capture to the GPTW is equipped with a noise filter. The noise filter samples input signals at the sampling clock and removes the pulses of which length is less than three sampling periods.
The noise filter functionality includes enabling and disabling of the noise filter for each pin and setting of the sampling clock for each channel.
Figure 24.194 shows the timing of noise filtering.

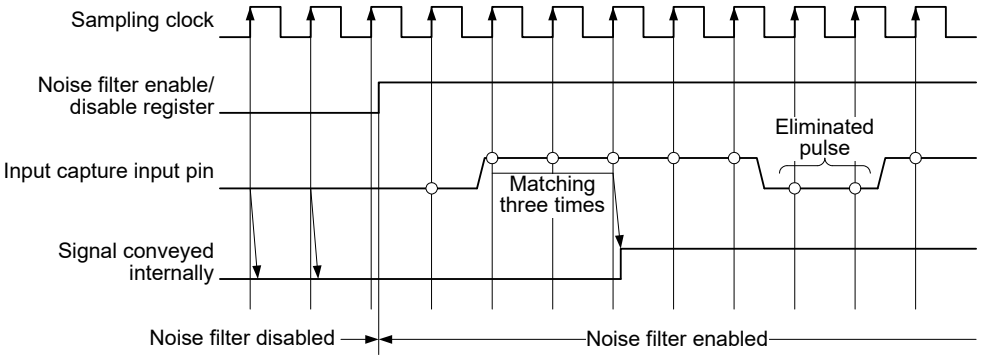


Figure 24.194 Timing of Noise Filtering

If noise filtering is enabled, input capture operation is performed on the edges of noise-filtered signal after a delay of (minimum sampling interval × 2 + PCLKC) due to noise filtering for the input capture input.

24.8 Protection Function

24.8.1 Write-Protection for Registers

In order to avoid incorrect writing to registers, write access to registers can be enabled or disabled per channel by setting the GTWP.WP bit for the given channel.
The WP bits enable or disable writing to the registered are listed below.
GTSSR, GTPSR, GTCSR, GTUPSR, GTDNSR, GTICASR, GTICBSR, GTCR, GTUDDTYC, GTIOR, GTINTAD, GTST, GTBER, GTITC, GTCNT, GTCCRA, GTCCRB, GTCCRC, GTCCRD, GTCCRE, GTCCRF, GTPR, GTPBR, GTPDBR, GTADTRA, GTADTBRA, GTADTDBRA, GTADTRB, GTADTBRB, GTADTDBRB, GTDTCR, GTDVU, GTDVD, GTDBU, GTDBD, GTSOS, GTSOTR, GTADSMR, GTEITC, GTEITLI1, GTEITLI2, GTEITLB, GTICLF, GTPC, GTADCMSC, GTADCMSS, GTBER2, GTOLBR, GTICCR
Every bit in registers GTSTR, GTSTP, and GTCLR, which can update the corresponding registers in other channels and can be updated by any of the corresponding registers in other channels conversely, can be protected by setting the GTWP.STRWP, STPWP, and CLRWP bits, respectively, per channel.
Likewise, writing to the GTSECSR and GTSECR registers, which can control all channels by writing to the GTSECSR and GTSECR registers of a given channel, can by enabled or disabled by the setting of the GTWP.CMNWP bit.
Protection by the GTWP register only targets writing operation by the CPU. Updating the register generated in related to the CPU writing is not protected.

24.8.2 Disabling of Buffer Operation

If the timing of buffer register write is too late for the buffer transfer timing, buffer operation can be suspended with setting the BD[0], BD[1], BD[2], and BD[3] bits in the GTBER resister.
Specifically, buffer transfer can be temporarily disabled, even though a buffer transfer condition is generated during buffer register write, by setting the BD[0], BD[1], BD[2], and BD[3] bits to 1 (buffer operation disabled) before buffer register write, and setting the bits to 0 (buffer operation enabled) after completion of writing to all the buffer registers.
The BD[0], BD[1], BD[2], and BD[3] bits can be set on channel basis by writing directly to the GTBER register or can be set simultaneously by setting the GTSECR register for multiple channels which were set by the GTSECSR register.
Figure 24.195 shows an example of operation for disabling buffer operation.

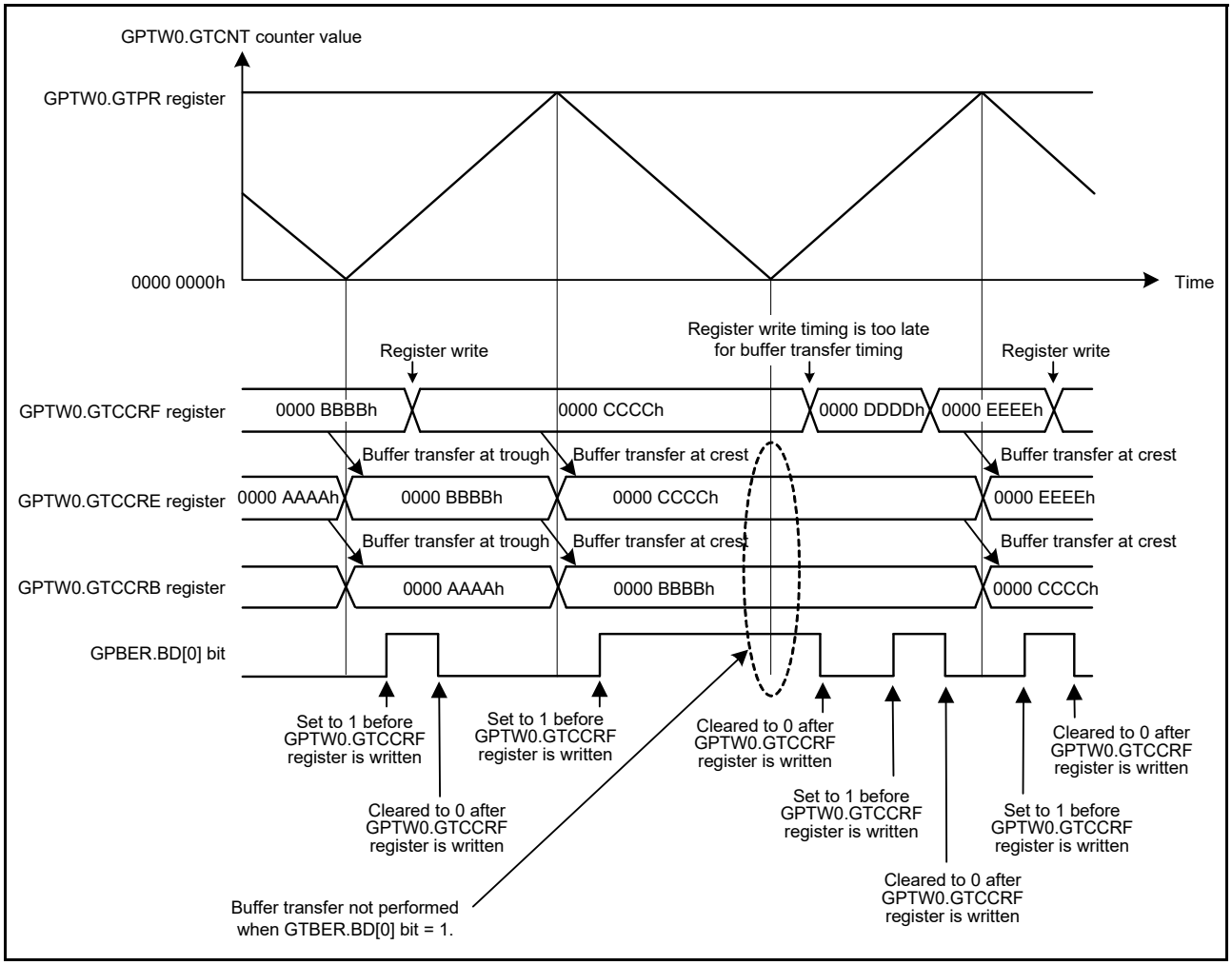


Figure 24.195 Example of Operation for Disabling Buffer Operation (Triangle Waves, Double Buffer Operation, Buffer Transfer at Both Troughs and Crests)

24.8.2.1 Simultaneous Control of Buffer Operations of Multiple Channels

The GTBER.BD bit can be set by writing directly to the GTBER register per channel or by making settings in the GTSECR register for multiple channels that have already set in the GTSECSR register.
Follow the procedure below to simultaneously set the GTBER.BD bits of multiple channels.

(1) Select the channels for simultaneously setting of the GTSECSR register

Set the GTSECSR register so that the values at the bit positions for the corresponding channels for simultaneously setting of the GTBER.BD bits become 1. All GTSECSR registers can be updated by writing to the GTSECSR register of any channel.

(2) Simultaneously set the GTBER.BD bits by updating the GTSECR register

In the GTSECR register, set the operation of the GTBER.BD bits (enabling or disabling of buffer operation) which are to be simultaneously set. Writing to a GTSECR register from any channel updates the GTBER.BD bits in all channels corresponding to the bits set as 1 in the GTSECSR register, in accordance with the value of the GTSECR register. Figure 24.196 and Figure 24.197 show examples of simultaneously controlling the enabling or disabling of buffer operation for multiple channels.

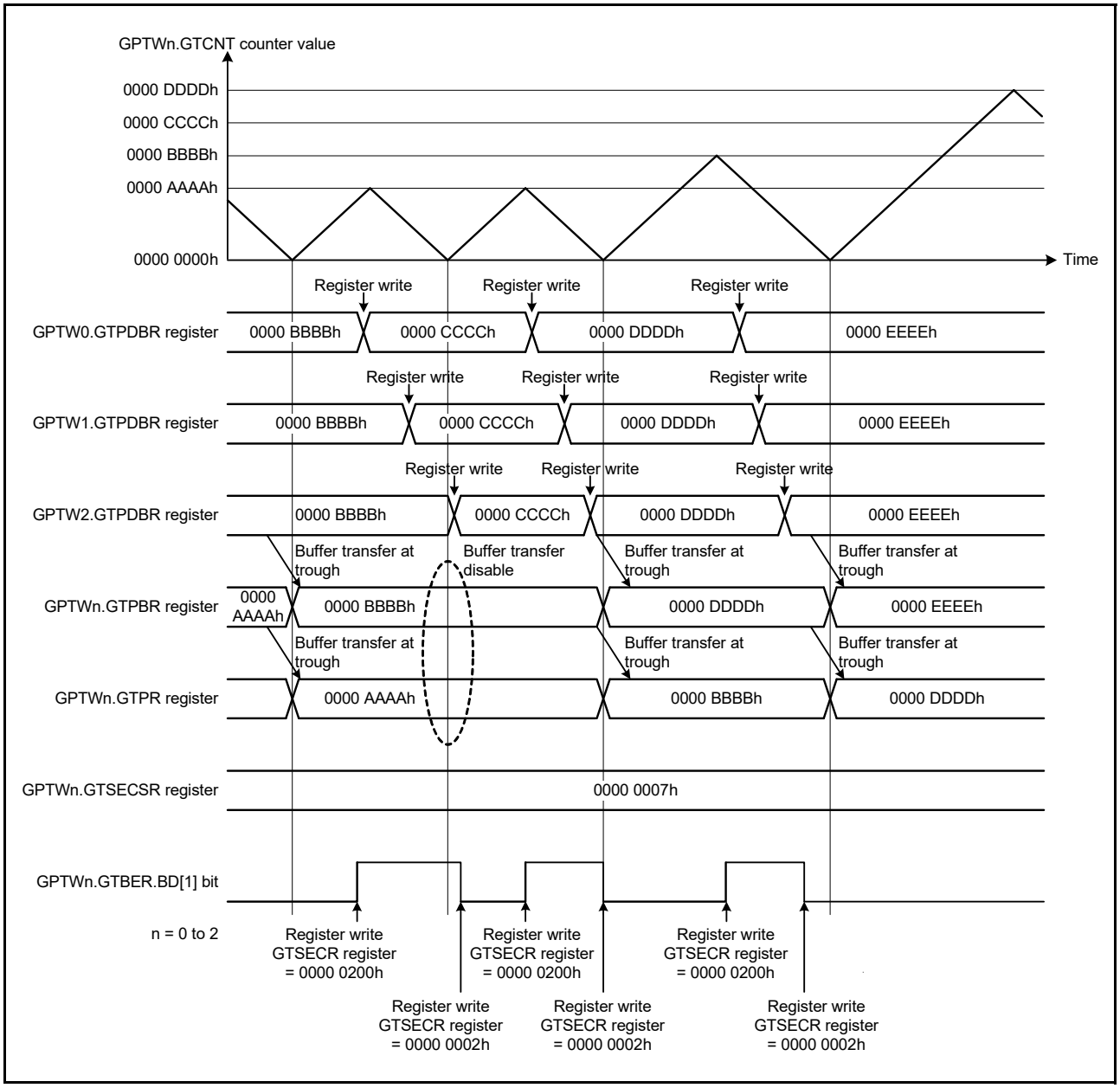


Figure 24.196 Example of Multiple Channel Operation for Disabling Buffer Operation (Triangle Waves, Double Buffer Operation)

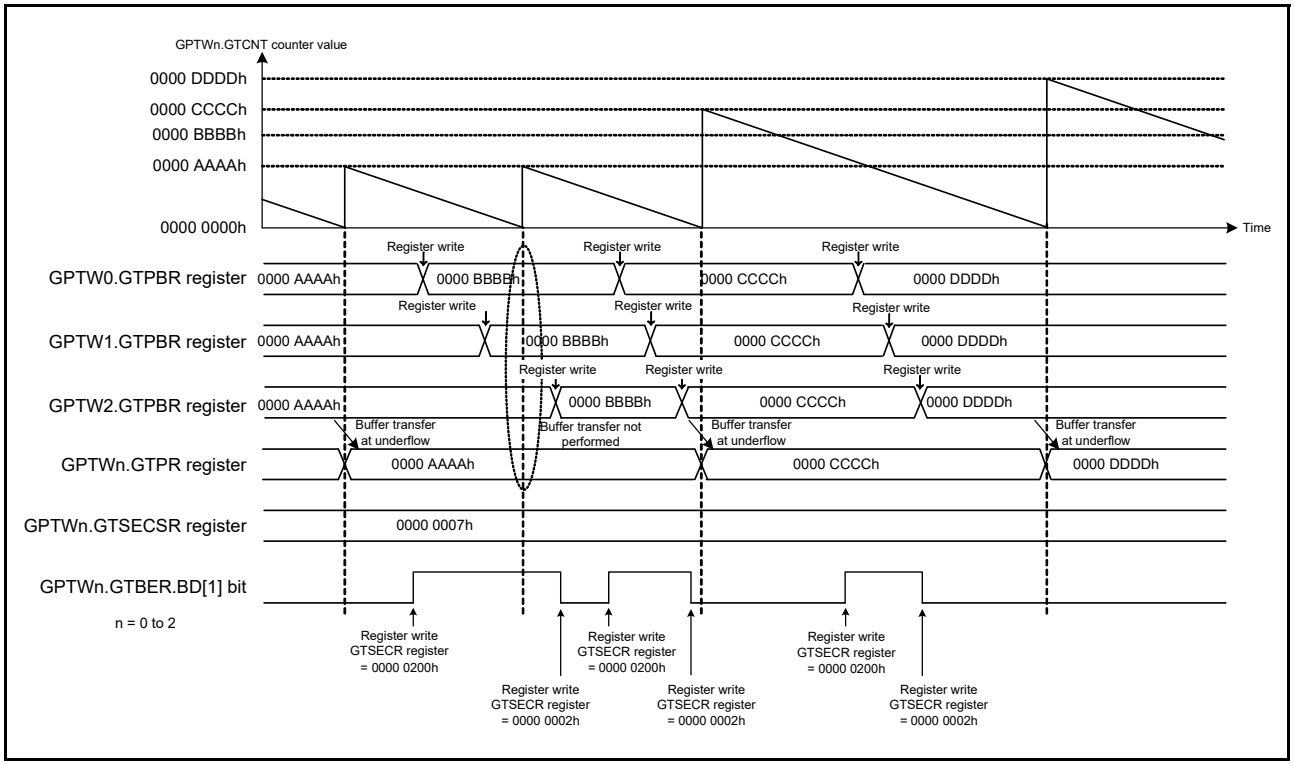


Figure 24.197 Example of Multiple Channel Operation for Disabling Buffer Operation (Sawtooth Waves, Single Buffer Operation)

24.8.2.2 Repeated Double-Buffered Operation When Disabling GTCCR Buffer Transfer

When a GTBER.DBRTECm (m = A, B) bit is set to 1 in sawtooth-wave one-shot pulse mode or triangle-wave PWM mode 3, transfer from the intermediate buffer to the GTCCRm (m = A, B) register is repeated on a cyclic basis even while buffer transfer is disabled by the setting of the GTBER.BD[0] bit or by the buffer transfer extended skipping function.

(1) In sawtooth-wave one-shot pulse mode

In sawtooth-wave one-shot pulse mode, the compare match value for the first half of a waveform is stored in temporary register x (x = C, E) as the intermediate buffer for the GTCCRx (x = C, E) register and the compare match value for the second half of a waveform is stored in temporary register m (m = A, B) as the intermediate buffer for the GTCCRy (y = D, F) register, respectively, for compare match values during repeated buffer operation, and the given values are alternately transferred to the GTCCRm (m = A, B) registers.

Table 24.41 lists the types of buffer transfer of the GTCCR register during counting in sawtooth-wave one-shot pulse mode. While counting is stopped, the setting of the value in the temporary register is transferred through forcible buffer transfer.

In forcible buffer transfer, the values of the GTCCRy (y = D, F) registers are transferred to temporary registers m (m = A, B), and the values of the GTCCRx (x = C, E) are transferred to temporary registers x (x = C, E), when the setting of the corresponding GTBER.DBRTECm (m = A, B) bit is 1, respectively.

When the setting of the GTBER.DBRTECm (m = A, B) bit is 1, values written by the CPU to the GTCCRm (m = A, B) registers are reflected as the values of temporary registers x (x = C, E).

Table 24.41 GTCCR Buffer Operation in Sawtooth-Wave One-Shot Pulse Mode

		Timing of transfer				
GTBER.DBRTECm	**Buffer transfer**	**GTCCRx ↓ GTCCRm**	**GTCCRx ↓ Temporary register x**	**Temporary register x ↓ GTCCRm**	**GTCCRy ↓ Temporary register m**	**Temporary register m ↓ GTCCRm**
0	Transfer-enabled period	Overflow or underflow	No transfer	No transfer	Overflow or underflow	GTCCRm compare match
	Transfer-disabled period	No transfer	No transfer	No transfer	No transfer	No transfer
1	Transfer-enabled period	Overflow or underflow	Overflow or underflow	No transfer	Overflow or underflow	GTCCRm compare match
	Transfer-disabled period	No transfer	No transfer	Overflow or underflow	No transfer	GTCCRm compare match

Figure 24.198 shows the operation of generating transfer-disabled period by extended buffer transfer skipping as an example of repeated double buffer operations when the GTCCR buffer transfer disabled in sawtooth-wave one-shot pulse mode.

Figure 24.199 shows the operation of generating transfer-disabled period by updating the GTBER.BD[0] bit as an example of repeated double-buffer operations when the GTCCR buffer transfer is disabled in sawtooth-wave one-shot pulse mode.

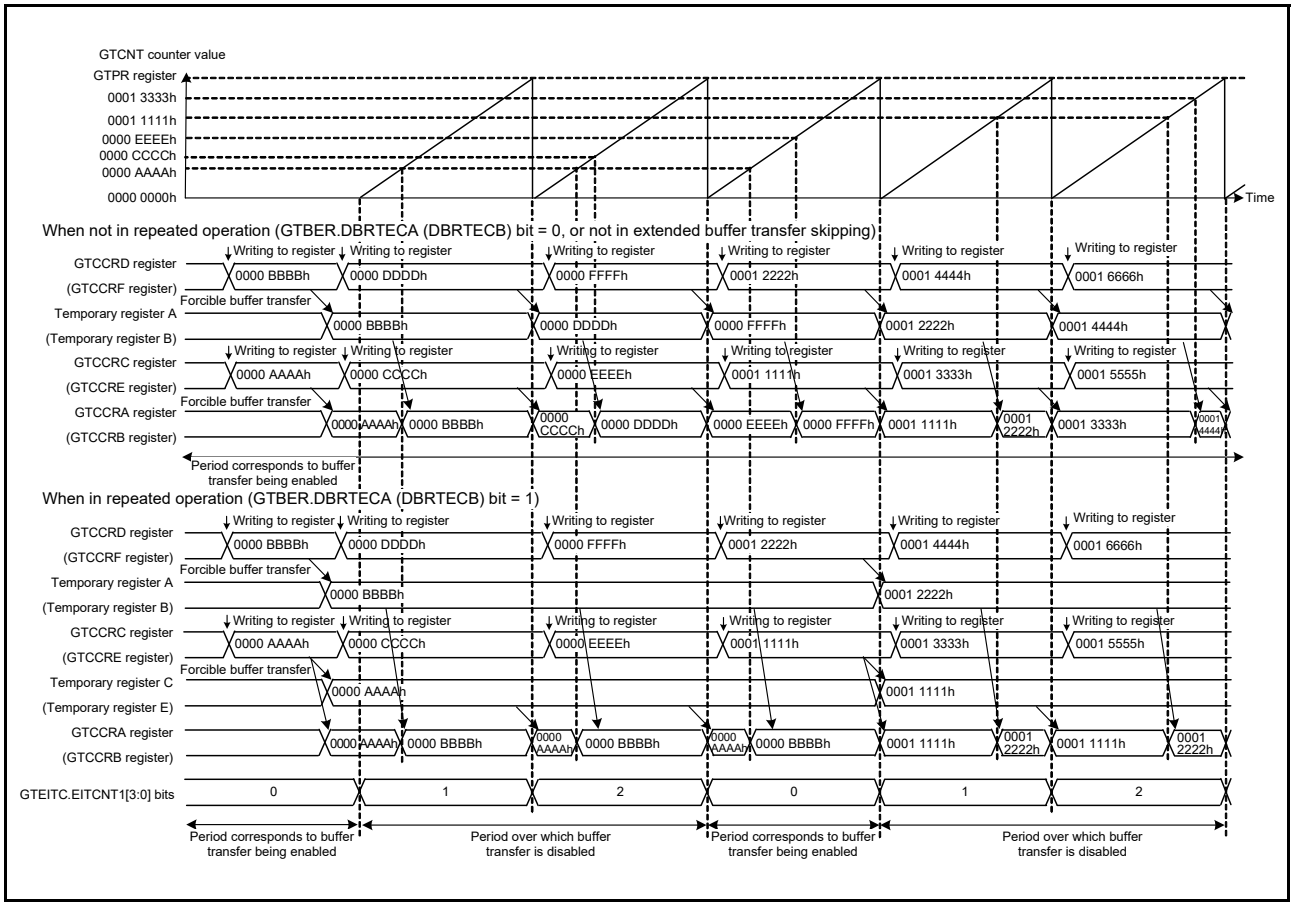


Figure 24.198 Example of Repeated Double-Buffer Operation When GTCCR Buffer Transfer is Disabled (Sawtooth-Wave One-Shot Pulse Mode, Using Extended Buffer Transfer Skipping, GTBER.BD[0] is Constantly 0)

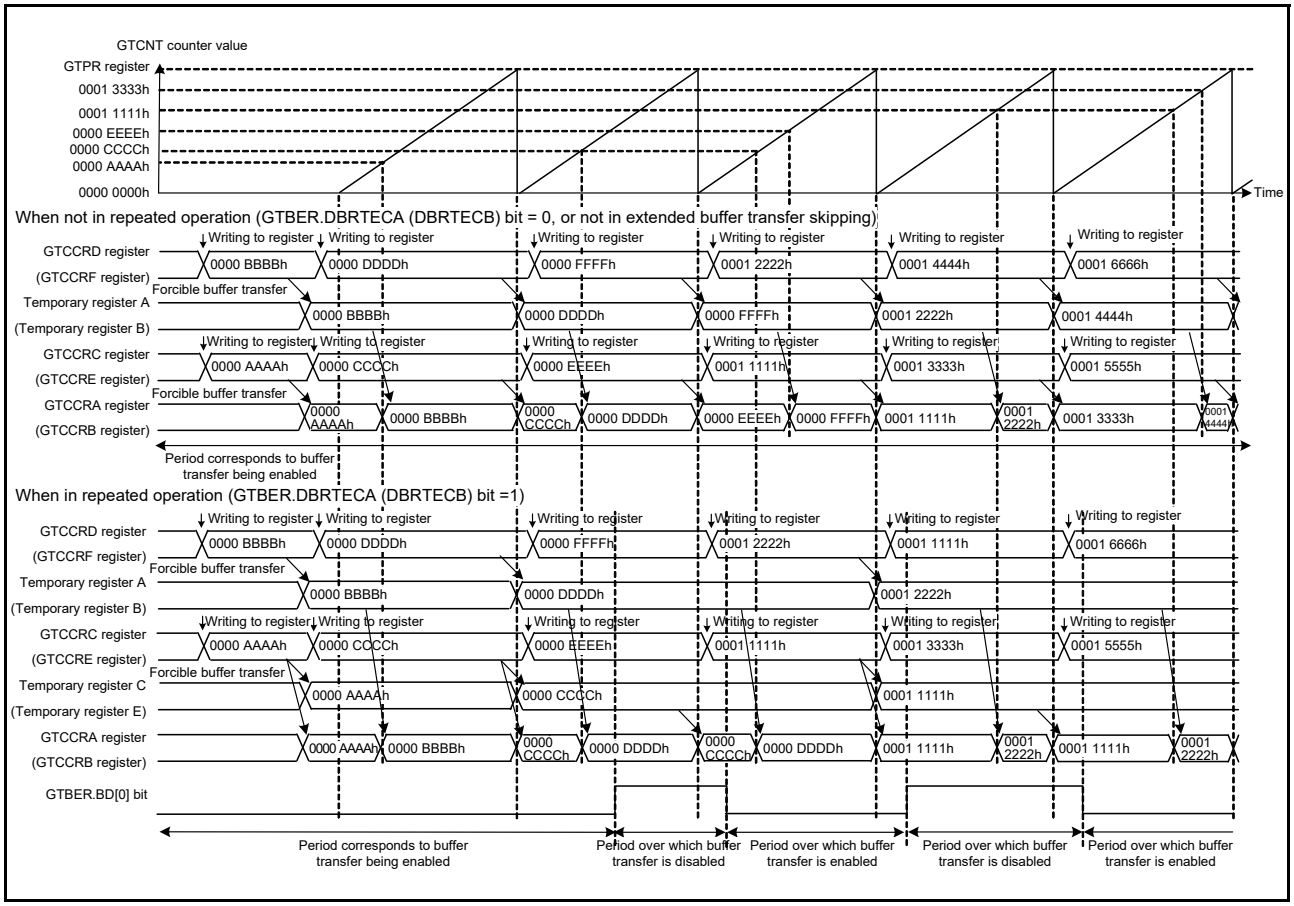


Figure 24.199 Example of Repeated Double-Buffer Operation When GTCCR Buffer Transfer is Disabled (Sawtooth-Wave One-Shot Pulse Mode, Updating the GTBER.BD[0] Bit)

(2) In triangle-wave PWM mode 3

In triangle-wave PWM mode 3, the compare match value for the first half of a waveform is stored in the temporary register x (x = C, E) as the intermediate buffer for the GTCCRx (x = C, E) register and the compare match value for the second half of a waveform is stored in temporary register m (m = A, B) as the intermediate buffer for the GTCCRy (y = D, F) register, respectively, for compare match values during repeated buffer operation, and the given values are alternately transferred to the GTCCRm (m = A, B) register.

Table 24.42 lists the types of buffer transfer of the GTCCR register during counting operation in triangle-wave PWM mode 3.

While counting is stopped, the setting of the value in the temporary register is transferred through forcible buffer transfer. In forcible buffer transfer, the values of the GTCCRy (y = D, F) registers are transferred to temporary registers m (m = A, B), and the values of the GTCCRx (x = C, E) are transferred to temporary registers x (x = C, E), when the setting of the corresponding GTBER.DBRTECm (m = A, B) bit is 1, respectively.

When the setting of the GTBER.DBRTECm (m = A, B) bit is set to 1, values written by the CPU to the GTCCRm (m = A, B) registers are reflected as the values of temporary registers x (x = C, E).

Table 24.42 GTCCR Buffer Operation in Triangle-Wave PWM Mode 3

		Timing of transfer				
GTBER.DBRTECm	**Buffer transfer**	**GTCCRx ↓ GTCCRm**	**GTCCRx ↓ Temporary register x**	**Temporary register x ↓ GTCCRm**	**GTCCRy ↓ Temporary register m**	**Temporary register m ↓ GTCCRm**
0	Transfer-enabled period	Trough	No transfer	No transfer	Trough	Crest
	Transfer-disabled period	No transfer	No transfer	No transfer	No transfer	No transfer
1	Transfer-enabled period	Trough	Trough	No transfer	Trough	Crest
	Transfer-disabled period	No transfer	No transfer	Trough	No transfer	Crest

Figure 24.200 shows the operation of generating transfer-disabled period by extended buffer transfer skipping as an example of repeated double buffer operations when the GTCCR buffer transfer disabled in triangle-wave PWM mode 3.

Figure 24.201 shows the operation of generating transfer-disabled period by updating the GTBER.BD[0] bit as an example of repeated double-buffer operations when the GTCCR buffer transfer is disabled in triangle-wave PWM mode 3.

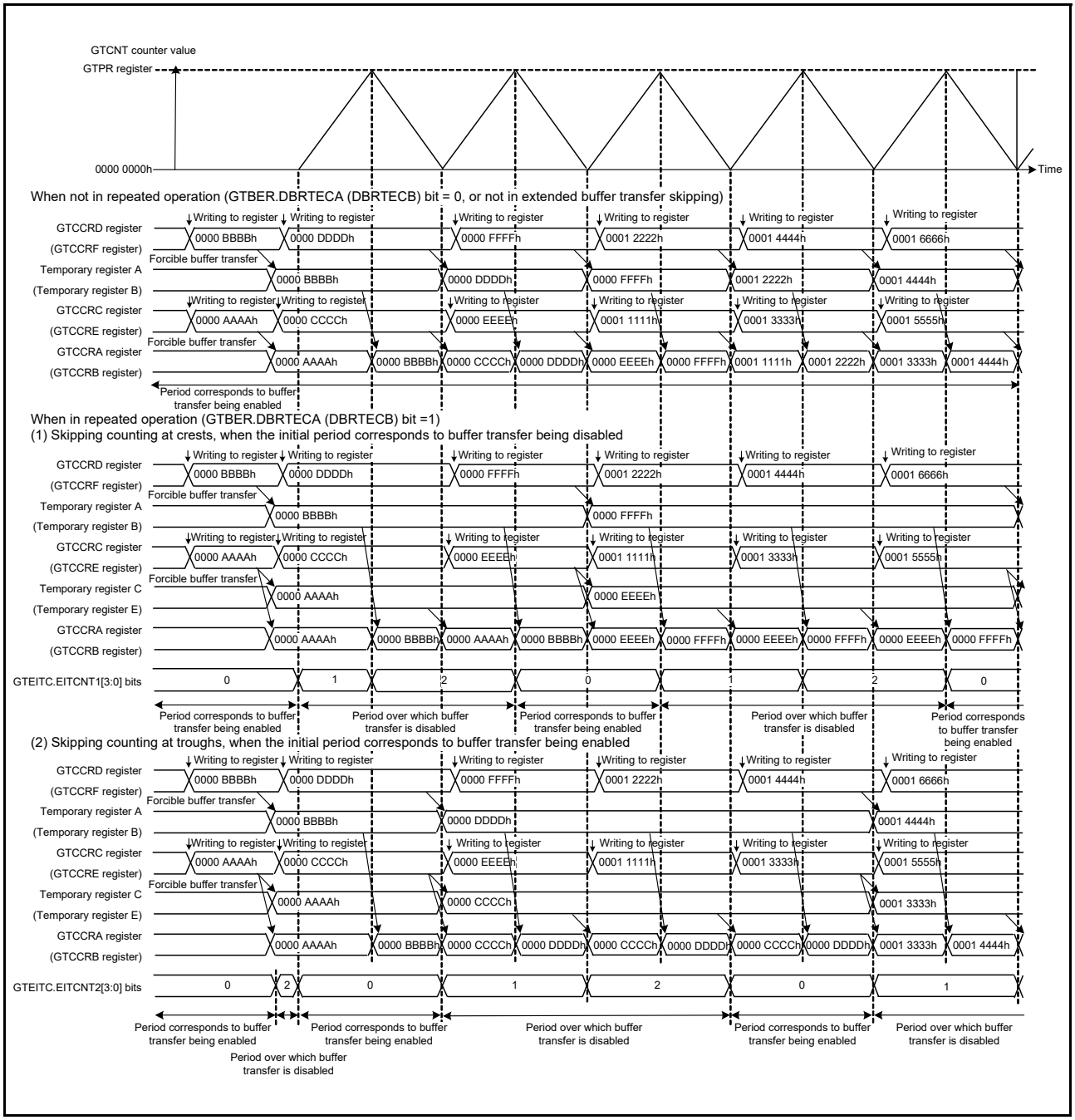


Figure 24.200 Example of Repeated Double-Buffer Operation When GTCCR Buffer Transfer is Disabled (Triangle-Wave PWM Mode 3, Using Extended Buffer Transfer Skipping, GTBER.BD[0] is Constantly 0)

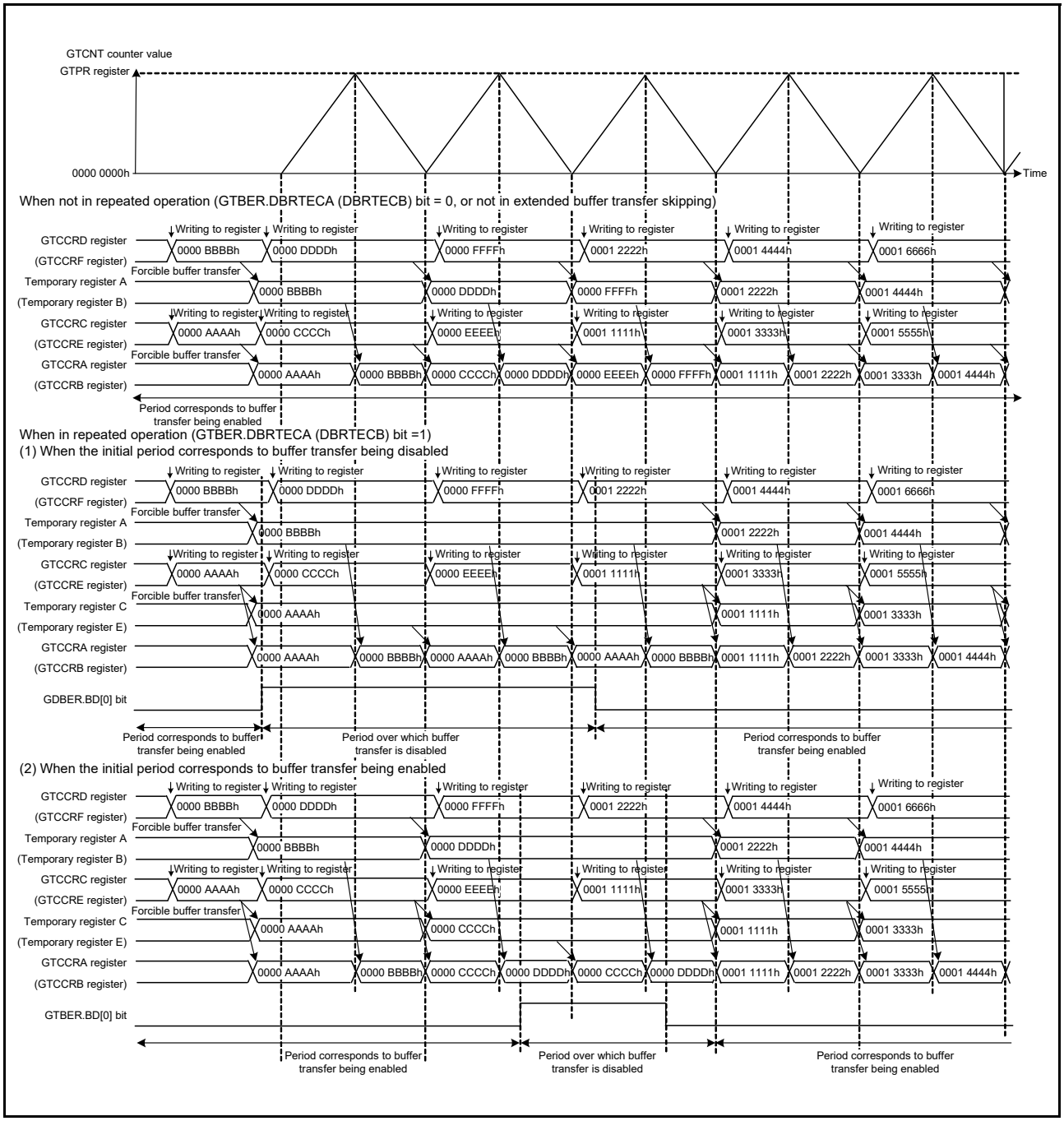


Figure 24.201 Example of Repeated Double-Buffer Operation When GTCCR Buffer Transfer is Disabled (Triangle-Wave PWM Mode 3, Updating the GTBER.BD[0] Bit)

24.8.3 GTIOCnm Pin Output Negate Control (n = 0 to 7; m = A, B)

For protection from system failure, the output negate control which changes the GTIOCnm pin output forcibly by the output stop request from the POEG is provided.

Output protection is required when a dead-time error or the same output level being on the GTIOCnA and GTIOCnB pins is detected. At that time, the detection of disabling of output is output to the group of the POEG set in the GTINTAD.GRP[1:0] bits based on the setting of the output disable detection enable bits, i.e. the GRPDTE, GRPABH, and GRPABL bits in the GTINTAD register. The POEG takes the logical OR of detection of the types described above and other forms of detection of the disabling of output, and outputs requests for the disabling of output to the GPTW.

By setting the GTINTAD.GRP[1:0] bits, one output stop request can be selected out of four output stop requests which are input from the POEG as output stop request signal common in the GTIOCnA and GTIOCnB pins. Selected output stop request can be checked by reading the GTST.ODF flag.

The states of outputs at the time of control for their negation can be set with the GTIOR.OADF[1:0] bits for the GTIOCnA pin and with the GTIOR.OBDF[1:0] bits for the GTIOCnB pin.

Transition to the output negate condition by generating the output stop request from the POEG is performed asynchronously, while a release from the output negate condition by making the output stop request no longer satisfied is performed at the end of the cycle. It is after 3 PCLKC at shortest when the output negate condition is released after the output stop request becomes no longer satisfied. To reliably control output negation, clear the flag of POEG for which the condition for the request to disable the output is no longer satisfied after 4 cycles of PCLKC.

To release the output stop condition during event count operation, in sawtooth-wave PWM mode 2, or without waiting the end of the cycle, set the OADF[1:0] bits to 00b for the GTIOCnA pin, and set the OBDF[1:0] bits to 00b for the GTIOCnB pin.

Figure 24.202 shows the output negate control operation for the GTIOCnm pin output.

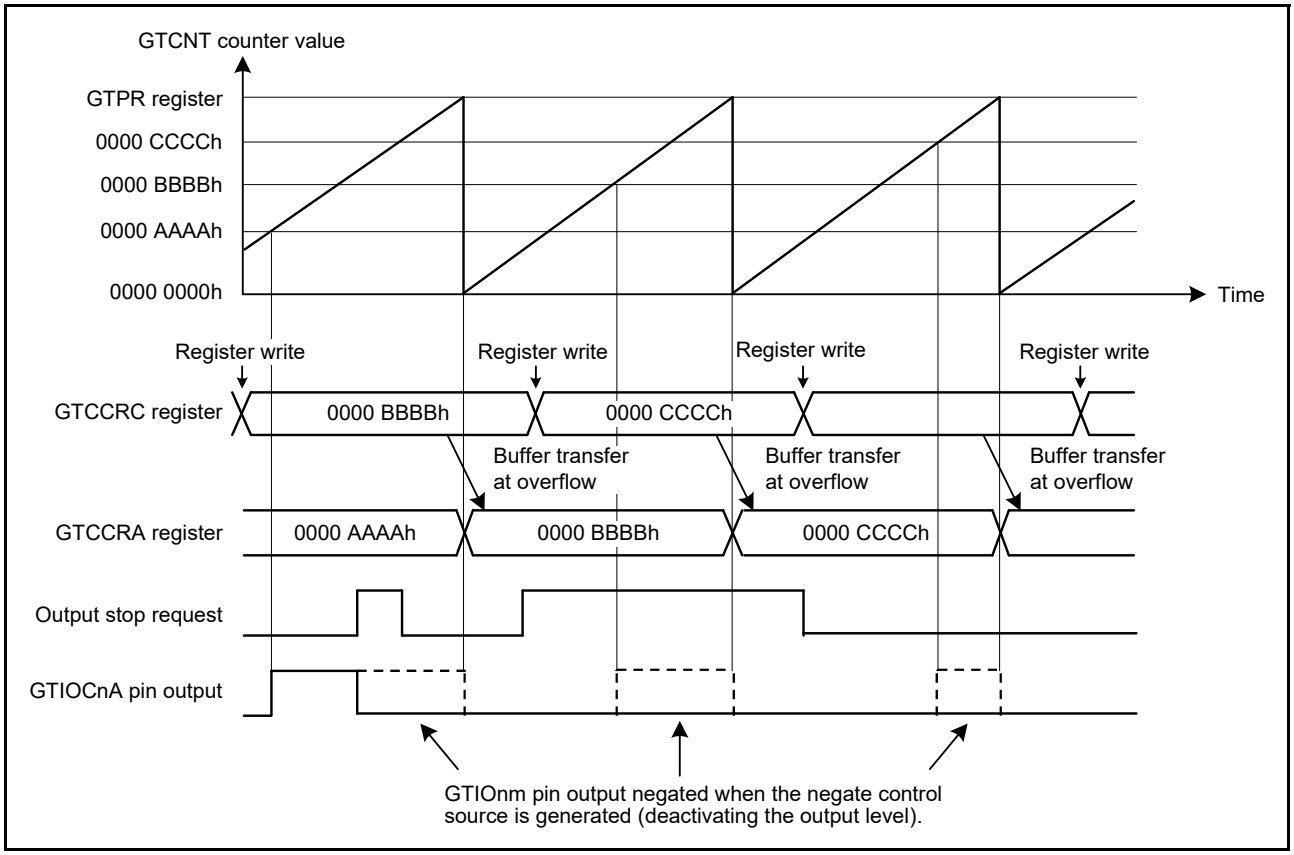


Figure 24.202 Example of Operation for GTIOCnm Pin Output Negate Control (Sawtooth-Wave Up-Counting, Buffer Operation, Active Level: High Output at GTCCRA Register Compare Match, Low Output at the End of the Cycle) and Low Output at Output Negate) (n = 0 to 7; m = A, B)

24.8.4 Output Protection Function for GTIOCnm Pin Output (n = 0 to 7; m = A, B)

To prepare for a case when an incorrect value (0000 0000h or a value greater than or equal to the GTPR register value) is set in the GTCCRA register, the output protection function for the GTIOCnm pin output (disabling function) is activated when the automatic dead time is set (GTDTCR.TDE bit = 1) in triangle-wave PWM mode.
The status of the output protection function can be read from the GTSOS.SOS[1:0] bits.
Figure 24.203 shows the output protection function state transition.

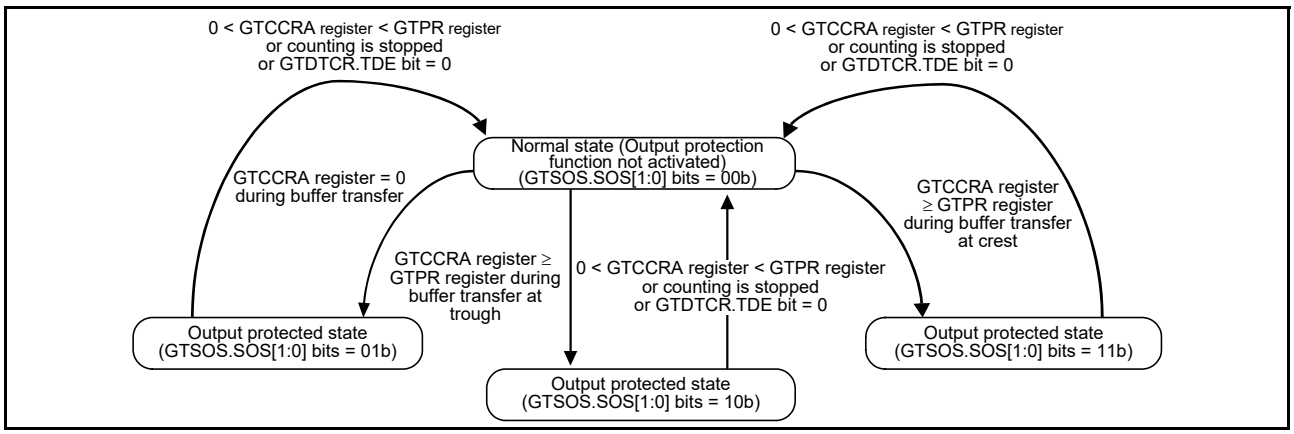
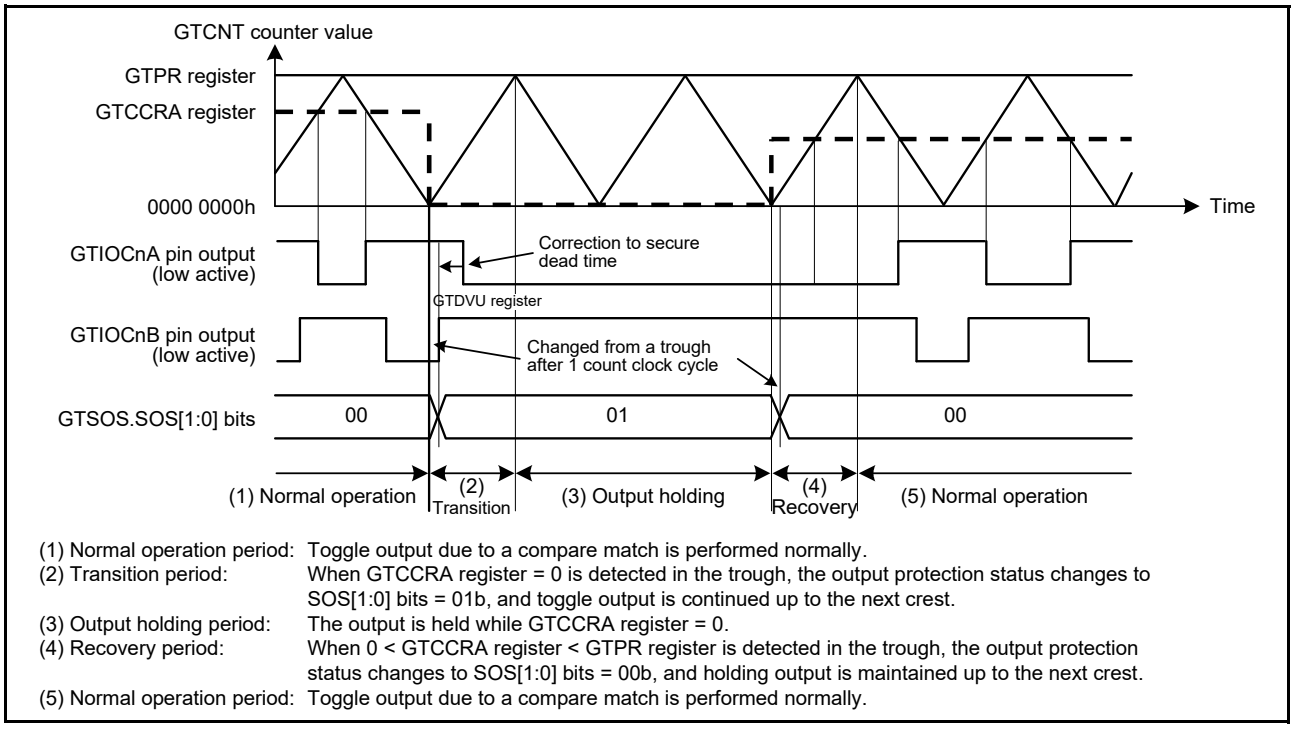


Figure 24.203 Output Protection Function

(1) Output Protection Function when the GTCCRA Register is Set to 0000 0000h during Buffer Transfer

Figure 24.204 and Figure 24.205 show examples of output protection function operation when the GTCCRA register is set to 0000 0000h during buffer transfer at troughs, and Figure 24.206 and Figure 24.207 show examples when the GTCCRA register is set to 0000 0000h during buffer transfer at crests.



(1) Normal operation period: Toggle output due to a compare match is performed normally.
(2) Transition period: When GTCCRA register = 0 is detected in the trough, the output protection status changes to SOS[1:0] bits = 01b, and toggle output is continued up to the next crest.
(3) Output holding period: The output is held while GTCCRA register = 0.
(4) Recovery period: When 0 < GTCCRA register < GTPR register is detected in the trough, the output protection status changes to SOS[1:0] bits = 00b, and holding output is maintained up to the next crest.
(5) Normal operation period: Toggle output due to a compare match is performed normally.

Figure 24.204 Example of Output Protection Function Operation When the GTCCRA Register is Set to 0000 0000h during Buffer Transfer at Troughs (Restored to 0 < GTCCRA Register < GTPR Register during Buffer Transfer at Troughs, Active Level: Low) (n = 0 to 7)

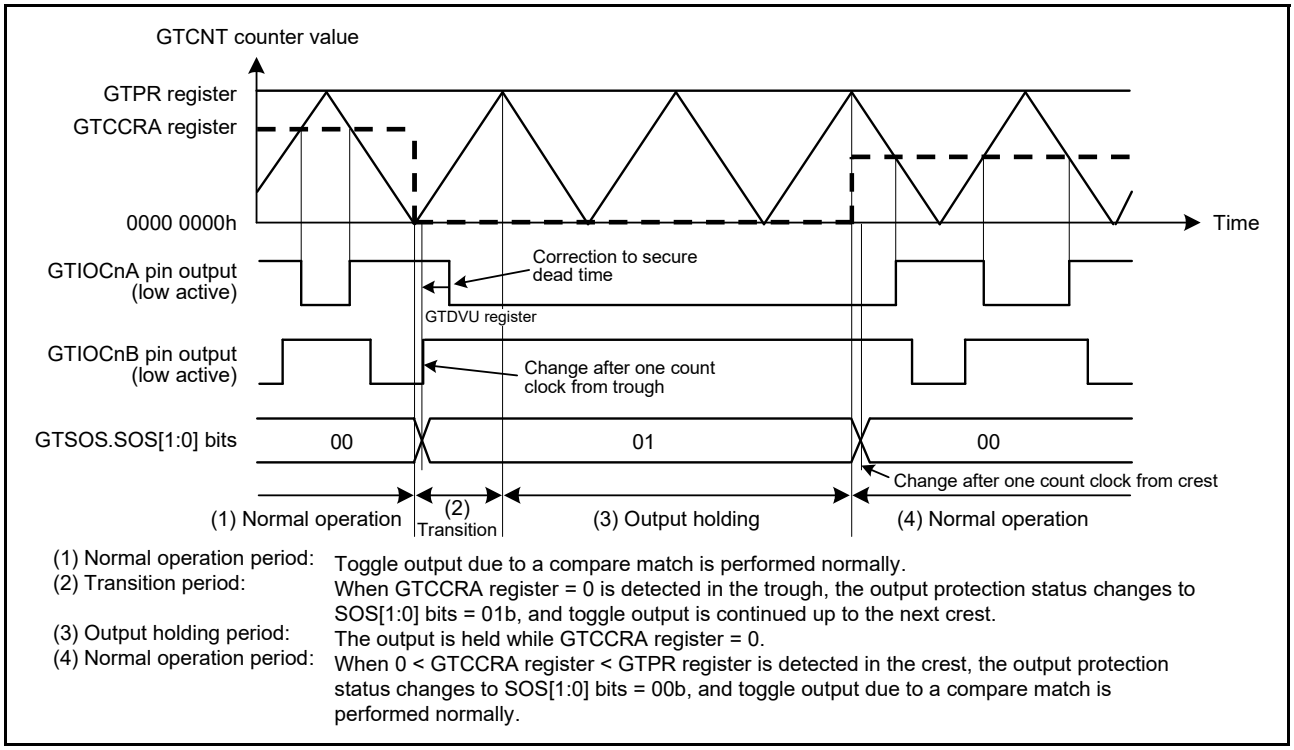


Figure 24.205 Example of Output Protection Function Operation When the GTCCRA Register is Set to 0000 0000h during Buffer Transfer at Troughs (Restored to 0 < GTCCRA Register < GTPR Register during Buffer Transfer at Crests, Active Level: Low) (n = 0 to 7)

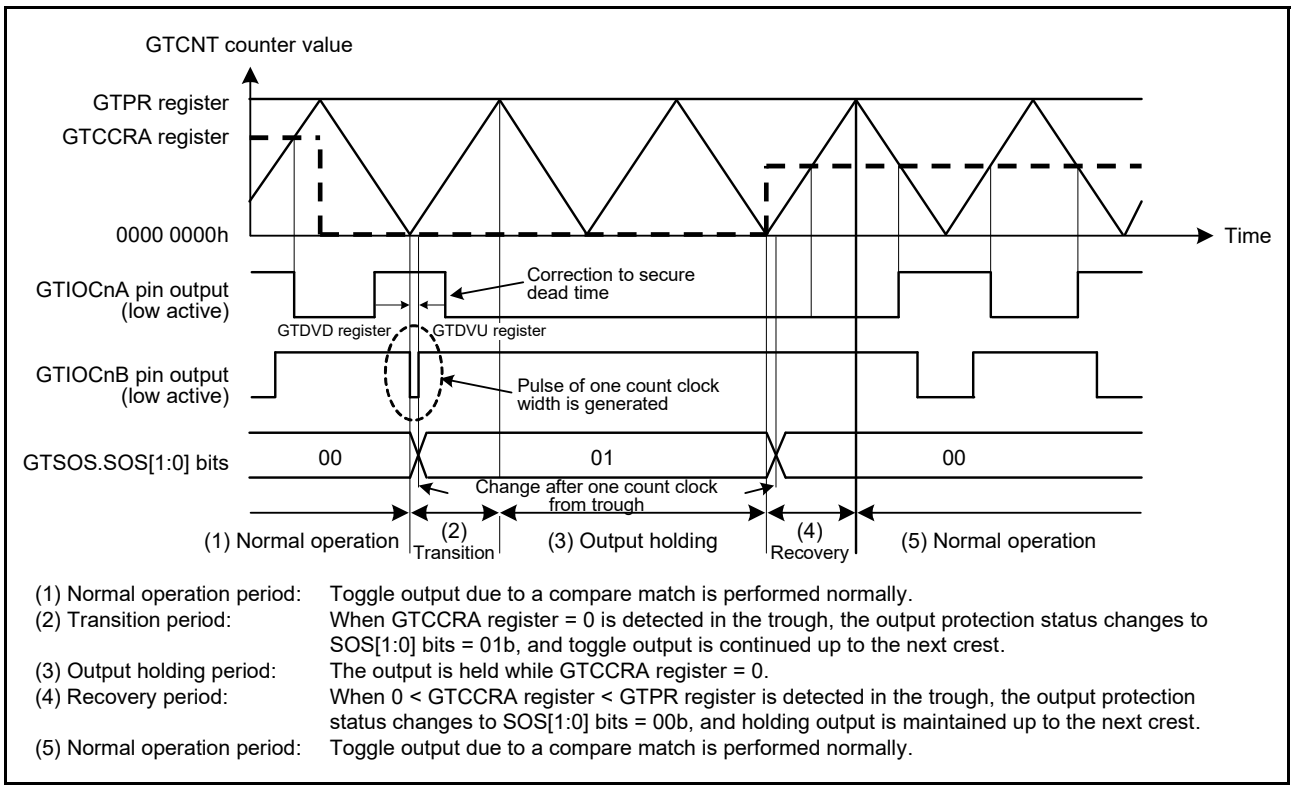


Figure 24.206 Example of Output Protection Function Operation When the GTCCRA Register is Set to 0000 0000h during Buffer Transfer at Crests (Restored to 0 < GTCCRA Register < GTPR Register during Buffer Transfer at Troughs, Active Level: Low) (n = 0 to 7)

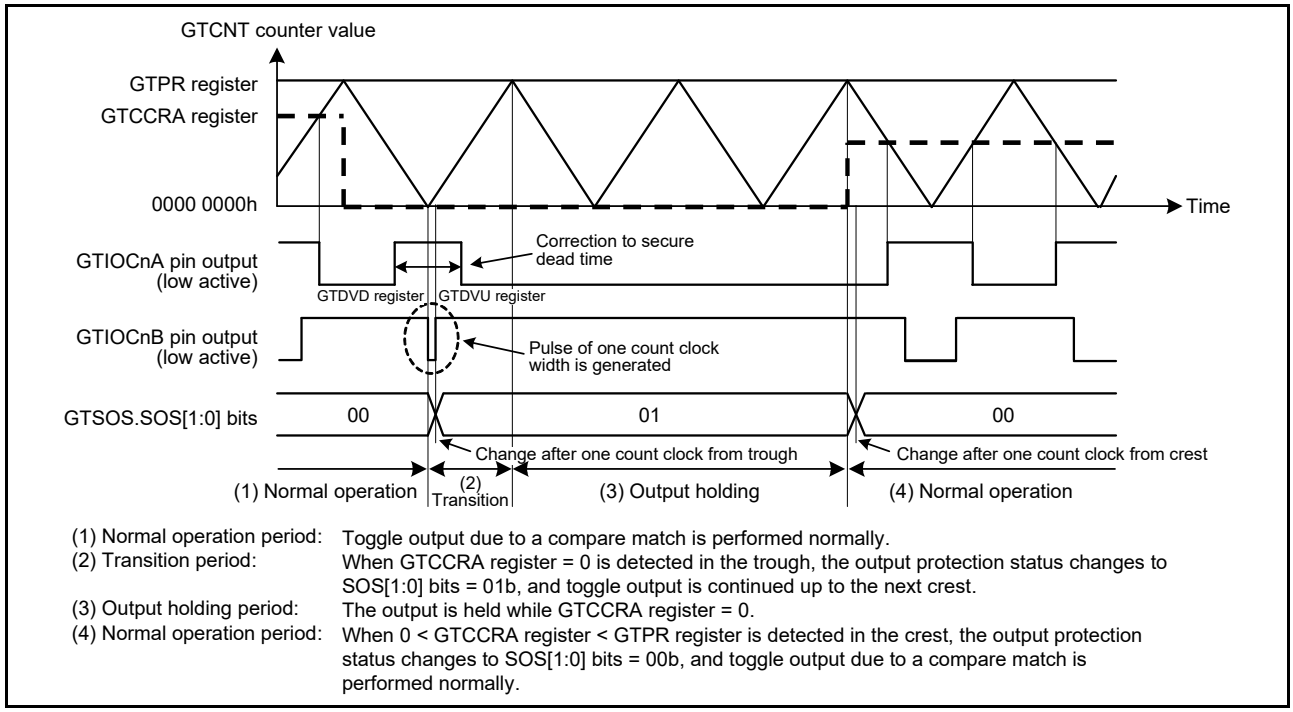


Figure 24.207 Example of Output Protection Function Operation When the GTCCRA Register is Set to 0000 0000h during Buffer Transfer at Crests (Restored to 0 < GTCCRA Register < GTPR Register during Buffer Transfer at Crests, Active Level: Low) (n = 0 to 7)

(2) Output Protection Function when GTCCRA Register ≥ GTPR Register is Set during Buffer Transfer at Troughs

Figure 24.208 and Figure 24.209 show examples of output protection function operation when GTCCRA register ≥ GTPR register is set during buffer transfer at troughs.

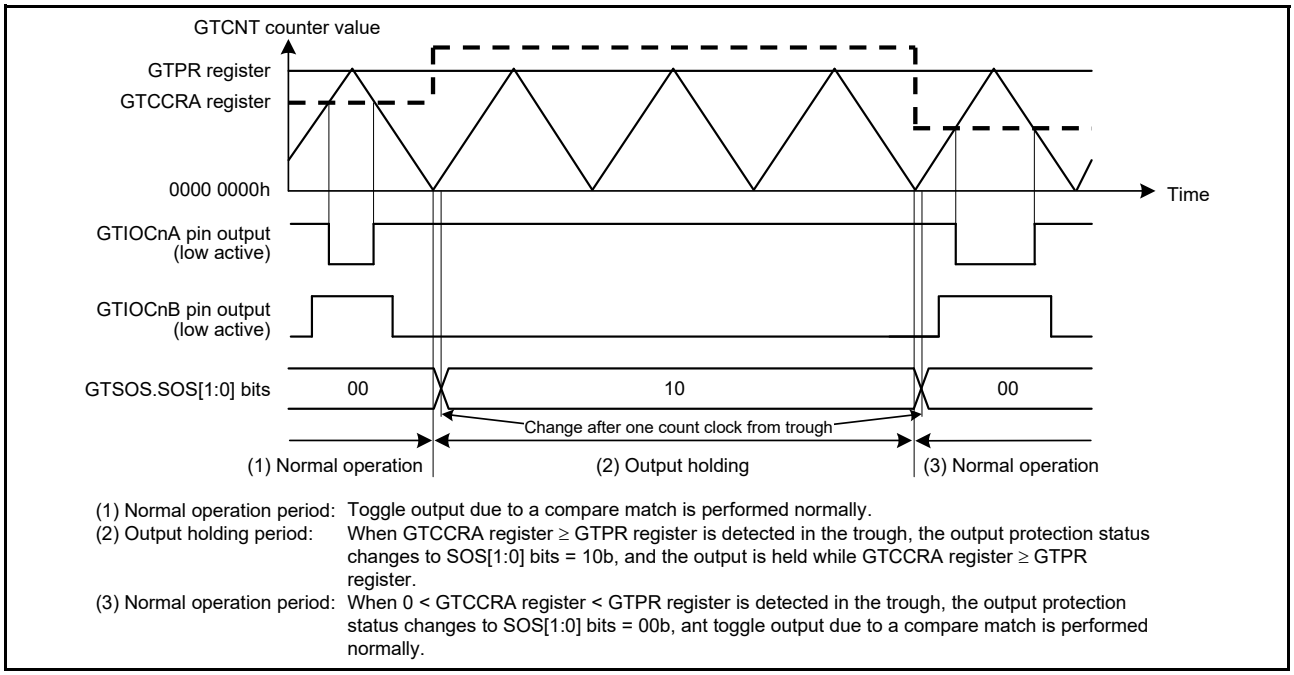
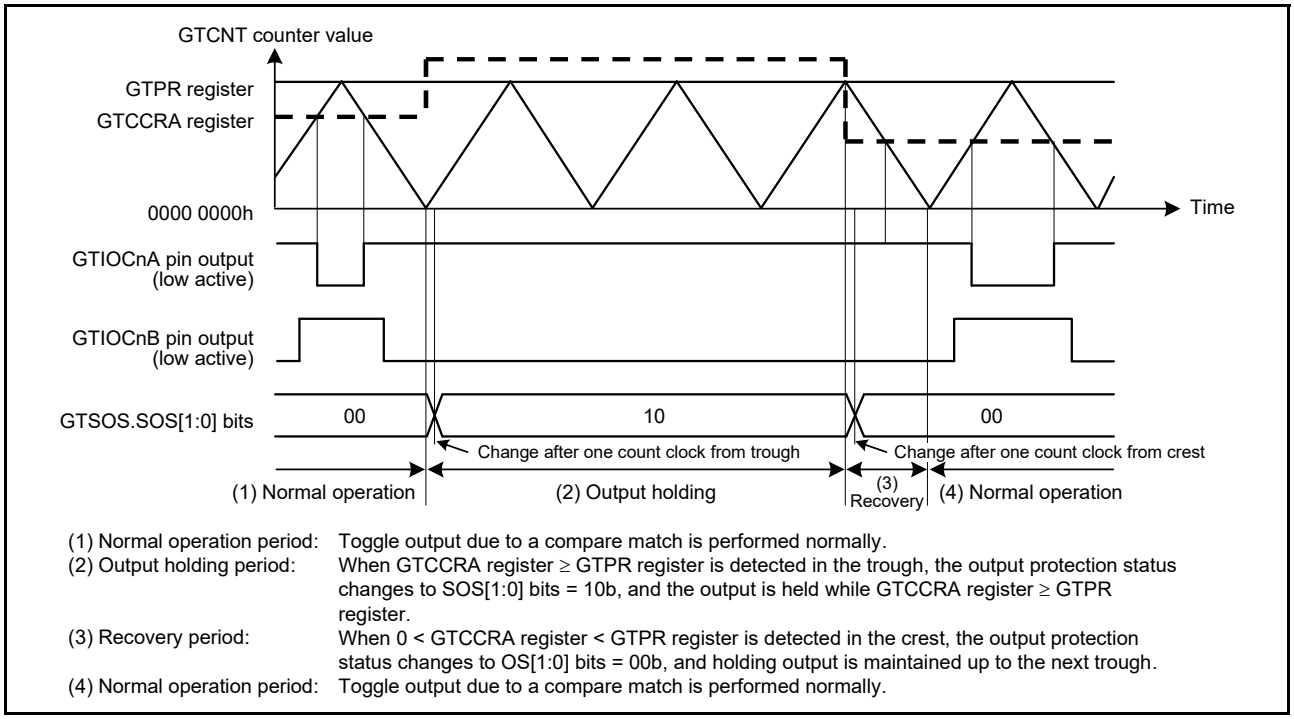


Figure 24.208 Example of Output Protection Function Operation When GTCCRA Register ≥ GTPR Register is set during Buffer Transfer at Troughs (Restored to 0 < GTCCRA Register < GTPR Register during Buffer Transfer at Troughs, Active Level: Low) (n = 0 to 7)

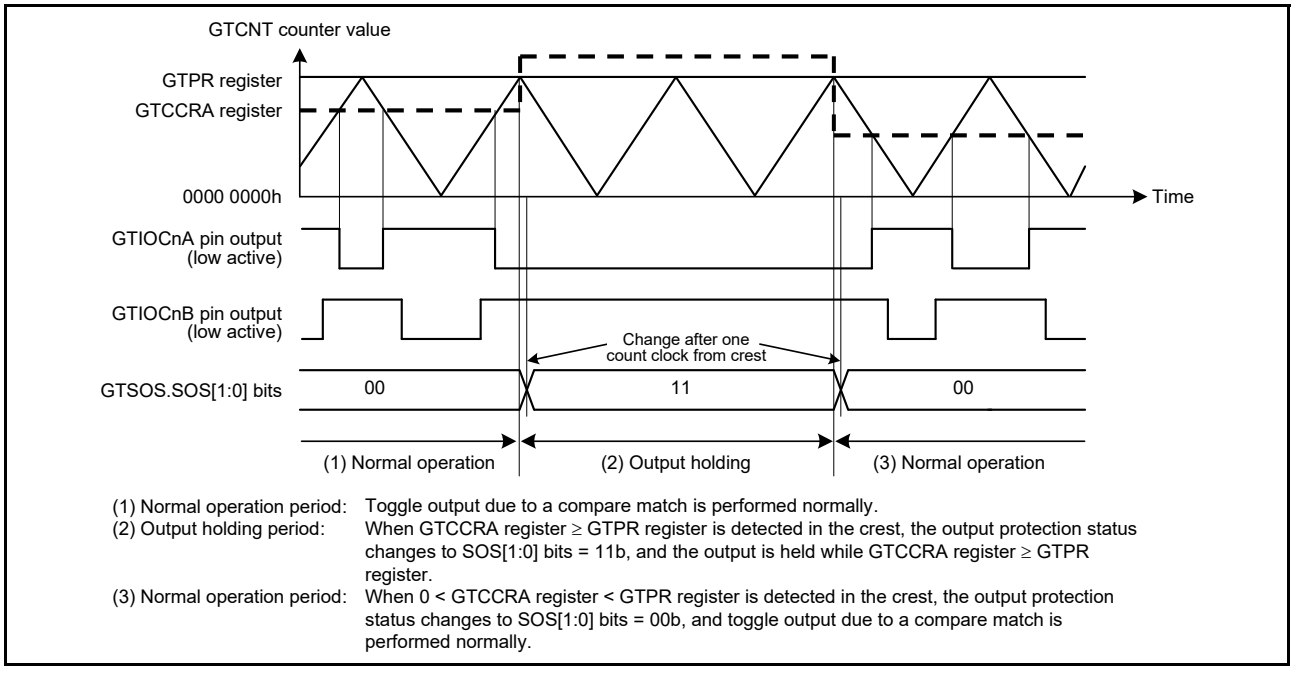


(1) Normal operation period: Toggle output due to a compare match is performed normally.
(2) Output holding period: When GTCCRA register ≥ GTPR register is detected in the trough, the output protection status changes to SOS[1:0] bits = 10b, and the output is held while GTCCRA register ≥ GTPR register.
(3) Recovery period: When 0 < GTCCRA register < GTPR register is detected in the crest, the output protection status changes to OS[1:0] bits = 00b, and holding output is maintained up to the next trough.
(4) Normal operation period: Toggle output due to a compare match is performed normally.

Figure 24.209 Example of Output Protection Function Operation When GTCCRA Register ≥ GTPR Register is Set during Buffer Transfer at Troughs (Restored to 0 < GTCCRA Register < GTPR Register during Buffer Transfer at Crests, Active Level: Low) (n = 0 to 7)

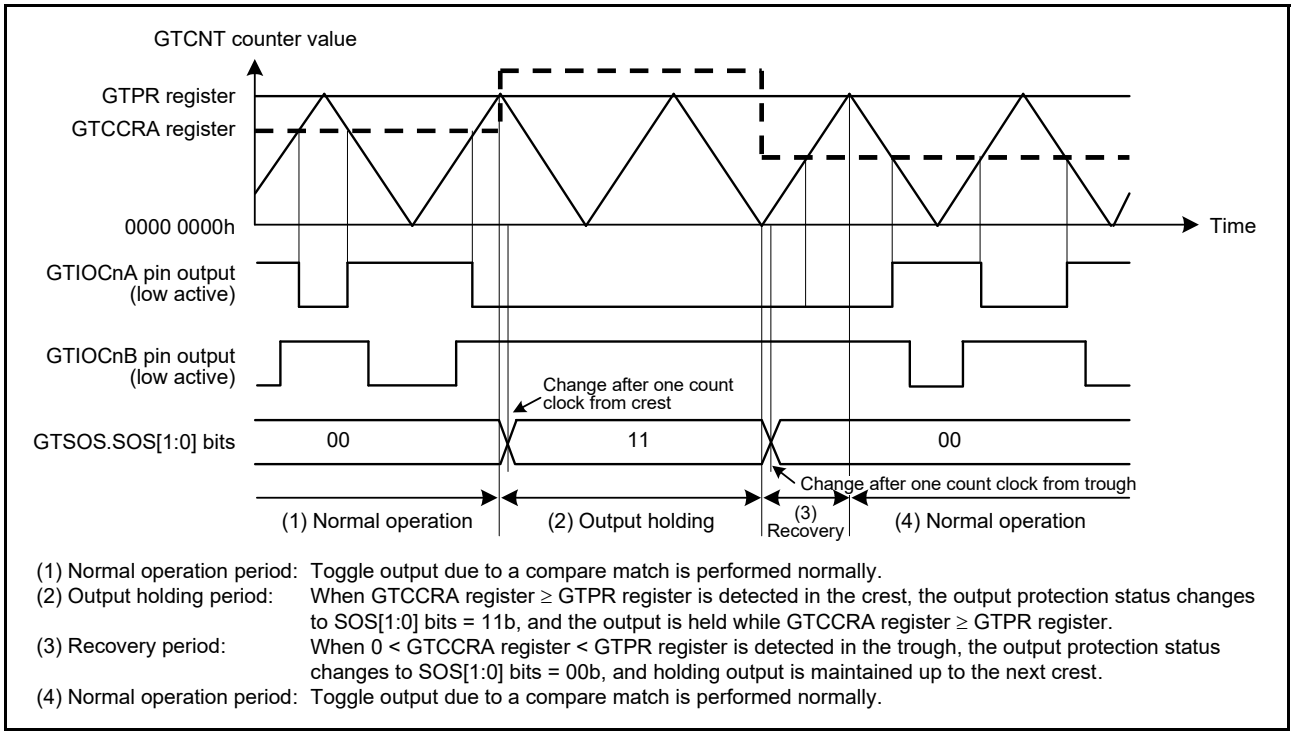
(3) Output Protection Function when GTCCRA Register ≥ GTPR Register is Set during Buffer Transfer at Crests

Figure 24.210 and Figure 24.211 show examples of output protection function operation when GTCCRA register ≥ GTPR register is set during buffer transfer at crests.



(1) Normal operation period: Toggle output due to a compare match is performed normally.
(2) Output holding period: When GTCCRA register ≥ GTPR register is detected in the crest, the output protection status changes to SOS[1:0] bits = 11b, and the output is held while GTCCRA register ≥ GTPR register.
(3) Normal operation period: When 0 < GTCCRA register < GTPR register is detected in the crest, the output protection status changes to SOS[1:0] bits = 00b, and toggle output due to a compare match is performed normally.

Figure 24.210 Example of Output Protection Function Operation When GTCCRA Register ≥ GTPR Register is Set during Buffer Transfer at Crests (Restored to 0 < GTCCRA Register < GTPR Register during Buffer Transfer at Crests, Active Level: Low) (n = 0 to 7)



(1) Normal operation period: Toggle output due to a compare match is performed normally.
(2) Output holding period: When GTCCRA register ≥ GTPR register is detected in the crest, the output protection status changes to SOS[1:0] bits = 11b, and the output is held while GTCCRA register ≥ GTPR register.
(3) Recovery period: When 0 < GTCCRA register < GTPR register is detected in the trough, the output protection status changes to SOS[1:0] bits = 00b, and holding output is maintained up to the next crest.
(4) Normal operation period: Toggle output due to a compare match is performed normally.

Figure 24.211 Example of Output Protection Function Operation When GTCCRA Register ≥ GTPR Register is Set during Buffer Transfer at Crests (Restored to 0 < GTCCRA Register < GTPR Register during Buffer Transfer at Troughs, Active Level: Low) (n = 0 to 7)

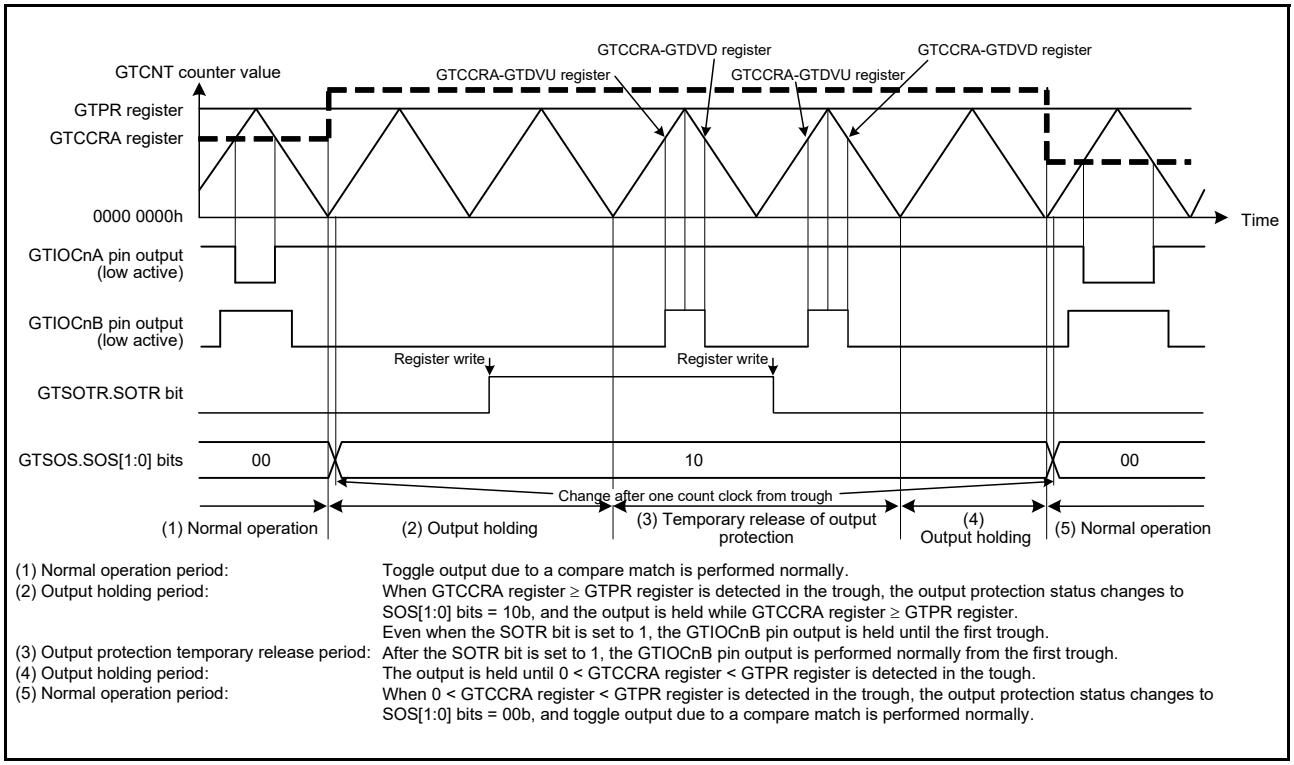
(4) Restricted Specification of Output Protection Function

Even if an incorrect value (0000 0000h or a value greater than or equal to the GTPR register value) is set in the GTCCRA register during count operation, the output protection functions in a specific way such that one of the positive- and negative-phase outputs becomes non-active. However, the output protection does not operate normally.

- When the GTCCRA register value at the start of count operation is greater than 0000 0000h, and less than the setting value of the GTPR register

(5) Temporary Release of Output Protection Function

When the GTSOS.SOS[1:0] bits = 10b (protected state in which GTCCRA register ≥ GTPR register has occurred during transfer at trough), the protected state of the GTIOCnB pin output can be temporarily released by setting the GTSOTR.SOTR bit to 1. The SOS[1:0] bits retain 10b even if the output protection function is released.
When the SOTR bit is set to 0, the GTIOCnB pin output protection can be restarted.
Figure 24.212 shows an example of the operation of temporary release of output protection when the setting of the GTCCRA register ≥ GTPR register during buffer transfer at troughs.



(1) Normal operation period: Toggle output due to a compare match is performed normally.
(2) Output holding period: When GTCCRA register ≥ GTPR register is detected in the trough, the output protection status changes to SOS[1:0] bits = 10b, and the output is held while GTCCRA register ≥ GTPR register.
Even when the SOTR bit is set to 1, the GTIOCnB pin output is held until the first trough.
(3) Output protection temporary release period: After the SOTR bit is set to 1, the GTIOCnB pin output is performed normally from the first trough.
(4) Output holding period: The output is held until 0 < GTCCRA register < GTPR register is detected in the tough.
(5) Normal operation period: When 0 < GTCCRA register < GTPR register is detected in the trough, the output protection status changes to SOS[1:0] bits = 00b, and toggle output due to a compare match is performed normally.

Figure 24.212 Example of the Operation of Temporary Release of Output Protection When the Setting of the GTCCRA Register ≥ GTPR Register during Buffer Transfer at Troughs (Restored to 0 < GTCCRA Register < GTPR Register during Buffer Transfer at Troughs, Active Level: Low) (n = 0 to 7)

24.9 Initialization Method of Output Pins

24.9.1 Pin Settings after Reset

The GPTW registers are initialized at a reset. Initialize the GPTW for output to the external pins by making the port mode settings and setting the OAE and OBE bits in the GTIOR register, and then start the counter counting.

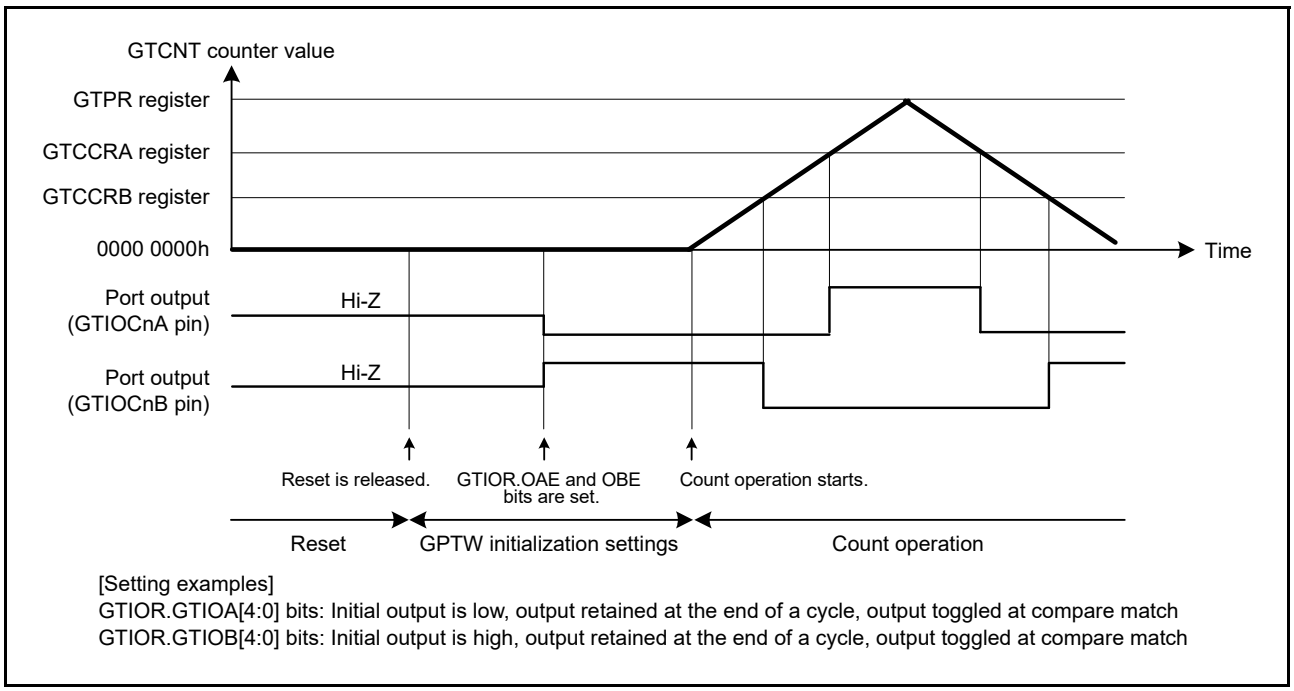


Figure 24.213 Example of Pin Settings after Reset (n = 0 to 7)

24.9.2 Pin Initialization Due to Error during Operation

If an error occurs during GPTW operation, the following four types of pin processing can be performed before pin initialization.

(1) Set OAHLD and OBHLD bits in the GTIOR register to 1 and retain the outputs at count stop.
(2) Set OAHLD and OBHLD bits to 0, specify arbitrary output values at OADFLT and OBDFLT bits in the GTIOR register, and output the arbitrary values at count stop.
(3) Set the pin to output an arbitrary value as a general output port by setting the PDR, PODR, and PMR registers of the I/O port in advance. Set the OAE and OBE bits in the GTIOR register to 0 and the control bit in PMR register that corresponds to the pin to 0 to allow the arbitrary values to be output from the pin set as a general output port when an error occurs.
(4) Drive the output to a high impedance state using the POEG function.

When automatic dead time setting has been made, set the GTDTCR.TDE bit to 0 once after counting is stopped.
When counting is stopped, only the values of registers that are changed by a GPTW external source will change. If counting is resumed, operation will carry on from where it was stopped.
If counting was stopped, registers should be initialized before counting is started.

24.10 Usage Notes

24.10.1 Module Stop Function Setting

Operation of the GPTW can be disabled or enabled by the module stop control register. The initial setting is for operation of the GPTW to be halted. Register access is enabled by clearing module stop state. For details, see section 11, Low Power Consumption.

24.10.2 Settings of the GTCCRm Register during Compare Match Operation (m = A to F)

(1) When Automatic Dead Time Setting has been Made in Triangle-Wave PWM Mode

The GTCCRA register should satisfy the following conditions:

GTCCRA register > GTDVU register,
GTCCRA register > GTDVD register, and
GTCCRA register < GTPR register.

When the GTCCRA register is set to 0000 0000h or a value greater than or equal to the GTPR register value during count operation, the output protection function is activated.
However, if the following conditions are not satisfied, the output protection does not operate normally.

- When the GTCCRA register value at the start of count operation is 0000 0001h or greater, and less than the setting value of the GTPR register

For details, refer to section 24.8.4, Output Protection Function for GTIOCnm Pin Output (n = 0 to 7; m = A, B).

(2) When Automatic Dead Time Setting has not been Made in Triangle-Wave PWM Mode

Set a value greater than 0000 0000h, and less than the setting value of the GTPR register in the GTCCRA register. When 0000 0000h or the same value as that of the GTPR register is set in the GTCCRA register, compare match is generated in one cycle only when [GTCCRA register = 0000 0000h] or [GTCCRA register = GTPR register] is met. Furthermore, a value exceeding the setting value of the GTPR register is set in the GTCCRA register, compare match does not occur. Similarly, set a value greater than 0000 0000h, and less than the setting value of the GTPR register in the GTCCRB register. When 0000 0000h or the same value as that of the GTPR register is set in the GTCCRB register, compare match is generated in one cycle only when [GTCCRB register = 0000 0000h] or [GTCCRB register = GTPR register] is met. Furthermore, a value exceeding the setting value of the GTPR register is set in the GTCCRB register, compare match does not occur.

(3) When Automatic Dead Time Setting has been Made in Sawtooth-Wave One-Shot Pulse Mode

The GTCCRC and GTCCRD registers should be set to satisfy the following restrictions. If the restrictions are not satisfied, correct output waveforms with secured dead time may not be obtained.

- In up-counting:
  GTCCRC register < GTCCRD register
  GTCCRC register > GTDVU register
  GTCCRD register < GTPR register – GTDVD register
- In down-counting:
  GTCCRC register > GTCCRD register
  GTCCRC register < GTPR register – GTDVU register
  GTCCRD register > GTDVD register

(4) When Automatic Dead Time Setting has not been Made in Sawtooth-Wave One-Shot Pulse Mode

The GTCCRC and GTCCRD registers should be set to satisfy the following restrictions. If the restrictions are not satisfied, two compare matches do not occur and pulse output cannot be performed.

- In up-counting: 0 < GTCCRC register < GTCCRD register < GTPR register
- In down-counting: GTPR register > GTCCRC register > GTCCRD register > 0

Similarly, GTCCRE and GTCCRF registers should be set to satisfy the following restrictions. If the restrictions are not satisfied, two compare matches do not occur and pulse output cannot be performed.

- In up-counting: 0 < GTCCRE register < GTCCRF register < GTPR register
- In down-counting: GTPR register > GTCCRE register > GTCCRF register > 0

(5) In Sawtooth-Wave PWM Mode

The GTCCRA register should be set with the range of 0000 0000h < GTCCRA register < GTPR register. If GTCCRA register = 0000 0000h or GTCCRA register = GTPR register is set, a compare match occurs within the cycle only when GTCCRA register = 0000 0000h or GTCCRA register = GTPR register is satisfied. If GTCCRA register > GTPR register is set, no compare match occurs.
Similarly, GTCCRB register should be set with the range of 0000 0000h < GTCCRB register < GTPR register. If GTCCRB register = 0000 0000h or GTCCRB register = GTPR register is set, a compare match occurs within the cycle only when GTCCRB register = 0000 0000h or GTCCRB register = GTPR register is satisfied. If GTCCRB register > GTPR register is set, no compare match occurs.

(6) In Complementary PWM mode 1, 2, 3

The GTCCRn register must be set with the range of 0 ≤ GTCCRn ≤ GTPR + GTDVU.

(7) In Complementary PWM mode 4

In single buffer operation, the GTCCRn register must be set with the range of 0 ≤ GTCCRn ≤ GTPR + GTDVU.
In double buffer operation, the GTCCRn register must be set with the range of GTDVU < GTCCRn < GTPR.

24.10.3 Range of Settings for the GTPBR and GTPDBR Registers in the Complementary PWM Mode

When buffer transfer with the use of the GTPR register is to proceed at the end of a crest section in complementary PWM mode 1, 3, or 4, set the GTPBR and GTPDBR registers so that the GTPR register is within the following range after the transfer.
The range consists of values no less than the GTCNT counter value of the master channel at the end of a crest section, that is, the GTPR register before the transfer – the GTDVU register (GTPBR register ≥ GTPR register – GTDVU register, GTPDBR register ≥ GTPBR register – GTDVU register)
When buffer transfer with the use of the GTPR register is to proceed at the end of a trough section or on counter clearing, no restriction applies to the ranges of settings for the GTPBR and GTPDBR registers.

24.10.4 Setting Range of the GTCNT Counter

Other than the sawtooth-wave PWM mode 2 and complementary PWM mode, set the range of the GTCNT counter within the range of 0 ≤ GTCNT counter ≤ GTPR register.

24.10.5 The GTCNT Counter Start/Stop

The GTCNT counter start/stop control by the GTCR.CST bit is synchronized with the count clock selected by the GTCR.TPCS[3:0] bits. Since the GTCNT counter starts or stops after one count clock cycle selected by the TPCS[3:0] bits following the CST bit updating, events until the GTCNT counter actually starts are ignored, where events may be accepted and the interrupt may be generated after the CST bit becomes 0.

24.10.6 Order of Priority in Events

(1) GTCNT Counter

Order of priority in events to update the GTCNT counter is shown below.

Table 24.43 Order of Priority in Updating GTCNT Counter

The GTCNT counter updating source	Order of priority
CPU writing (GTCNT counter writing/GTCLR register writing)	High
Clearing by a hardware source set by the GTCSR register	↑
Up-counting/down-counting by a hardware source set by the GTUPSR and GTDNSR registers	↑
Count operation	Low

In case of contention between up-counting by the GTUPSR register and down-counting by the GTDNSR register, the counter value is not updated.
When updating of the GTCNT counter conflicts with reading by the CPU, the value before updating is reflected.

(2) The GTCR.CST Bit

In case that start/stop by a hardware source set by the GTSSR and GTPSR registers conflicts with the CPU writing (GTCR register writing/GTSTR register writing /GTSTP register writing), the CPU writing takes priority.
In case that stop by the cycle count function conflicts with start by the CPU writing (GTCR register writing/GTSTR register writing), the cycle count operation is finished with setting the GTST.PCF flag. The CST bit is not changed and the GTCNT continues to count.
In case of contention between start by a hardware source set by the GTSSR register and stop by a hardware source set by the GTPSR register, the state of the CST bit does not change.
In case of contention between the CST bit updating and the CPU reading (GTCR register reading/GTSTR register reading /GTSTP register reading), data before updating is read.

(3) GTCCRm Register (m = A to F)

In case of contention between the GTCCRm register writing and an input capture/buffer transfer, an input capture/buffer transfer takes priority over the GTCCRm register writing.
In case that an input capture conflicts with the CPU writing or the counter updating by a hardware source, the counter value before updating is captured.
In case of contention between the GTCCRm register updating and the CPU reading, data before updating is read.

(4) GTPR Register

In case of contention between a buffer transfer and the GTPR register writing, the GTPR register writing takes priority over a buffer transfer.
In case of contention between the GTPR register updating and the CPU reading, data before updating is read.

(5) GTADTRm Register (m = A, B)

In case of contention between a buffer transfer and the GTADTRm register writing, the GTADTRm register writing takes priority over a buffer transfer.
In case of contention between the GTADTRm register updating and the CPU reading, data before updating is read.

(6) GTDVm Register (m = U, D)

In case of contention between a buffer transfer and the GTDVm register writing, the GTDVm register writing takes priority over a buffer transfer.
In case of contention between the GTDVm register updating and the CPU reading, data before updating is read.

(7) GTIOR.GTIOm bit (m = A, B)

When there is a conflict between buffer transfer operation and writing to GTIOR.GTIOm bit, writing to GTIOR.GTIOm bit has priority over buffer transfer operation.
When there is a conflict between updating the GTIOR.GTIOm bit and reading by the CPU, pre-update data is ready.

24.10.7 Note on Counter Clearing in the Complementary PWM Mode

Counter clearing at the end of a trough section or the end of the initial output section must be avoided in the complementary PWM mode. This can be achieved by using synchronous clearing in response to a match in comparison in another channel set as the source of the trigger, as described in section 24.3.8.3, Synchronous Clear Operation by Inter Channel Cooperation. When using counter clearing other than as stated above, adjust the timing of the operations to avoid counter clearing at the end of a trough section or the end of the initial output section.

24.10.8 Note on Disabling PWM Initial Output After Synchronous Clearing in the Complementary PWM Mode

When initial output on the GTIOCnA and GTIOCnB pins after synchronous clearing in a trough section in the complementary PWM mode has been disabled by setting the GTIOR.CPSCIR bit to 1, the respective values set in the compare match registers (the GTCCRA, GTCCRC, GTCCRD,GTCCRE, and GTCCRF registers) must be more than twice that of the GTDVU register.

25. High Resolution PWM Waveform Generation Circuit (HRPWM)

25.1 Overview

This MCU has a high resolution PWM waveform generation circuit (HRPWM) which finely adjusts a PWM waveform generated by GPTW0 to GPTW3 with a resolution of at least around 260 ps. Using the HRPWM, a digital power supply control with high conversion efficiency can be realized. Table 25.1 lists the specifications of the HRPWM, Figure 25.1 shows a block diagram, and Table 25.2 lists the I/O pins.

Table 25.1 Specifications of the HRPWM

Item	Description
Function	• A maximum of four channels of complementary PWM waveforms are finely adjusted at high resolution. • Using a delay locked loop (DLL) circuit, a high resolution of 1/32 times of a PCLKC period (minimum of around 260 ps). • Timing for rising and falling of the PWM waveform can be adjusted individually.*1 • A waveform generated by the GPTW can be output directly by bypassing the HRPWM.
Operating frequency (f(PCLKC))	80 to 120 MHz

Note 1. Adjustment of timing for rising and falling of PWM signals which are output from the GPTW0 to GPTW3 is realized by a delay of a resolution of 1/32 times of a PCLKC period.

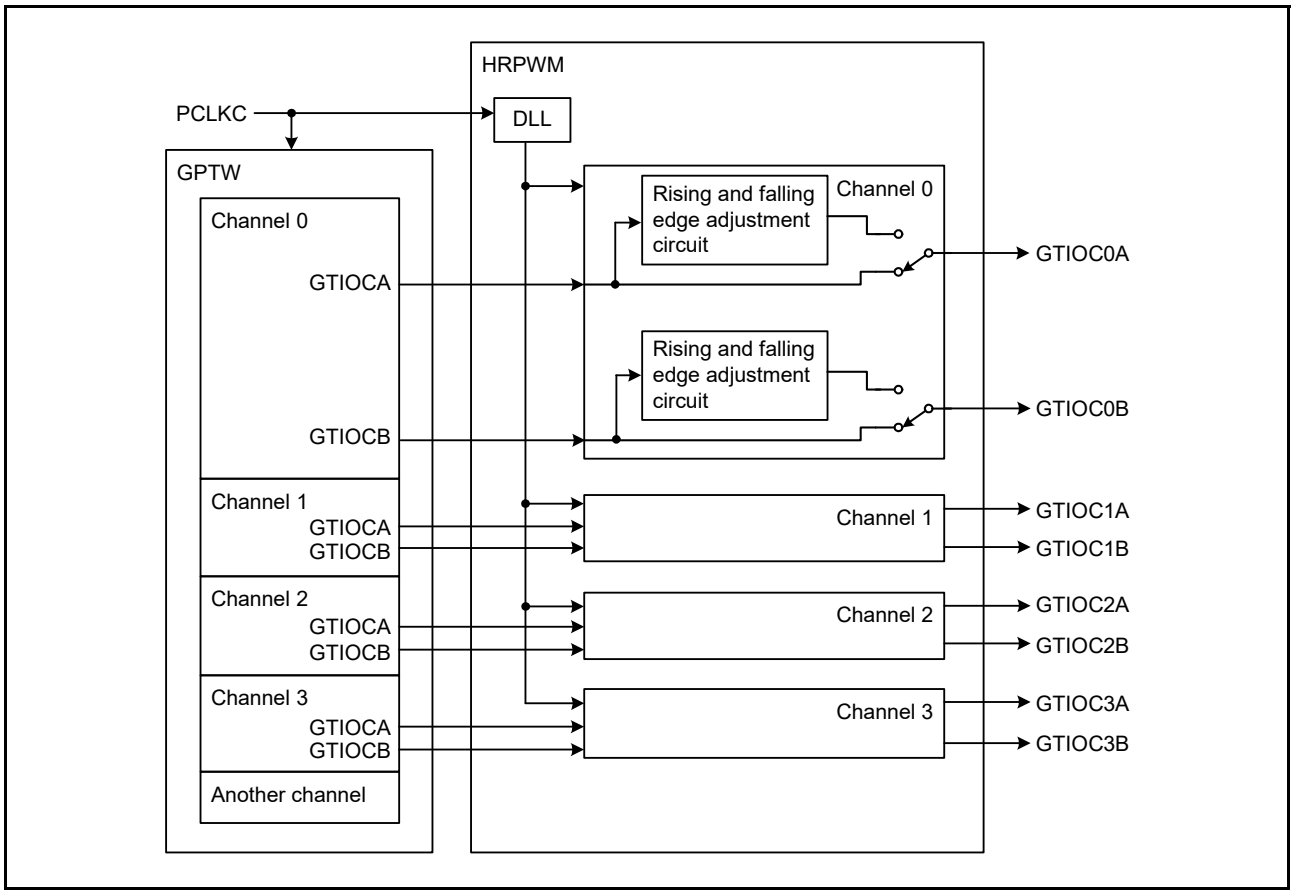


Figure 25.1 Block Diagram of the HRPWM

Table 25.2 I/O Pins of the HRPWM

Pin Name	I/O	Function
GTIOC0A	Output	High resolution output of a PWM waveform generated by the GPTW0
GTIOC0B	Output	High resolution output of a PWM waveform generated by the GPTW0
GTIOC1A	Output	High resolution output of a PWM waveform generated by the GPTW1
GTIOC1B	Output	High resolution output of a PWM waveform generated by the GPTW1
GTIOC2A	Output	High resolution output of a PWM waveform generated by the GPTW2
GTIOC2B	Output	High resolution output of a PWM waveform generated by the GPTW2
GTIOC3A	Output	High resolution output of a PWM waveform generated by the GPTW3
GTIOC3B	Output	High resolution output of a PWM waveform generated by the GPTW3

25.2 Register Descriptions

25.2.1 HRPWM Operation Control Register (HROCR)

Address(es): HRPWM.HROCR 000C 2A00h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	HRRST	DLLEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DLLEN	DLL Operation Enable	0: DLL operation is disabled 1: DLL operation is enabled	R/W
b1	HRRST	High Resolution PWM Waveform Generation Circuit Reset	0: Release from reset 1: Reset	R/W
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register controls the operation of the entire HRPWM module. The register can be written when the register write disable bit in the GPTW write-protection register is enabled (GPTW0.GTWP.WP = 0).

DLLEN Bit (DLL Operation Enable)

This bit selects whether the internal DLL in the HRPWM is activated or not. After activation, wait for 20 µs until the output of the DLL becomes stable.

HRRST Bit (High Resolution PWM Waveform Generation Circuit Reset)

This bit resets the internal state of the HRPWM. After releasing from reset, wait for 12 cycles of PCLKC. When resetting during operation, the PWM output is fixed to low after five cycles of PCLKC.

25.2.2 HRPWM Operation Control Register 2 (HROCR2)

Address(es): HRPWM.HROCR2 000C 2A02h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	HRDIS3	HRDIS2	HRDIS1	HRDIS0	—	—	—	—	HRSEL3	HRSEL2	HRSEL1	HRSEL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	HRSEL0	Channel 0 High Resolution PWM Waveform Output Select	0: Output an original PWM waveform generated by the GPTW0 from the GTIOC0A and GTIOC0B pins 1: Output a high resolution PWM waveform from the GTIOC0A and GTIOC0B pins	R/W
b1	HRSEL1	Channel 1 High Resolution PWM Waveform Output Select	0: Output an original PWM waveform generated by the GPTW1 from the GTIOC1A and GTIOC1B pins 1: Output a high resolution PWM waveform from the GTIOC1A and GTIOC1B pins	R/W
b2	HRSEL2	Channel 2 High Resolution PWM Waveform Output Select	0: Output an original PWM waveform generated by the GPTW2 from the GTIOC2A and GTIOC2B pins 1: Output a high resolution PWM waveform from the GTIOC2A and GTIOC2B pins	R/W
b3	HRSEL3	Channel 3 High Resolution PWM Waveform Output Select	0: Output an original PWM waveform generated by the GPTW3 from the GTIOC3A and GTIOC3B pins 1: Output a high resolution PWM waveform from the GTIOC3A and GTIOC3B pins	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	HRDIS0	Channel 0 Rising and Falling Edge Adjustment Circuit Disable	0: Enable operation of Channel 0 rising and falling edge adjustment circuit 1: Disable operation of Channel 0 rising and falling edge adjustment circuit	R/W
b9	HRDIS1	Channel 1 Rising and Falling Edge Adjustment Circuit Disable	0: Enable operation of Channel 1 rising and falling edge adjustment circuit 1: Disable operation of Channel 1 rising and falling edge adjustment circuit	R/W
b10	HRDIS2	Channel 2 Rising and Falling Edge Adjustment Circuit Disable	0: Enable operation of Channel 2 rising and falling edge adjustment circuit 1: Disable operation of Channel 2 rising and falling edge adjustment circuit	R/W
b11	HRDIS3	Channel 3 Rising and Falling Edge Adjustment Circuit Disable	0: Enable operation of Channel 3 rising and falling edge adjustment circuit 1: Disable operation of Channel 3 rising and falling edge adjustment circuit	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register controls the HRPWM module by channel. The register can be written when the register write disable bit in the GPTW write-protection register is enabled (GPTW0.GTWP.WP = 0).

HRSELn Bit (Channel n High Resolution PWM Waveform Output Select) (n = 0 to 3)

This bit allows to select whether to output a waveform generated by the HRPWM from the GTIOCnA and GTIOCnB pins (n = 0 to 3) or to output an original waveform of the GPTW by bypassing the circuit.
The PWM waveform generated by the HRPWM is output with a delay of three full cycles of PCLKC from the output when the circuit is bypassed, even if the setting for the amount of delay is 0.

HRDISn Bit (Channel n Rising and Falling Edge Adjustment Circuit Disable) (n = 0 to 3)

This bit enables/disables the operation of the rising and falling edge adjustment circuit by relevant channel in the GPTW.

Set this bit to 1 for a channel which will not be used.

25.2.3 GTIOCnA Pin Rising Edge Adjustment Register (HRREARnA) (n = 0 to 3)

Address(es): HRPWM.HRREAR0A 000C 2A18h, HRPWM.HRREAR1A 000C 2A1Ch, HRPWM.HRREAR2A 000C 2A20h, HRPWM.HRREAR3A 000C 2A24h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	DLY[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	DLY[4:0]	Amount of Delay Select	b4 b0 0 0 0 0 0: Apply delay of 0/32 times PCLKC period 0 0 0 0 1: Apply delay of 1/32 times PCLKC period 0 0 0 1 0: Apply delay of 2/32 times PCLKC period 0 0 0 1 1: Apply delay of 3/32 times PCLKC period 0 0 1 0 0: Apply delay of 4/32 times PCLKC period 0 0 1 0 1: Apply delay of 5/32 times PCLKC period 0 0 1 1 0: Apply delay of 6/32 times PCLKC period 0 0 1 1 1: Apply delay of 7/32 times PCLKC period 0 1 0 0 0: Apply delay of 8/32 times PCLKC period 0 1 0 0 1: Apply delay of 9/32 times PCLKC period 0 1 0 1 0: Apply delay of 10/32 times PCLKC period 0 1 0 1 1: Apply delay of 11/32 times PCLKC period 0 1 1 0 0: Apply delay of 12/32 times PCLKC period 0 1 1 0 1: Apply delay of 13/32 times PCLKC period 0 1 1 1 0: Apply delay of 14/32 times PCLKC period 0 1 1 1 1: Apply delay of 15/32 times PCLKC period 1 0 0 0 0: Apply delay of 16/32 times PCLKC period 1 0 0 0 1: Apply delay of 17/32 times PCLKC period 1 0 0 1 0: Apply delay of 18/32 times PCLKC period 1 0 0 1 1: Apply delay of 19/32 times PCLKC period 1 0 1 0 0: Apply delay of 20/32 times PCLKC period 1 0 1 0 1: Apply delay of 21/32 times PCLKC period 1 0 1 1 0: Apply delay of 22/32 times PCLKC period 1 0 1 1 1: Apply delay of 23/32 times PCLKC period 1 1 0 0 0: Apply delay of 24/32 times PCLKC period 1 1 0 0 1: Apply delay of 25/32 times PCLKC period 1 1 0 1 0: Apply delay of 26/32 times PCLKC period 1 1 0 1 1: Apply delay of 27/32 times PCLKC period 1 1 1 0 0: Apply delay of 28/32 times PCLKC period 1 1 1 0 1: Apply delay of 29/32 times PCLKC period 1 1 1 1 0: Apply delay of 30/32 times PCLKC period 1 1 1 1 1: Apply delay of 31/32 times PCLKC period	R/W
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register specifies the amount of delay to apply to rising edge of a PWM waveform output from the GTIOCnA pin. For the transfer timing of register settings, refer to section 25.3.2, Transfer Timing of HRREARnA, HRREARnB, HRFEARnA, and HRFEARnB Register Settings (n = 0 to 3).

This register can be written when the register write disable bit in the GPTW write-protection register is enabled (GPTWn.GTWP.WP = 0).

25.2.4 GTIOCnA Pin Falling Edge Adjustment Register (HRFEARnA) (n = 0 to 3)

Address(es): HRPWM.HRFEAR0A 000C 2A28h, HRPWM.HRFEAR1A 000C 2A2Ch, HRPWM.HRFEAR2A 000C 2A30h, HRPWM.HRFEAR3A 000C 2A34h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	DLY[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	DLY[4:0]	Amount of Delay Select	b4 b0 0 0 0 0 0: Apply delay of 0/32 times PCLKC period 0 0 0 0 1: Apply delay of 1/32 times PCLKC period 0 0 0 1 0: Apply delay of 2/32 times PCLKC period 0 0 0 1 1: Apply delay of 3/32 times PCLKC period 0 0 1 0 0: Apply delay of 4/32 times PCLKC period 0 0 1 0 1: Apply delay of 5/32 times PCLKC period 0 0 1 1 0: Apply delay of 6/32 times PCLKC period 0 0 1 1 1: Apply delay of 7/32 times PCLKC period 0 1 0 0 0: Apply delay of 8/32 times PCLKC period 0 1 0 0 1: Apply delay of 9/32 times PCLKC period 0 1 0 1 0: Apply delay of 10/32 times PCLKC period 0 1 0 1 1: Apply delay of 11/32 times PCLKC period 0 1 1 0 0: Apply delay of 12/32 times PCLKC period 0 1 1 0 1: Apply delay of 13/32 times PCLKC period 0 1 1 1 0: Apply delay of 14/32 times PCLKC period 0 1 1 1 1: Apply delay of 15/32 times PCLKC period 1 0 0 0 0: Apply delay of 16/32 times PCLKC period 1 0 0 0 1: Apply delay of 17/32 times PCLKC period 1 0 0 1 0: Apply delay of 18/32 times PCLKC period 1 0 0 1 1: Apply delay of 19/32 times PCLKC period 1 0 1 0 0: Apply delay of 20/32 times PCLKC period 1 0 1 0 1: Apply delay of 21/32 times PCLKC period 1 0 1 1 0: Apply delay of 22/32 times PCLKC period 1 0 1 1 1: Apply delay of 23/32 times PCLKC period 1 1 0 0 0: Apply delay of 24/32 times PCLKC period 1 1 0 0 1: Apply delay of 25/32 times PCLKC period 1 1 0 1 0: Apply delay of 26/32 times PCLKC period 1 1 0 1 1: Apply delay of 27/32 times PCLKC period 1 1 1 0 0: Apply delay of 28/32 times PCLKC period 1 1 1 0 1: Apply delay of 29/32 times PCLKC period 1 1 1 1 0: Apply delay of 30/32 times PCLKC period 1 1 1 1 1: Apply delay of 31/32 times PCLKC period	R/W
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register specifies the amount of delay to apply to falling edge of a PWM waveform output from the GTIOCnA pin. For the transfer timing of register settings, refer to section 25.3.2, Transfer Timing of HRREARnA, HRREARnB, HRFEARnA, and HRFEARnB Register Settings (n = 0 to 3).

This register can be written when the register write disable bit in the GPTW write-protection register is enabled (GPTWn.GTWP.WP = 0).

25.2.5 GTIOCnB Pin Rising Edge Adjustment Register (HRREARnB) (n = 0 to 3)

Address(es): HRPWM.HRREAR0B 000C 2A1Ah, HRPWM.HRREAR1B 000C 2A1Eh, HRPWM.HRREAR2B 000C 2A22h, HRPWM.HRREAR3B 000C 2A26h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	DLY[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	DLY[4:0]	Amount of Delay Select	b4 b0 0 0 0 0 0: Apply delay of 0/32 times PCLKC period 0 0 0 0 1: Apply delay of 1/32 times PCLKC period 0 0 0 1 0: Apply delay of 2/32 times PCLKC period 0 0 0 1 1: Apply delay of 3/32 times PCLKC period 0 0 1 0 0: Apply delay of 4/32 times PCLKC period 0 0 1 0 1: Apply delay of 5/32 times PCLKC period 0 0 1 1 0: Apply delay of 6/32 times PCLKC period 0 0 1 1 1: Apply delay of 7/32 times PCLKC period 0 1 0 0 0: Apply delay of 8/32 times PCLKC period 0 1 0 0 1: Apply delay of 9/32 times PCLKC period 0 1 0 1 0: Apply delay of 10/32 times PCLKC period 0 1 0 1 1: Apply delay of 11/32 times PCLKC period 0 1 1 0 0: Apply delay of 12/32 times PCLKC period 0 1 1 0 1: Apply delay of 13/32 times PCLKC period 0 1 1 1 0: Apply delay of 14/32 times PCLKC period 0 1 1 1 1: Apply delay of 15/32 times PCLKC period 1 0 0 0 0: Apply delay of 16/32 times PCLKC period 1 0 0 0 1: Apply delay of 17/32 times PCLKC period 1 0 0 1 0: Apply delay of 18/32 times PCLKC period 1 0 0 1 1: Apply delay of 19/32 times PCLKC period 1 0 1 0 0: Apply delay of 20/32 times PCLKC period 1 0 1 0 1: Apply delay of 21/32 times PCLKC period 1 0 1 1 0: Apply delay of 22/32 times PCLKC period 1 0 1 1 1: Apply delay of 23/32 times PCLKC period 1 1 0 0 0: Apply delay of 24/32 times PCLKC period 1 1 0 0 1: Apply delay of 25/32 times PCLKC period 1 1 0 1 0: Apply delay of 26/32 times PCLKC period 1 1 0 1 1: Apply delay of 27/32 times PCLKC period 1 1 1 0 0: Apply delay of 28/32 times PCLKC period 1 1 1 0 1: Apply delay of 29/32 times PCLKC period 1 1 1 1 0: Apply delay of 30/32 times PCLKC period 1 1 1 1 1: Apply delay of 31/32 times PCLKC period	R/W
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register specifies the amount of delay to apply to rising edge of a PWM waveform output from the GTIOCnB pin. For the transfer timing of register settings, refer to section 25.3.2, Transfer Timing of HRREARnA, HRREARnB, HRFEARnA, and HRFEARnB Register Settings (n = 0 to 3).

This register can be written when the register write disable bit in the GPTW write-protection register is enabled (GPTWn.GTWP.WP = 0).

25.2.6 GTIOCnB Pin Falling Edge Adjustment Register (HRFEARnB) (n = 0 to 3)

Address(es): HRPWM.HRFEAR0B 000C 2A2Ah, HRPWM.HRFEAR1B 000C 2A2Eh, HRPWM.HRFEAR2B 000C 2A32h, HRPWM.HRFEAR3B 000C 2A36h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	DLY[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	DLY[4:0]	Amount of Delay Select	b4 b0 0 0 0 0 0: Apply delay of 0/32 times PCLKC period 0 0 0 0 1: Apply delay of 1/32 times PCLKC period 0 0 0 1 0: Apply delay of 2/32 times PCLKC period 0 0 0 1 1: Apply delay of 3/32 times PCLKC period 0 0 1 0 0: Apply delay of 4/32 times PCLKC period 0 0 1 0 1: Apply delay of 5/32 times PCLKC period 0 0 1 1 0: Apply delay of 6/32 times PCLKC period 0 0 1 1 1: Apply delay of 7/32 times PCLKC period 0 1 0 0 0: Apply delay of 8/32 times PCLKC period 0 1 0 0 1: Apply delay of 9/32 times PCLKC period 0 1 0 1 0: Apply delay of 10/32 times PCLKC period 0 1 0 1 1: Apply delay of 11/32 times PCLKC period 0 1 1 0 0: Apply delay of 12/32 times PCLKC period 0 1 1 0 1: Apply delay of 13/32 times PCLKC period 0 1 1 1 0: Apply delay of 14/32 times PCLKC period 0 1 1 1 1: Apply delay of 15/32 times PCLKC period 1 0 0 0 0: Apply delay of 16/32 times PCLKC period 1 0 0 0 1: Apply delay of 17/32 times PCLKC period 1 0 0 1 0: Apply delay of 18/32 times PCLKC period 1 0 0 1 1: Apply delay of 19/32 times PCLKC period 1 0 1 0 0: Apply delay of 20/32 times PCLKC period 1 0 1 0 1: Apply delay of 21/32 times PCLKC period 1 0 1 1 0: Apply delay of 22/32 times PCLKC period 1 0 1 1 1: Apply delay of 23/32 times PCLKC period 1 1 0 0 0: Apply delay of 24/32 times PCLKC period 1 1 0 0 1: Apply delay of 25/32 times PCLKC period 1 1 0 1 0: Apply delay of 26/32 times PCLKC period 1 1 0 1 1: Apply delay of 27/32 times PCLKC period 1 1 1 0 0: Apply delay of 28/32 times PCLKC period 1 1 1 0 1: Apply delay of 29/32 times PCLKC period 1 1 1 1 0: Apply delay of 30/32 times PCLKC period 1 1 1 1 1: Apply delay of 31/32 times PCLKC period	R/W
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register specifies the amount of delay to apply to falling edge of a PWM waveform output from the GTIOCnB pin. For the transfer timing of register settings, refer to section 25.3.2, Transfer Timing of HRREARnA, HRREARnB, HRFEARnA, and HRFEARnB Register Settings (n = 0 to 3).

This register can be written when the register write disable bit in the GPTW write-protection register is enabled (GPTWn.GTWP.WP = 0).

25.3 Operation

25.3.1 Adjustment to the Timing of Rising and Falling Edges in PWM Waveforms

The HRPWM delays timing for rising and falling of the PWM signals which are output from the GPTW with a resolution of 1/32 times (at least around 260 ps) of a PCLKC period (tC(PCLKC)) based on the delay locked loop (DLL) operating with PCLKC as a reference clock. The PWM signal which was delayed is output from the GTIOCnA and GTIOCnB pins (n = 0 to 3).

The amount of delay can be adjusted within a range of 0/32 to 31/32 of tC(PCLKC). The amount of delay out of this range cannot be specified. When setting the amount of delay with the value out of such range, change the setting in the general PWM timer compare capture register m (GTCCRm) in the GPTW.

The HRPWM is required to set the HROCR.DLLEN bit to 1 before operation and wait for stabilization time as it has the DLL. In addition, wait for 12 cycles of PCLKC after setting the HROCR.HRRST bit to 0 in order to output the reliable PWM waveform. Refer to Figure 25.2 for the detailed procedures.

Specify the amount of delay by the HRREARnA and HRFEARnA registers for the GTIOCnA pin and by the HRREARnB and HRFEARnB registers for the GTIOCnB pin. The PWM waveform has specific setting registers for delay at rising and at falling respectively, which allows fine adjustments for the width of high and low. Refer to Table 25.3 for the correspondence between pins and registers.

The amount of delay specified by a relevant register is reflected to the pin via a temporary register. Timing for reflecting the setting differs by PWM output operation mode in the general PWM timer. Refer to section 25.3.2, Transfer Timing of HRREARnA, HRREARnB, HRFEARnA, and HRFEARnB Register Settings (n = 0 to 3).

In the HRPWM, setting of DLL enabling/disabling and reset setting/releasing is common for all channels; however, the rest of settings can be specified individually by channel.

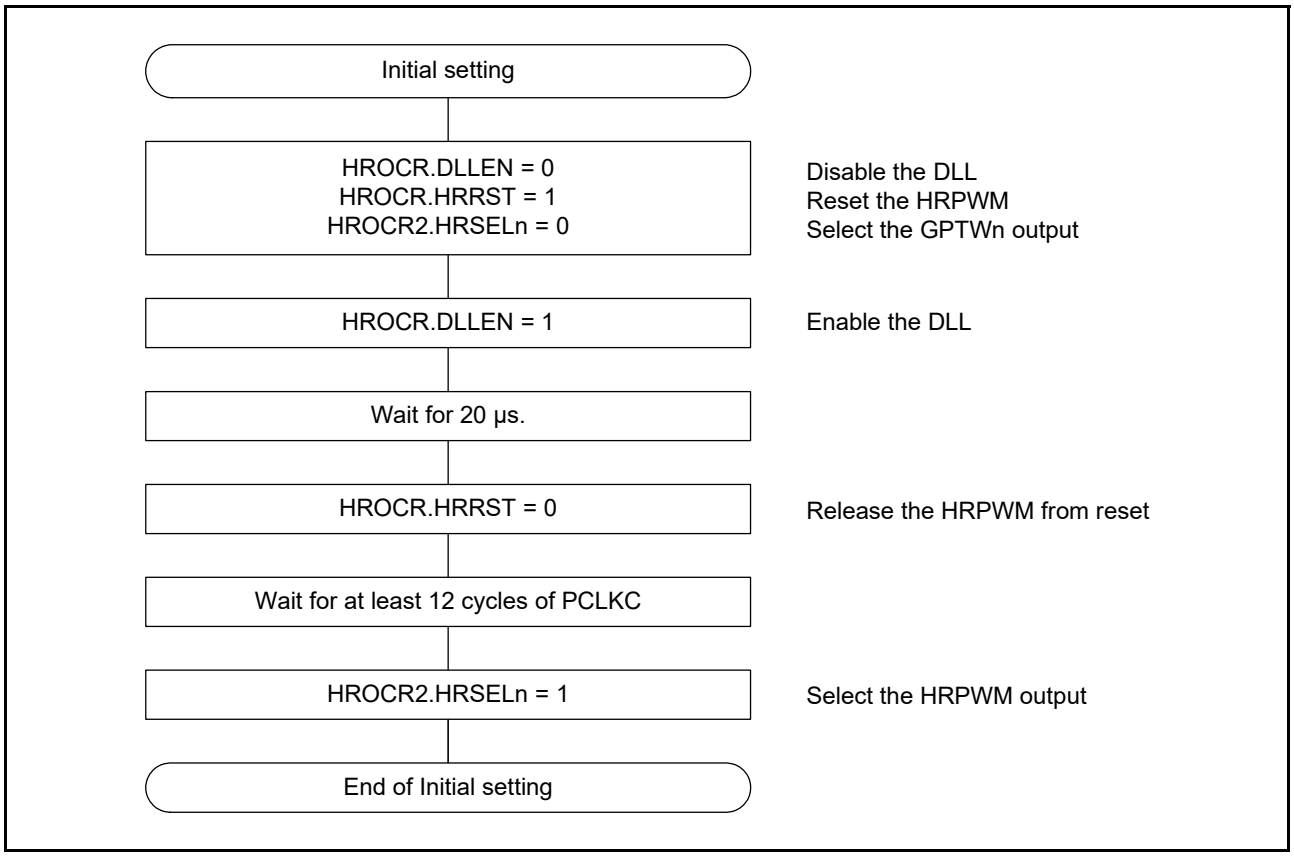


Figure 25.2 Initialization flow for the HRPWM (n = 0 to 3)

Table 25.3 Correspondence between PWM Waveform Output Pins and Delay Setting Registers

PWM Waveform Output Pin	Rising Edge Adjustment Register	Falling Edge Adjustment Register
GTIOC0A	HRREAR0A	HRFEAR0A
GTIOC0B	HRREAR0B	HRFEAR0B
GTIOC1A	HRREAR1A	HRFEAR1A
GTIOC1B	HRREAR1B	HRFEAR1B
GTIOC2A	HRREAR2A	HRFEAR2A
GTIOC2B	HRREAR2B	HRFEAR2B
GTIOC3A	HRREAR3A	HRFEAR3A
GTIOC3B	HRREAR3B	HRFEAR3B

25.3.2 Transfer Timing of HRREARnA, HRREARnB, HRFEARnA, and HRFEARnB Register Settings (n = 0 to 3)

Settings of the HRREARnA, HRREARnB, HRFEARnA, and HRFEARnB registers are transferred to a temporary register first, and reflected to the amount of delay for the PWM waveform in the GTIOCnA and GTIOCnB pins then. Such settings are transferred to a temporary register at the end of PWM period (at an overflow (during up-count operation) or an underflow (during down-count operation) in sawtooth wave mode, and at a trough in triangle wave mode).

Figure 25.3 and Figure 25.4 show examples of the operation of the HRREAR0A and HRFEAR0A registers.

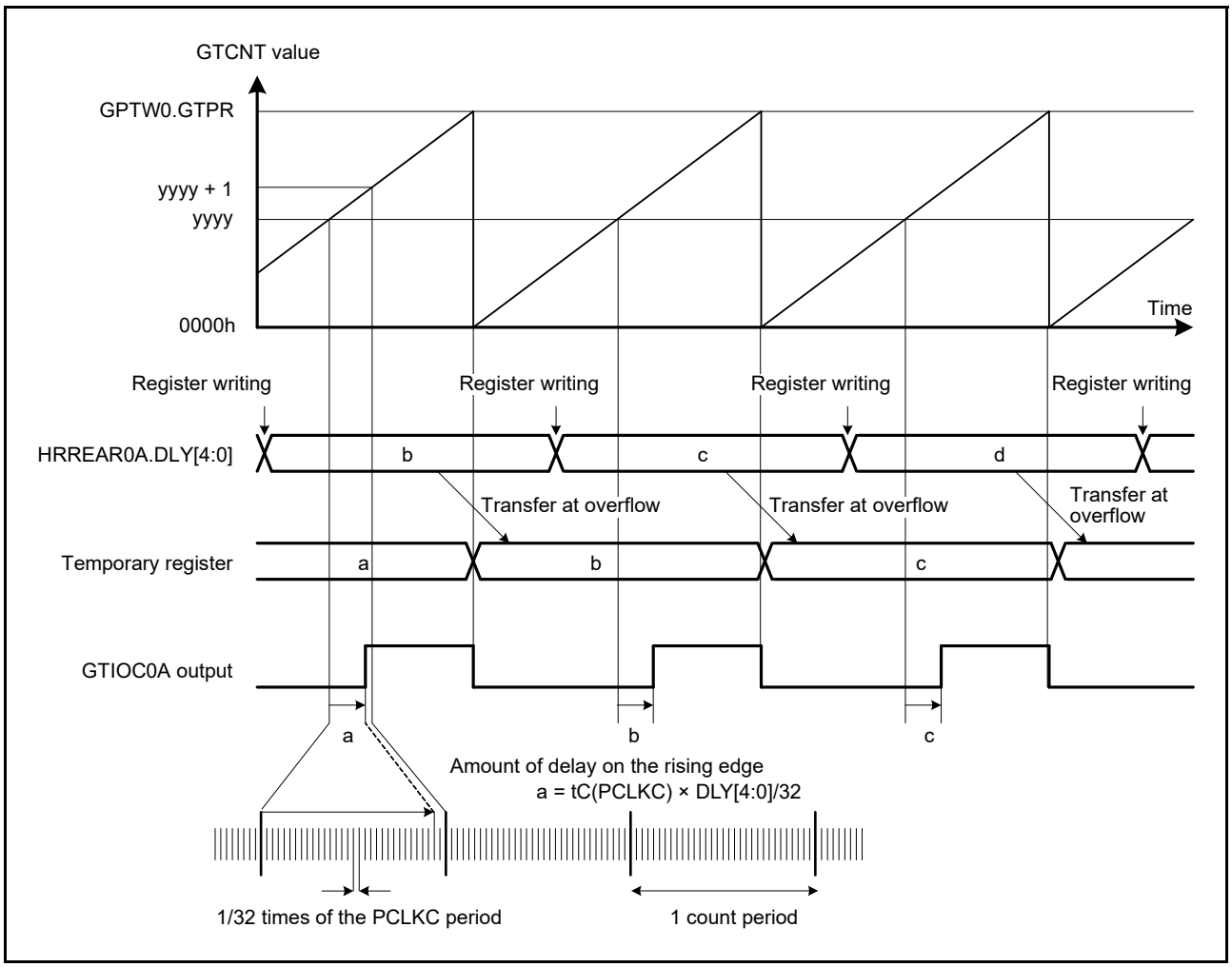


Figure 25.3 Operation Example When Changing the HRREAR0A Register Setting (in Sawtooth Wave Mode)

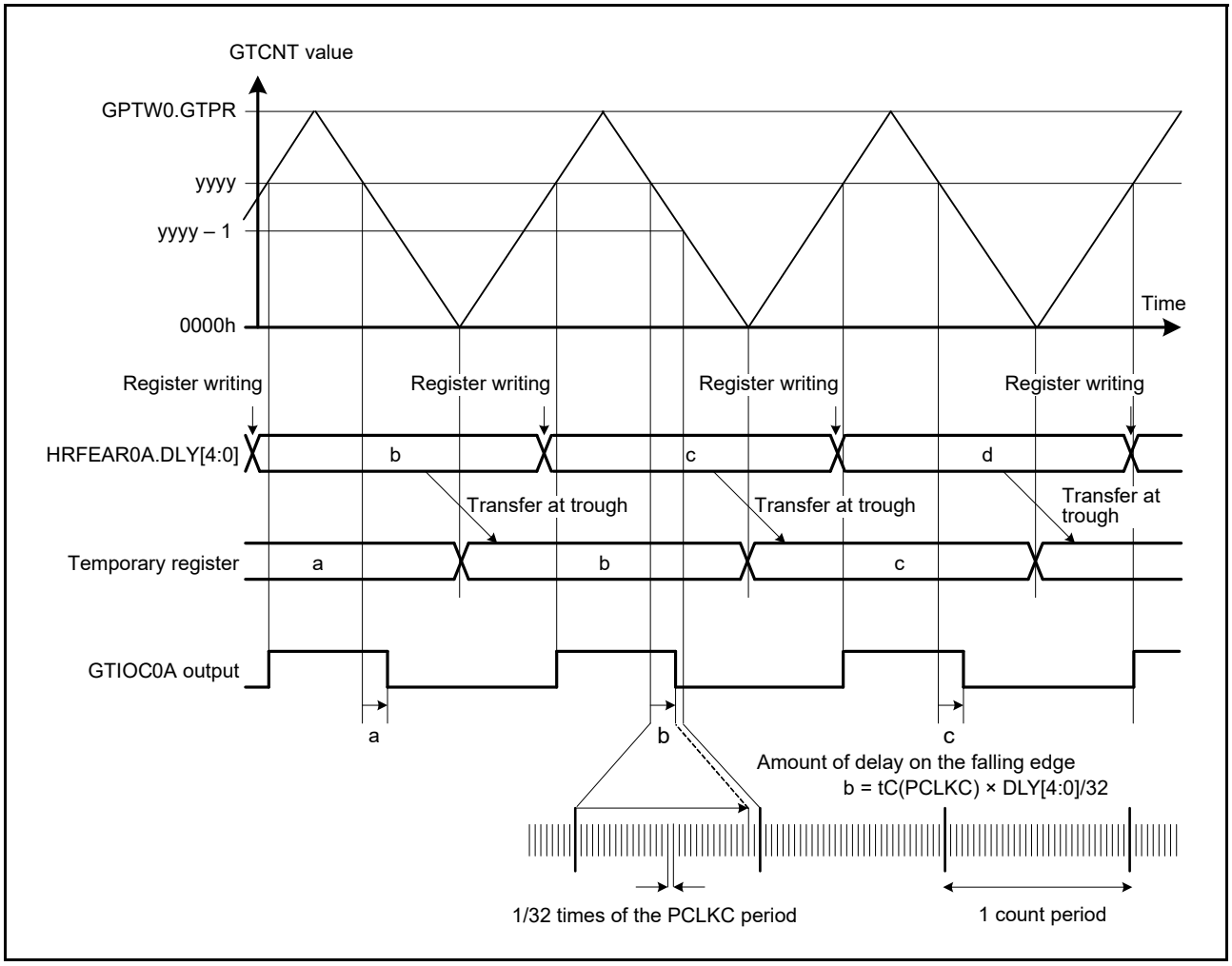


Figure 25.4 Operation Example When Changing the HRFEAR0A Register Setting (in Triangle Wave Mode)

25.4 Usage Notes

25.4.1 Settings for the Module-Stop Function

Operation of the HRPWM can be disabled or enabled using Module Stop Control Register (MSTPCRA). The HRPWM is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, refer to section 11, Low Power Consumption.

25.4.2 Note on Selecting Timer Prescaler in the GTCNT Counter While Using the HRPWM

The PWM signals output from the GPTW are generated based on the GTCNT counter operating at resolution equivalent to one clock period which is set by the GPTWn.GTCR.TPCS[3:0] bits (n = 0 to 3). Note that a delay at resolution of 0/32 × tC(PCLKC) to 31/32 × tC(PCLKC) is applied to edges of the PWM signals at low resolution generated based on a selected clock when a setting for the GPTWn.GTCCRm register (m = A to F) is changed while the TPCS[3:0] bits are specified to the setting other than 0000b (PCLKC/1).

25.4.3 Notes on Delay Settings for HRPWM

Within three cycles of PCLKC from the end of the PWM period, do not change the settings for the HRFEARnA, HRREARnA, HRFEARnB, and HRREARnB registers. Any changes made during the above period will be reflected with a delay of one cycle of the PWM waveform.

26. GPTW Port Output Enable (POEG)

26.1 Overview

The POEG issues requests to stop output from output pins of the general PWM timer (GPTW). The combination of output pins of the POEG to be stopped can be specified from any channel of the GPTW, which is a different form of operation from that of the POE3. Use the POE3 to stop output from pins of the MTU3 and GPTW at the same time. Select the method of detection for stopping the output from the list below.

- Detection of active level or edge input on the GTETRGn pin (n = A to D)
- Detection from the GPTW to stop output
- Detection by comparator (edge detection or level detection)
- Detection of stopping of oscillation by the oscillation stop detection circuit for the main clock
- Register setting

The GTETRGn pin can be used for output to the external trigger input pins of the GPTW.
Table 26.1 shows specifications, Figure 26.1 shows a schematic view of the system, Figure 26.2 shows a block diagram, and Table 26.2 shows input pins.

Table 26.1 POEG Specifications

Item	Description
Request of output stopping in response to detection of active level or edge input	• The request to stop output is issued to the GPTW when a POEGGn.PIDF flag is set in response to the detection of either input of the selected level or the selected edge on the corresponding GTETRGn pin (n = A to D). • The request to stop output is issued to the GPTW immediately upon detection of input of the selected level on the corresponding GTETRGn pin.
Requests to stop output in the form of an output stopping signal from the GPTW	• The request to stop output is issued to the GPTW when the GPTW detects the active level (high or low) on the GTIOCA and GTIOCB pins at the same time while the corresponding POEGGn.IOCF flag is set. • The request to stop output is issued to the GPTW when the GPTW detects a dead-time error while the corresponding POEGGn.IOCF flag is set.
Requests to stop output in response to detection by a comparator	• The request to stop output is issued to the GPTW when a POEGGn.IOCF flag is set in response to edge-detection by a comparator. • The request to stop output is issued directly to the GPTW upon detecting level on a comparator. • The PWM output level on the GPTW output pin selected by the POEGICRn.MSEL[4:0] bits enables or disables requests to stop output.
Requests to stop output in response to detecting the stopping of oscillation	A request to stop output is issued to the GPTW when the oscillation stop detection circuit for the main clock detects the stopping of oscillation while the corresponding POEGGn.OSTPF flag is set.
Requests by software to stop output	The request to stop output is issued to the GPTW when the software sets the POEGGn.SSF flag.
Interrupt	• An interrupt is generated in response to the request to stop output in the form of the POEGGn.PIDF flag. • An interrupt is generated in response to the request to stop output in the form of the POEGGn.IOCF flag.
External trigger output to the GPTW	The GTETRGn pin is used for output as external triggers.
Noise removal	• Each GTETRGn pin has a digital noise filter. • Eight types of sampling clock are selectable for the filter. • The number of samples is selectable as a value from 3 to 6.

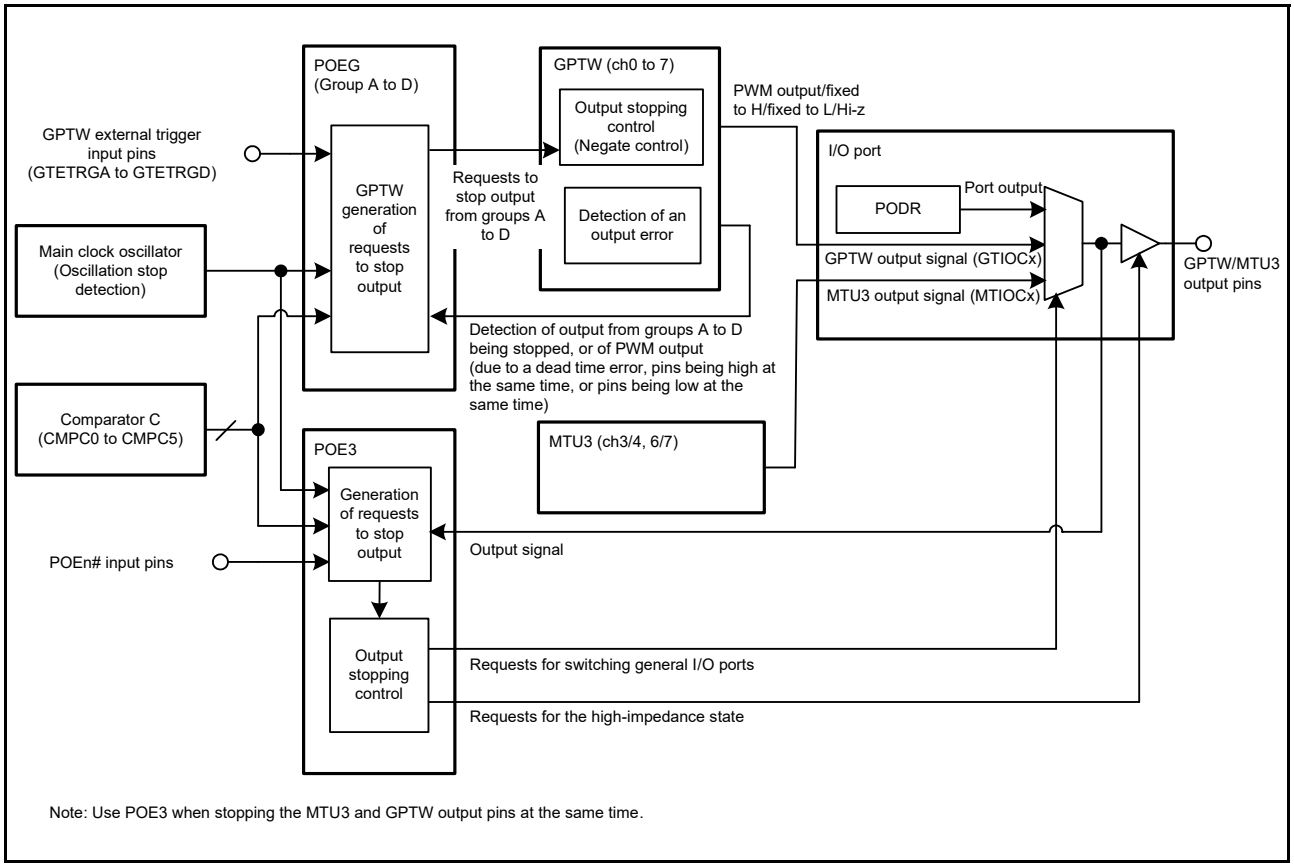


Figure 26.1 Schematic View of the POEG System

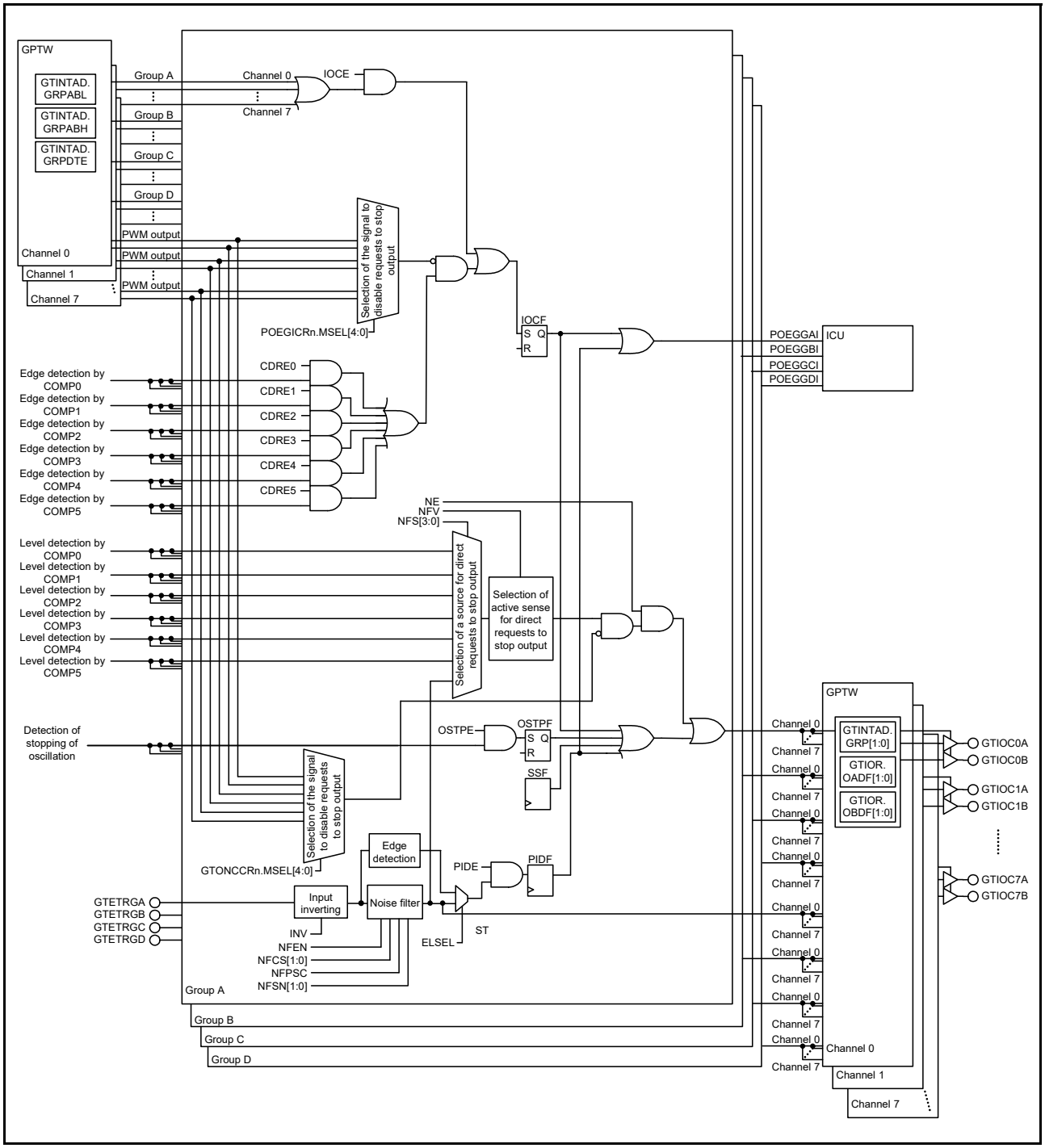


Figure 26.2 POEG Block Diagram

Table 26.2 POEG I/O Pins

Pin Name	I/O	Function
GTETRGA	Input	Output stop request signal for GPTW output pin and GPTW external trigger input pin A
GTETRGB	Input	Output stop request signal for GPTW output pin and GPTW external trigger input pin B
GTETRGC	Input	Output stop request signal for GPTW output pin and GPTW external trigger input pin C
GTETRGD	Input	Output stop request signal for GPTW output pin and GPTW external trigger input pin D

26.2 Register Descriptions

26.2.1 POEG Group n Setting Register (POEGGn) (n = A to D)

Address(es): POEG.POEGGA 0009 E000h, POEG.POEGGB 0009 E100h, POEG.POEGGC 0009 E200h, POEG.POEGGD 0009 E300h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	NFCS[1:0]		NFEN	INV	NFSN[1:0]		ELSEL	NFPSC	—	—	—	—	—	—	—	ST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CDRE5	CDRE4	CDRE3	CDRE2	CDRE1	CDRE0	—	OSTPE	IOCE	PIDE	SSF	OSTPF	IOCF	PIDF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PIDF	Port Input Detection Flag	0: The selected input level was not detected on the GTETRGn pin. 1: The selected input level was detected on the GTETRGn pin.	R/(W) *1
b1	IOCF	GPTW or CMPC Output Stop Request Detection Flag	0: Neither stopping of GPTW output nor a comparator edge was detected. 1: Either stopping of GPTW output or comparator edge was detected.	R/(W) *1
b2	OSTPF	Oscillation Stop Detection Flag	0: Stopping of oscillation was not detected. 1: Stopping of oscillation was detected.	R/(W) *1
b3	SSF	Software Stop Flag	0: Software has not stopped output. 1: Software has stopped output.	R/W
b4	PIDE	Port Input Detection Enable	0: Detection of input levels on the corresponding GTETRGn pin is disabled. 1: Detection of input levels on the corresponding GTETRGn pin is enabled.	R/W*2
b5	IOCE	GPTW Output Stop Request Enable	0: Detection of stopping of output from the GPTW is disabled. 1: Detection of stopping of output from the GPTW is enabled.	R/W*2
b6	OSTPE	Enable Stopping Output on Stopping of Oscillation	0: Detection of stopping of oscillation is disabled. 1: Detection of stopping of oscillation is enabled.	R/W*2
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b8	CDRE0	CMPC_0 Enable	0: Comparator edge detection 0 is disabled. 1: Comparator edge detection 0 is enabled.	R/W*2
b9	CDRE1	CMPC_1 Enable	0: Comparator edge detection 1 is disabled. 1: Comparator edge detection 1 is enabled.	R/W*2
b10	CDRE2	CMPC_2 Enable	0: Comparator edge detection 2 is disabled. 1: Comparator edge detection 2 is enabled.	R/W*2
b11	CDRE3	CMPC_3 Enable	0: Comparator edge detection 3 is disabled. 1: Comparator edge detection 3 is enabled.	R/W*2
b12	CDRE4	CMPC_4 Enable	0: Comparator edge detection 4 is disabled. 1: Comparator edge detection 4 is enabled.	R/W*2
b13	CDRE5	CMPC_5 Enable	0: Comparator edge detection 5 is disabled. 1: Comparator edge detection 5 is enabled.	R/W*2
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	ST	GTETRGn Input Status Flag	0: The corresponding external trigger for output to the GPTW is 0. 1: The corresponding external trigger for output to the GPTW is 1.	R

Bit	Symbol	Bit Name	Description	R/W
b23 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b24	NFPSC	Noise Filter Clock Select	These bits are used to set the clock division ratio in combination with the NFCS[1:0] bits.	R/W
b25	ELSEL	GTETRGn Input Edge or Level Detection Select	0: Level detection 1: Edge detection	R/W
b27, b26	NFSN[1:0]	Noise Filter Sampling Count Select	b27 b26 0 0: 3 times 0 1: 4 times 1 0: 5 times 1 1: 6 times	R/W
b28	INV	GTETRGn Input Inverting	0: Input on the GTETRGn pin is not inverted. 1: Input on the GTETRGn pin is inverted.	R/W
b29	NFEN	Noise Filter Enable	0: Digital noise filter on the GTETRGn pin is disabled. 1: Digital noise filter on the GTETRGn pin is enabled.	R/W
b31, b30	NFCS[1:0]	Noise Filter Clock Select	<When the setting of the NFPSC bit is 0> b31 b30 0 0: The input level on the GTETRGn pin is sampled per PCLK_GPTn/1 clock interval. 0 1: The input level on the GTETRGn pin is sampled per PCLK_GPTn/8 clock interval. 1 0: The input level on the GTETRGn pin is sampled per PCLK_GPTn/32 clock interval. 1 1: The input level on the GTETRGn pin is sampled per PCLK_GPTn/128 clock interval. <When the setting of the NFPSC bit is 1> b31 b30 0 0: The input level on the GTETRGn pin is sampled per PCLK_GPTn/2 clock interval. 0 1: The input level on the GTETRGn pin is sampled per PCLK_GPTn/4 clock interval. 1 0: The input level on the GTETRGn pin is sampled per PCLK_GPTn/16 clock interval. 1 1: The input level on the GTETRGn pin is sampled per PCLK_GPTn/64 clock interval.	R/W

Note 1. Only 0 to clear the flag can be written.
Note 2. This bit can be written only once after resetting.

The POEGGn register (n = A to D) controls requests for stopping output and an external trigger for the GPTW based in response to the detection of various signals.

SSF Flag (Software Stop Flag)

Writing 1 to the SSF flag leads to a request to stop output being issued to the GPTW, and writing 0 to the flag releases the GPTW from the request to stop output. In addition, requests to stop output issued by software can be monitored by reading the flag.

26.2.2 POEG Group n Input Control Register (POEGICRn) (n = A to D)

Address(es): POEG.POEGICRA 0009 E004h, POEG.POEGICRB 0009 E104h, POEG.POEGICRC 0009 E204h, POEG.POEGICRD 0009 E304h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	MSEL[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	MSEL[4:0]	Output Stop Request Disabling Signal Select	The detection of requests to stop output in response to the CMPC output selected by the POEGGn.CDRE[5:0] bits is disabled when the selected signal is at the high level. b4 b0 0 0 0 0 0: Requests to stop output are not disabled. 0 0 0 0 1: The GTIOC0A signal controls requests to stop output. 0 0 0 1 0: The GTIOC0B signal controls requests to stop output. 0 0 0 1 1: The GTIOC1A signal controls requests to stop output. 0 0 1 0 0: The GTIOC1B signal controls requests to stop output. 0 0 1 0 1: The GTIOC2A signal controls requests to stop output. 0 0 1 1 0: The GTIOC2B signal controls requests to stop output. 0 0 1 1 1: The GTIOC3A signal controls requests to stop output. 0 1 0 0 0: The GTIOC3B signal controls requests to stop output. 0 1 0 0 1: The GTIOC4A signal controls requests to stop output. 0 1 0 1 0: The GTIOC4B signal controls requests to stop output. 0 1 0 1 1: The GTIOC5A signal controls requests to stop output. 0 1 1 0 0: The GTIOC5B signal controls requests to stop output. 0 1 1 0 1: The GTIOC6A signal controls requests to stop output. 0 1 1 1 0: The GTIOC6B signal controls requests to stop output. 0 1 1 1 1: The GTIOC7A signal controls requests to stop output. 1 0 0 0 0: The GTIOC7B signal controls requests to stop output. Other settings are prohibited.	R/W *1
b31 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. This bit can be written only once after resetting.

26.2.3 GPTW Output Stopping Control Group n Write Protection Register (GTONCWPn) (n = A to D)

Address(es): POEG.GTONCWPA 0009 E040h, POEG.GTONCWPB 0009 E140h, POEG.GTONCWPC 0009 E240h, POEG.GTONCWPD 0009 E340h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PRKEY[7:0]								—	—	—	—	—	—	—	WP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	WP	Register Writing Disable	0: Writing to the GTONCCRn register is enabled. 1: Writing to the GTONCCRn register is disabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	PRKEY[7:0]	Key Code	These bits control whether or not writing new values to the GTONCWPn register is possible. To write to the GTONCWPn register, write A5h to the 8 higher-order bits and write any value to make up a 16-bit unit to the 8 lower-order bits at the same time. These bits are read as 00h.	R/W

The GTONCWPn register (n = A to D) enables or disables writing to the GTONCCRn register in order to avoid incorrect writing to registers.

26.2.4 GPTW Output Stopping Control Group n Controlling Register (GTONCCRn) (n = A to D)

Address(es): POEG.GTONCCRA 0009 E044h, POEG.GTONCCRB 0009 E144h, POEG.GTONCCRC 0009 E244h, POEG.GTONCCRD 0009 E344h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MSEL[4:0]					—	—	NFV	NFS[3:0]				—	—	—	NE
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	NE	Direct Stopping Request Setting	0: The signal for detection is not set as a direct stopping request signal. 1: The signal for detection is set as a direct stopping request signal.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7 to b4	NFS[3:0]	Direct Stopping Request Select	b7 b4 0 0 0 0: Comparator level detection 0 0 0 0 1: Comparator level detection 1 0 0 1 0: Comparator level detection 2 0 1 0 0: Comparator level detection 3 0 1 0 1: Comparator level detection 4 0 1 1 0: Comparator level detection 5 0 1 1 1: GTETRGn pin input level detection (n = A to D) Other settings are prohibited.	R/W
b8	NFV	Direct Stopping Request Active Sense	0: Stopping output is requested when the output stopping detection signal is 0. 1: Stopping output is requested when the output stopping detection signal is 1.	R/W
b10 to b9	—	Reserve	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit Name	Description	R/W
b15 to b11	MSEL[4:0]	Negate Control Disabling Signal Select	Control for negation is disabled when the selected signal is at the high level. b15 b11 0 0 0 0 0: Control for negation is not disabled. 0 0 0 0 1: The GTIOC0A signal controls the enabling and disabling of negation. 0 0 0 1 0: The GTIOC0B signal controls the enabling and disabling of negation. 0 0 0 1 1: The GTIOC1A signal controls the enabling and disabling of negation. 0 0 1 0 0: The GTIOC1B signal controls the enabling and disabling of negation. 0 0 1 0 1: The GTIOC2A signal controls the enabling and disabling of negation. 0 0 1 1 0: The GTIOC2B signal controls the enabling and disabling of negation. 0 0 1 1 1: The GTIOC3A signal controls the enabling and disabling of negation. 0 1 0 0 0: The GTIOC3B signal controls the enabling and disabling of negation. 0 1 0 0 1: The GTIOC4A signal controls the enabling and disabling of negation. 0 1 0 1 0: The GTIOC4B signal controls the enabling and disabling of negation. 0 1 0 1 1: The GTIOC5A signal controls the enabling and disabling of negation. 0 1 1 0 0: The GTIOC5B signal controls the enabling and disabling of negation. 0 1 1 0 1: The GTIOC6A signal controls the enabling and disabling of negation. 0 1 1 1 0: The GTIOC6B signal controls the enabling and disabling of negation. 0 1 1 1 1: The GTIOC7A signal controls the enabling and disabling of negation. 1 0 0 0 0: The GTIOC7B signal controls the enabling and disabling of negation. Other settings are prohibited.	R/W

The GTONCCRn register (n = A to D) set up requests for stopping output in response to detected signals.

NE Bit (Direct Stopping Request Setting)

Writing 1 to the NE bit leads to direct output of the stopping request signal in response to the signal for detection selected by the NFS[3:0] bits.

26.3 Operation

26.3.1 Request to Stop Output in Response to Detection of Active Level or Edge Input on the Corresponding GTETRGn Pin (n = A to D)

There are two types of output stopping request: requests in response to setting of the POEGGn.PIDF flag (n = A to D), and requests directly in response to the detected signal.

(1) To request the stopping of output in response to setting of the PIDF flag, the input level or edge in accord with the settings of the following bits of the POEGGn register—inversion or non-inversion in the INV bit, detection of a level or edge in the ELSEL bit, use or non-use of the clock division ratio in the NFPSC bit, number of samples for filtering in the NFSN[1:0] bits, enabling or disabling of filtering by the NFEN bit, and the sample clock for filtering in the NFCS[1:0] bits—is detected while the POEGGn.PIDE bit is 1. After the POEGGn.PIDF flag is set to 1 in response to this, a request to stop output is issued per group to each channel of the GPTW. To de-assert the request signal for stopping output, clear the POEGGn.PIDF flag. For canceling requests to stop output, refer to section 26.3.6, Canceling Requests to Stop Output.

(2) To request the stopping of output in direct response to detection of the active level input of a selected signal, the input level in accord with the settings of the following bits of registers—inversion or non-inversion in the INV bit, use or non-use of the clock division ratio in the NFPSC bit, number of samples for filtering in the NFSN[1:0] bits, enabling or disabling of filtering by the NFEN bit, and the sample clock for filtering in the NFCS[1:0] bits of the POEGGn register, and the active sense in the NFV bit of the GTONCCRn register—on the GTETRGn pin selected in the GTONCCRn.NFS[3:0] bits is detected while the GTONCCRn.NE bit is 1, and a request to stop output is then directly issued per group to each channel of the GPTW. The signal to request stopping of output is de-asserted when the detected input level does not match the conditions for issuing a request. For details, refer to section 26.3.7, Requests to Stop Output in Response to Detected Signals and Canceling the Requests.

26.3.1.1 Digital Noise Filter

Each GTETRGn pin input has a digital noise filter. Figure 26.3 shows the operation example in detection of the high level with the use of the filter. When the digital noise filter is enabled, that is, the setting of POEGGn.NFEN bit is 1, and the high level is detected consecutively for the number of samples selected by the POEGGn.NFSN[1:0] bits with the sampling clock period selected by the combination of the POEGGn.NFPSC and POEGGn.NFCS[1:0] bits with inversion or non-inversion in accord with the setting of the POEGGn.INV bit, this is regarded as detection of the high level, and a request to stop output is issued to the GPTW.

At this time, if the low level is detected even once in the sequence, it is not regarded as detection of the high level. In addition, changes in the levels on pins GTETRGA through GTETRGD are ignored while the sampling clock is not being output.

Digital noise filters can be used with requests to stop output in response to setting of the POEGGn.PIDF flags (n = A to D), requests to stop output in direct response to a detected signal, and the output of external triggers to the GPTW.

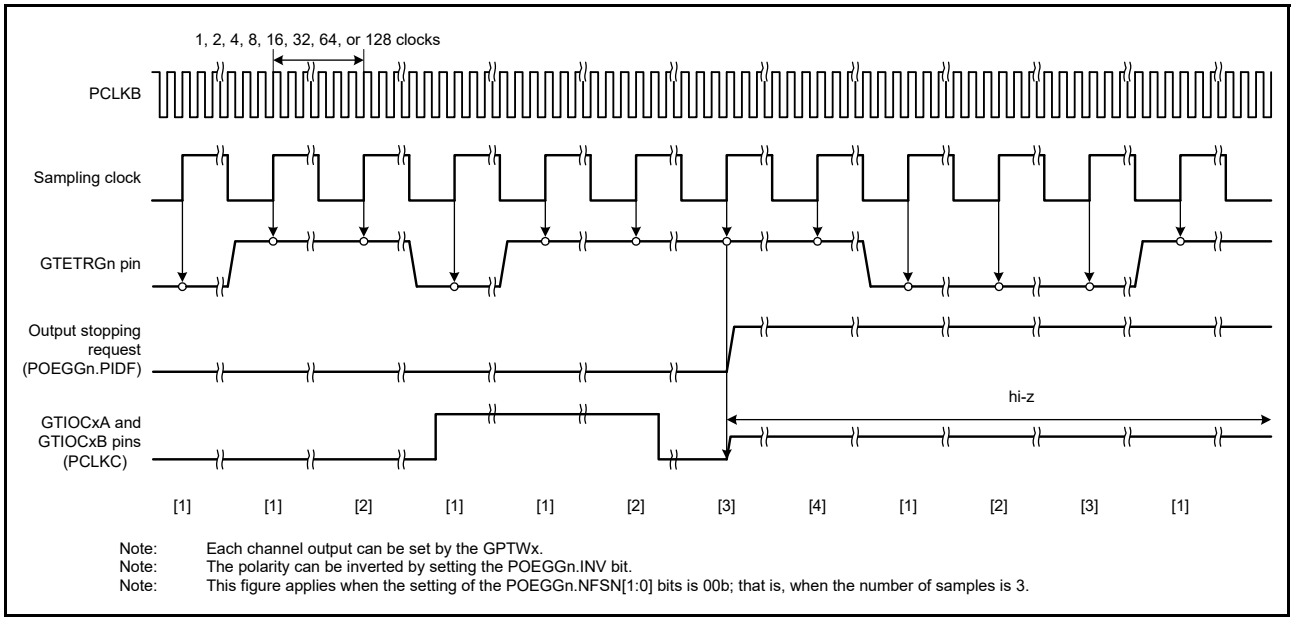


Figure 26.3 Example for Operation of Digital Noise Filter

26.3.2 Requests to Stop Output in Response to Detection of Output Stopping form GPTW

When any among a dead-time error or simultaneous high- or low-level output from the GPTW is detected, the corresponding POEGGn.IOCF flag is set to 1 and the request to stop output per group is issued to each channel in the GPTW. The POEGGn.IOCF flag is used to indicate both edge detection by the comparator and requests to stop output. To cancel a request to stop output, clear the given POEGGn.IOCF flag. For details, refer to section 26.3.6, Canceling Requests to Stop Output.

To detect any among a dead-time error or simultaneous high- or low-level output from the GPTW, detection of output stopping must be permitted in the GRPDTE, GRPABH, and GRPABL bits of the given GPTWn.GTINTAD register. Specify the group of the GPTW for which stopping is to be detected in the GPTWn.GTINTAD.GRP[1:0] bits. For details, refer to section 24.2.15, General PWM Timer Interrupt Output Setting Register (GTINTAD).

26.3.3 Request to Stop Output in Response to Comparator Detection

A request to stop output can be issued to the GPTW in response to detection by a comparator. There are two types of request to stop output: requests output in response to setting of a POEGGn.IOCF flag (n = A to D) due to edge detection by the comparator and requests that are directly output in response to level detection by the comparator. Requests to stop output due to comparator interrupt requests can be controlled in accord with the PWM output level on the GPTW output pin selected by the POEGICRn.MSEL[4:0] bits. The requests are disabled while the output on the given pin is at the high level.

(1) To stop output in response to setting of an IOCF flag, the corresponding edge is detected by the comparator while the POEGGn.CDRE[5:0] bits are 1. The POEGGn.IOCF flag is then set to 1, which leads to a request to stop output from the given group for each channel of the GPTW. The POEGGn.IOCF flag is used for both comparator edge detection and the request to stop output. To cancel the request to stop output, clear the POEGGn.IOCF flag. For details, refer to section 26.3.6, Canceling Requests to Stop Output.

(2) The output can also be stopped by using a detected signal as a direct request to stop output. When the comparator level detection signal selected in the GTONCCRn.NFS[3:0] bits matches the level set in the GTONCCRn.NFV bit, the request to stop output per group is issued to each channel of the GPTW. The request to stop output is released when the comparator level detection does not match the issuing conditions. For details, refer to section 26.3.7, Requests to Stop Output in Response to Detected Signals and Canceling the Requests.

26.3.4 Requests to Stop Output by Oscillation Stop Detection

When the circuit for detecting stopping of oscillation by the main clock oscillator detects the stopping of the oscillation while a POEGGn.OSTPE bit is 1, the POEGGn.OSTPF flag is set to 1 and a request to stop output per group is issued to each channel of the GPTW. To cancel the request to stop output, clear the POEGGn.OSTPF flag. For details, refer to section 26.3.6, Canceling Requests to Stop Output.

26.3.5 Requests to Stop Output by a Register

Writing 1 to the software stop flag (POEGGn.SSF) leads to a request to stop output per group to each channel in the GPTW. To release from the request to stop output, clear the POEGGn.SSF flag. For details, refer to section 26.3.6, Canceling Requests to Stop Output.

26.3.6 Canceling Requests to Stop Output

Requests to stop output are canceled in any of the following three ways.

(1) Cancellation by a reset (return to the initial state)

(2) Cancellation by clearing all flags in the POEGGn register

(3) Cancellation in response to direct input of a detected signal

(1) Cancellation by a reset

Any type of reset can cancel a request to stop output. For details, refer to section 6, Resets.

(2) Cancellation by clearing all flags in the POEGGn register

- POEGGn.PIDF
- POEGGn.IOCF
- POEGGn.OSTPF
- POEGGn.SSF

Cancellation of the request is taken into the GPTW at the end of the cycle of counting by the GPTW, and the output pins are released from being stopped no less than 3 PCLKC cycles from that time. Figure 26.4 shows the timing of release from the output-stopped state. To clear each of the flags, read the status flag for each source to check that the source condition is not being detected, and then write 0. Sources other than the edge detection by the comparators cannot be cleared even if the flag is cleared in the detected state. Since the comparators detect edges in edge detection, writing 0 in the detected state clears the flag, and the flag is not set until the relevant source generates the next edge. Status flags for the respective sources are listed below.

- Detection of input level — POEGGn.ST (GTETRGn input status flag)
- Comparator edge detection — CMPCn.CMPMON.CMPMON0 (comparator output monitoring flag)
- Oscillation stop detection — OSTDSR.OSTDF (oscillation stop detection flag)
- Stopping detection from the GPTW — GPTWn.GTST.DTEF (dead time error flag)
  GPTWn.GTST.OABLF (simultaneous low output flag)
  GPTWn.GTST.OABHF (simultaneous high output flag)

Note 1. CMPCn.CMPMON.CMPMON0 are the monitoring flags for the comparator outputs.

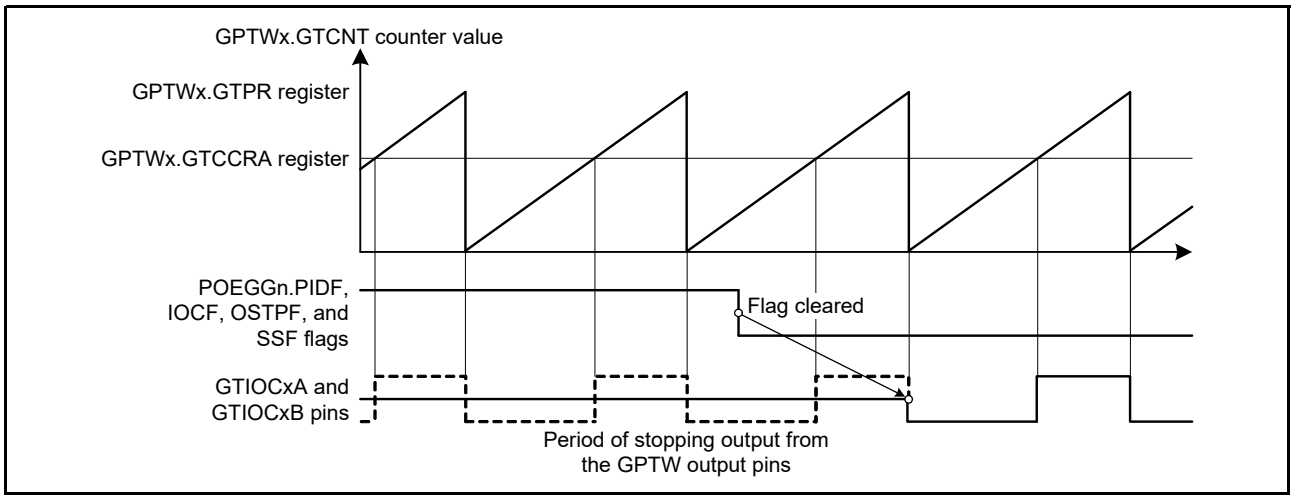


Figure 26.4 Timing of Re-enabling Output from the GPTW Output Pins Following Cancellation of a Request to Stop Output

(3) Cancellation in response to direct input of a detected signal

For details, refer to section 26.3.7, Requests to Stop Output in Response to Detected Signals and Canceling the Requests.

26.3.7 Requests to Stop Output in Response to Detected Signals and Canceling the Requests

The input level detection signals on the GTETRGn pin (n = A to D) and comparator level detection signals on the CMPCm pin (m = 0 to 5) can be used as direct requests to stop output in response to detected signals. The sources of the signals for detection are selected in the GTONCCRn.NFS[3:0] bits, and the active senses of the signals for detection are set in the GTONCCRn.NFV bit. Setting a GTONCCRn.NE bit to activate direct requests to stop output results in the issuing of a request to stop output to the GPTW when the selected source signal for output stopping detection is generated.

The request to stop output is released when the input level detection signal on the given GTETRGn pin or comparator level detection does not match the issuing conditions. To cancel a request, confirm that the values of the PIDF and IOCF flags in the POEGGn register have become 0.

Figure 26.5 shows the operation of a request to stop output issued in response to level detection by the COMP0. The example shows the case where the comparator detects the analog input voltage on CMPC0 becoming higher than the reference voltage while the GPTW is producing PWM waveforms on the GTIOC0A pin on a cyclic-counting basis, and the level-detection signal from the comparator is input to the POEG. The POEG outputs a request to stop output to the GPTW on the basis of detection as described above, and the GPTW remains stopped until the end of the cycle of counting, even if the request to stop output is canceled before then.

If the request to stop output has not been canceled at the end of the cycle of counting cycle, output from the GPTW remains stopped until the end of the next cycle of counting, and so on.

The output-stopped state can be checked by reading the ODF flag in General PWM Timer Status Register (GTST) of the GPTW. For details, refer to the description of the ODF flag in section 24.2.16, General PWM Timer Status Register (GTST). Figure 26.6 shows the procedure for setting requests to stop output in response to detected signals.

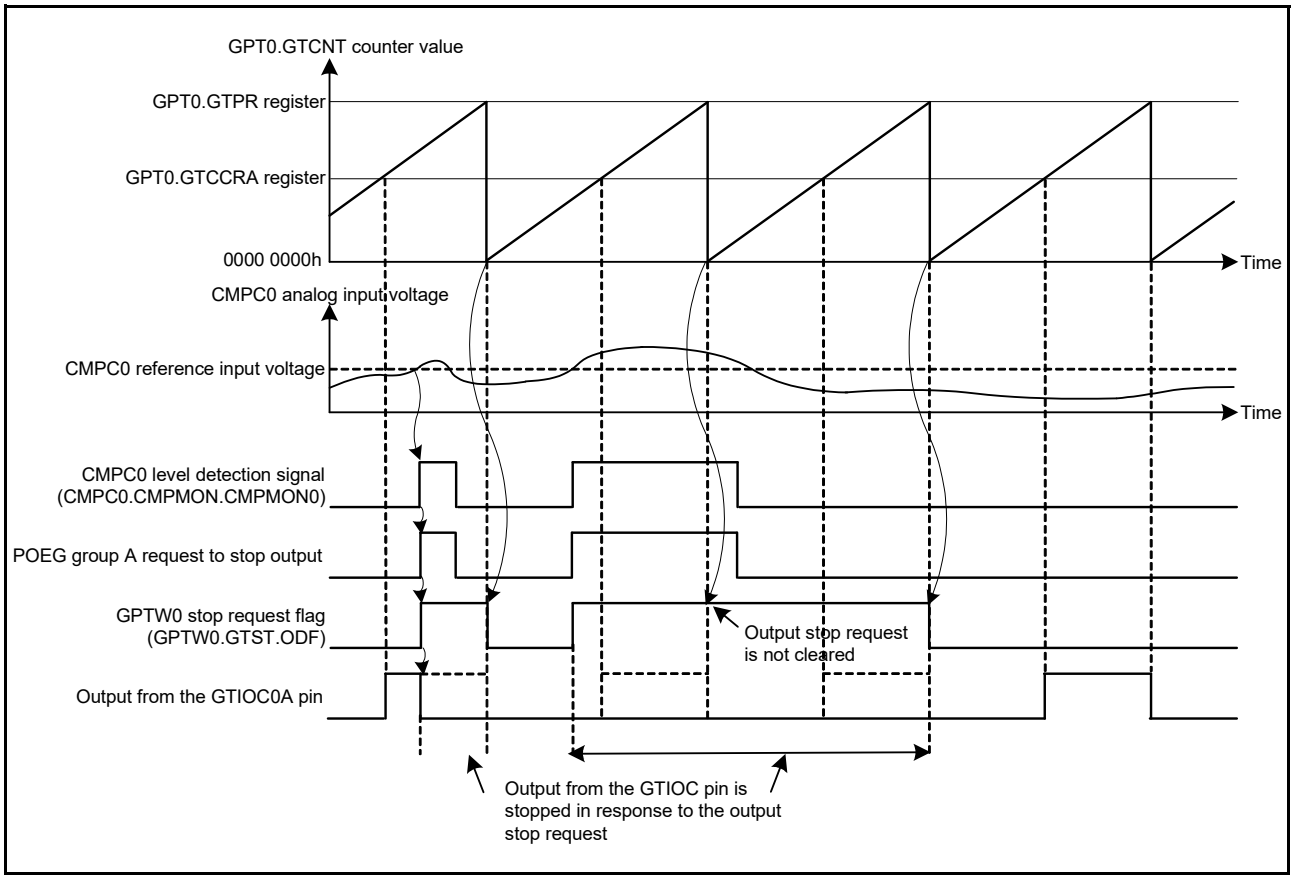
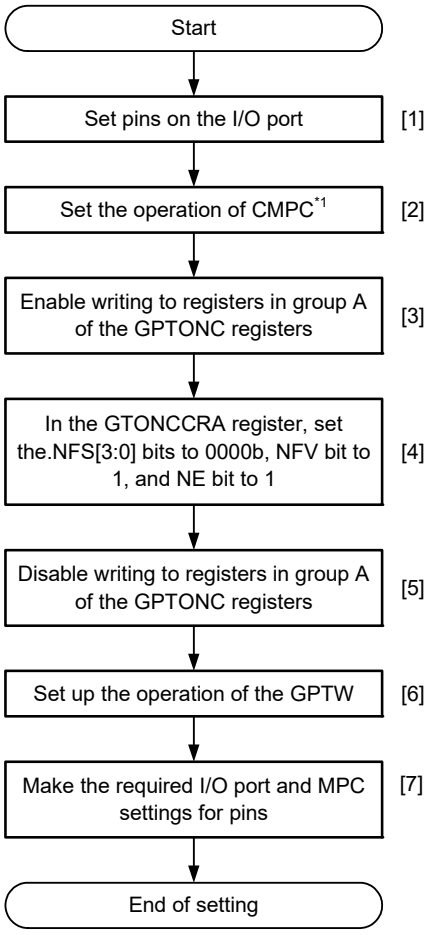


Figure 26.5 Example of Operation to Stop Output from the GTIOC Pin in Response to Level Detection by a Comparator



[1] Make the I/O port settings.
Set the corresponding pins of the I/O port to general purpose I/O port pins in order to use the CMPC0 analog input pin function and GPTW output pin function. For details, refer to section 20, I/O Ports.

[2] Set the operation of CMPC0 to output a level detection signal to the POEG in response to comparing the analog input voltage and reference input voltage. For details, refer to section 45, Comparator C (CMPCa).*1

[3] Write A500h to the GTONCWPA register to enable writing to the GTONCCPRA register.

[4] Select level detection for the comparator by setting the GTONCCRA.NFS[3:0] bits to 0000b. Set the GTONCCRA.NFV bit to 1 so that the request to stop output is sent when the selected source indicates 1.
Set the GTONCCRA.NE bit to 1 to enable the output of requests to stop output.

[5] Write A501h to the GTONCWPA register to disable writing to the GTONCCPRA register.

[6] Set the GPTW for the output of a PWM waveform on its GTIOCn pin. For details, refer to section 24, General PWM Timer (GPTWa).

[7] Set the I/O port and multifunction-pin controller.
Set the I/O port and multifunction-pin controller in order to use the required analog input pin of CMPC0 and output pin of the GPTW. For details, refer to section 20, I/O Ports, and section 21, Multi-Function Pin Controller (MPC).

The settings described above lead to the output of requests to stop output in response to comparison of the analog input voltage and reference input voltage while a PWM waveform is being output from the GPTW pin, stopping output from the GPTW pin.

Note 1. In this example, the comparator output is non-inverted.

Figure 26.6 Example of Settings to Stop Output on the GTIOC Pin in Response to Level Detection by a Comparator

26.4 Interrupt Sources

The POEG generates interrupts for the interrupt controller when any of the following is detected.

- Detection of level of input, indicated by the POEGGn.PIDF flag
- Detection of stopping of output from the GPTW, indicated by the POEGGn.IOCF flag
- Detection of edges by the comparator, indicated by the POEGGn.IOCF flag

Table 26.3 shows interrupt sources and conditions.

Table 26.3 Interrupt sources and conditions

Interrupt Source	Symbol	Corresponding Flag	Trigger Condition
POEG Group A Interrupt	POEGGAI	POEGGA.IOCF	Detection of stopping of output from the GPTW
			Detection of edges by the comparator
		POEGGA.PIDF	Detection of the level input from the GTETRGA pin
POEG Group B Interrupt	POEGGBI	POEGGB.IOCF	Detection of stopping of output from the GPTW
			Detection of edges by the comparator
		POEGGB.PIDF	Detection of the level input from the GTETRGB pin
POEG Group C Interrupt	POEGGCI	POEGGC.IOCF	Detection of stopping of output from the GPTW
			Detection of edges by the comparator
		POEGGC.PIDF	Detection of the level input from the GTETRGC pin
POEG Group D Interrupt	POEGGDI	POEGGD.IOCF	Detection of stopping of output from the GPTW
			Detection of edges by the comparator
		POEGGD.PIDF	Detection of the level input from the GTETRGD pin

26.5 External Trigger Output to GPTW

The POEG outputs the inputs on the GTETRGn pin (n = A to D) as external trigger signals. The external trigger signals can be monitored by the POEGGn.ST flag via active-sense selection and digital noise filtering. The GPTW can function as follows in response to external trigger signals.

- Count start
- Count stop
- Counter clearing
- Up-counting
- Down-counting
- Input capture

For details of the above functions, refer to section 24, General PWM Timer (GPTWa).

26.6 Usage Notes

26.6.1 Transitions to Low-Power Consumption Mode

When the POEG is used, it should not be transitioned to software standby mode. In this mode, the stopping of output cannot be requested since the POEG is stopped.

26.6.2 Setting the Function for Stopping the Module

The module-stop control register can be set to enable or disable operation of the POEG. The POEG is stopped immediately after a reset. Registers of the POEG become accessible following release from the module-stopped state. For details, refer to section 11, Low Power Consumption. The module-stop bit for the POEG also stops and starts the GPTW.

26.6.3 Duplication of Requests to Stop Output

While either the PIDF or IOCF flag in the POEGGn register is 1, cancellation of requests to stop by the detection signal set in the GTONCCRn register does not work due to the request to stop still being output because of the value of the flag. That is, note that requests to stop output will not be canceled when a corresponding flag is set stop output in response to detection. Request signals to stop output in response to flag setting are obtained as the logical OR of the corresponding detection signals for stopping output.

27. 8-Bit Timer (TMRb)

This MCU has four units (unit 0, unit 1, unit 2, unit 3) of an on-chip 8-bit timer (TMR) module that comprise two 8-bit counter channels, totaling eight channels. The 8-bit timer module can be used to count external events and also be used as a multi-function timer in a variety of applications, such as generation of counter reset signal, interrupt requests, and pulse output with a desired duty cycle using a compare-match signal with two registers.
Unit 0, unit 1, unit 2, and unit 3 have the same functions, and unit 0 and unit 1 can generate a base clock for the SCI.
In this section, “PCLK” is used to refer to PCLKB.

27.1 Overview

Table 27.1 lists the specifications of the TMR. Table 27.2 and Table 27.3 list the TMR functions.
Figure 27.1 shows a block diagram of the 8-bit timer module (unit 0), Figure 27.2 shows that of the 8-bit timer module (unit 1), Figure 27.3 shows that of the 8-bit timer module (unit 2), and Figure 27.3 shows that of the 8-bit timer module (unit 3).

Table 27.1 Specifications of TMR

Item	Description
Count clock	• Internal clock: PCLK/1, PCLK/2, PCLK/8, PCLK/32, PCLK/64, PCLK/1024, PCLK/8192 • External clock: external count clock
Number of channels	(8 bits × 2 channels) × 4 units
Compare match	• 8-bit mode (compare match A, compare match B) • 16-bit mode (compare match A, compare match B)
Counter clear	Selected by compare match A or B, or an external counter reset signal.
Timer output	Output pulses with a desired duty cycle or PWM output
Cascading of two channels	• 16-bit count mode 16-bit timer using TMR0 for the upper 8 bits and TMR1 for the lower 8 bits (TMR2 for the upper 8 bits and TMR3 for the lower 8 bits, TMR4 for the upper 8 bits and TMR5 for the lower 8 bits, TMR6 for the upper 8 bits and TMR7 for the lower 8 bits) • Compare match count mode TMR1 can be used to count TMR0 compare matches (TMR3 can be used to count TMR2 compare matches, TMR5 can be used to count TMR4 compare matches, TMR7 can be used to count TMR6 compare matches).
Interrupt sources	Compare match A, compare match B, and overflow
Event link function (Output)	Compare match A, compare match B, and overflow (TMR0 to TMR3)
Event link function (Input)	One of the following three operations proceeds in response to an event reception: (1) Counting start operation (TMR0 to TMR3) (2) Event counting operation (TMR0 to TMR3) (3) Counting restart operation (TMR0 to TMR3)
DTC activation	DTC can be activated by compare match A interrupts or compare match B interrupts.
A/D conversion start trigger of the A/D converter	Compare match A of TMR0, TMR2, TMR4, and TMR6
Capable of generating base clock for SCI	Generates base clock for SCI.*1
Low power consumption function	Each unit can be placed in a module stop state

Note 1. For details, refer to section 32, Serial Communications Interface (SCIk, SCIh).

Table 27.2 TMR Functions (1)

Item		**Unit 0**			**Unit 1**		
Counter mode		**8 Bits**		**16 Bits**	**8 Bits**		**16 Bits**
Channel		**TMR0**	**TMR1**	**TMR0 + TMR1**	**TMR2**	**TMR3**	**TMR2 + TMR3**
Count clock		PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI0	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI1	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI1	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI2	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI3	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI3
Counter clear		TMR0.TCORA TMR0.TCORB TMRI0	TMR1.TCORA TMR1.TCORB TMRI1	TMR0.TCORA + TMR1.TCORA TMR0.TCORB + TMR1.TCORB TMRI0	TMR2.TCORA TMR2.TCORB TMRI2	TMR3.TCORA TMR3.TCORB TMRI3	TMR2.TCORA + TMR3.TCORA TMR2.TCORB + TMR3.TCORB TMRI2
Compare match	Compare match A	✓	✓	✓	✓	✓	✓
	Compare match B	✓	✓	✓	✓	✓	✓
Timer output	Low output	✓	✓	✓	✓	✓	✓
	High output	✓	✓	✓	✓	✓	✓
	Toggle output	✓	✓	✓	✓	✓	✓
DTC activation	Compare match A	✓	✓	✓	✓	✓	✓
	Compare match B	✓	✓	✓	✓	✓	✓
	TCNT overflow	—	—	—	—	—	—
Interrupt	Compare match A	CMIA0	CMIA1	CMIA0	CMIA2	CMIA3	CMIA2
	Compare match B	CMIB0	CMIB1	CMIB0	CMIB2	CMIB3	CMIB2
	TCNT overflow	OVI0	OVI1	OVI0	OVI2	OVI3	OVI2
Cascaded connection		TMR1 overflow	TMR0 compare match A	—	TMR3 overflow	TMR2 compare match A	—
A/D conversion start trigger of the A/D converter*1		✓	—	✓	✓	—	✓
SCI base clock generation*2		✓		—	✓		—
ELC output event	Compare match A	✓	✓	✓	✓	✓	✓
	Compare match B	✓	✓	✓	✓	✓	✓
	TCNT overflow	✓	✓	✓	✓	✓	✓
ELC input event	Counting start	✓	✓	—	✓	✓	—
	Event counting	✓	✓	—	✓	✓	—
	Counting restart	✓	✓	—	✓	✓	—
Module stop setting*3		MSTPCRA.MSTPA5 bit (unit 0), MSTPCRA.MSTPA4 bit (unit 1)					

Table 27.3 TMR Functions (2)

Item		Unit 2			Unit 3		
Counter mode		**8 Bits**		**16 Bits**	**8 Bits**		**16 Bits**
Channel		**TMR4**	**TMR5**	**TMR4 + TMR5**	**TMR6**	**TMR7**	**TMR6 + TMR7**
Count clock		PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI4	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI5	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI5	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI6	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI7	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI7
Counter clear		TMR4.TCORA TMR4.TCORB TMRI4	TMR5.TCORA TMR5.TCORB TMRI5	TMR4.TCORA + TMR5.TCORA TMR4.TCORB + TMR5.TCORB TMRI4	TMR6.TCORA TMR6.TCORB TMRI6	TMR7.TCORA TMR7.TCORB TMRI7	TMR6.TCORA + TMR7.TCORA TMR6.TCORB + TMR7.TCORB TMRI6
Compare match	Compare match A	✓	✓	✓	✓	✓	✓
	Compare match B	✓	✓	✓	✓	✓	✓
Timer output	Low output	✓	✓	✓	✓	✓	✓
	High output	✓	✓	✓	✓	✓	✓
	Toggle output	✓	✓	✓	✓	✓	✓
DTC activation	Compare match A	✓	✓	✓	✓	✓	✓
	Compare match B	✓	✓	✓	✓	✓	✓
	TCNT overflow	—	—	—	—	—	—
Interrupt	Compare match A	CMIA4	CMIA5	CMIA4	CMIA6	CMIA7	CMIA6
	Compare match B	CMIB4	CMIB5	CMIB4	CMIB6	CMIB7	CMIB6
	TCNT overflow	OVI4	OVI5	OVI4	OVI6	OVI7	OVI6
Cascaded connection		TMR5 overflow	TMR4 compare match A	—	TMR7 overflow	TMR6 compare match A	—
A/D conversion start trigger of the A/D converter*1		✓	—	✓	✓	—	✓
SCI base clock generation*2		—	—	—	—	—	—
ELC output event	Compare match A	✓	✓	✓	✓	✓	✓
	Compare match B	✓	✓	✓	✓	✓	✓
	TCNT overflow	✓	✓	✓	✓	✓	✓
ELC input event	Counting start	✓	✓	—	✓	✓	—
	Event counting	✓	✓	—	✓	✓	—
	Counting restart	✓	✓	—	✓	✓	—
Module stop setting*3		MSTPCRA.MSTPA3 bit (unit 2), MSTPCRA.MSTPA2 bit (unit 3)					

✓: Possible

—: Impossible

Note 1. For details, refer to section 42, 12-Bit A/D Converter (S12ADHa).

Note 2. For details, refer to section 32, Serial Communications Interface (SCIk, SCIh).

Note 3. For details, refer to section 11, Low Power Consumption.

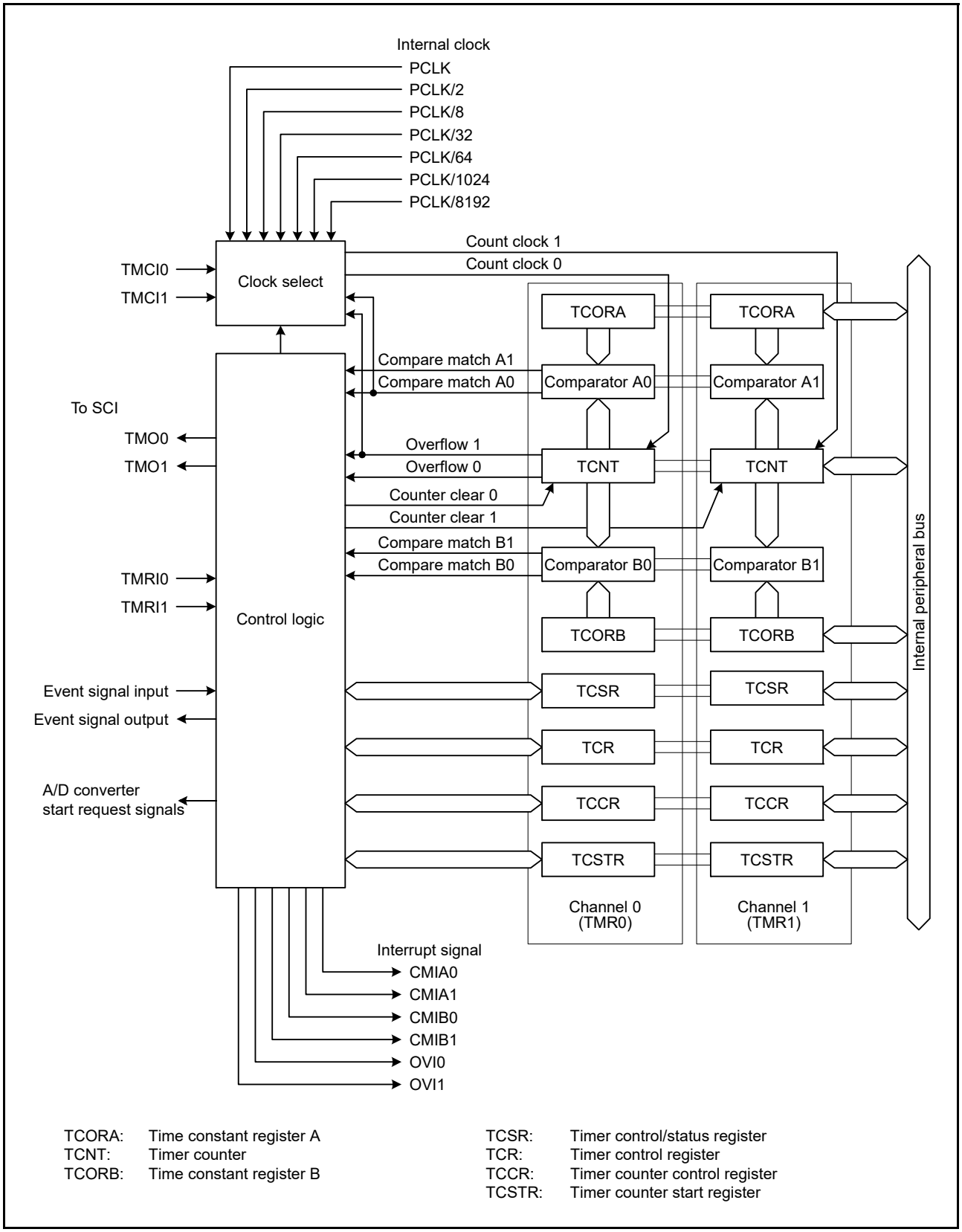


Figure 27.1 Block Diagram of TMR (Unit 0)

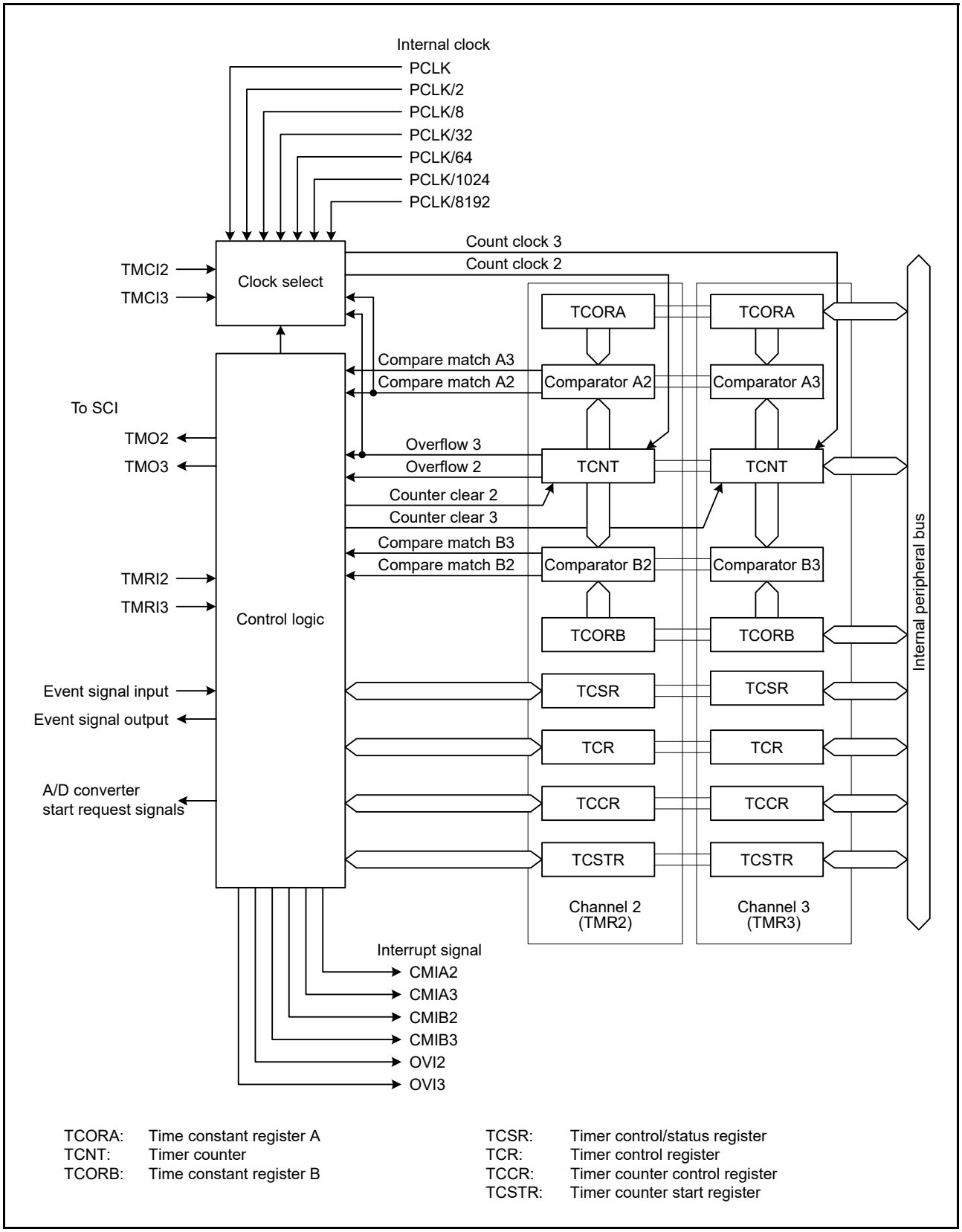


Figure 27.2 Block Diagram of TMR (Unit 1)

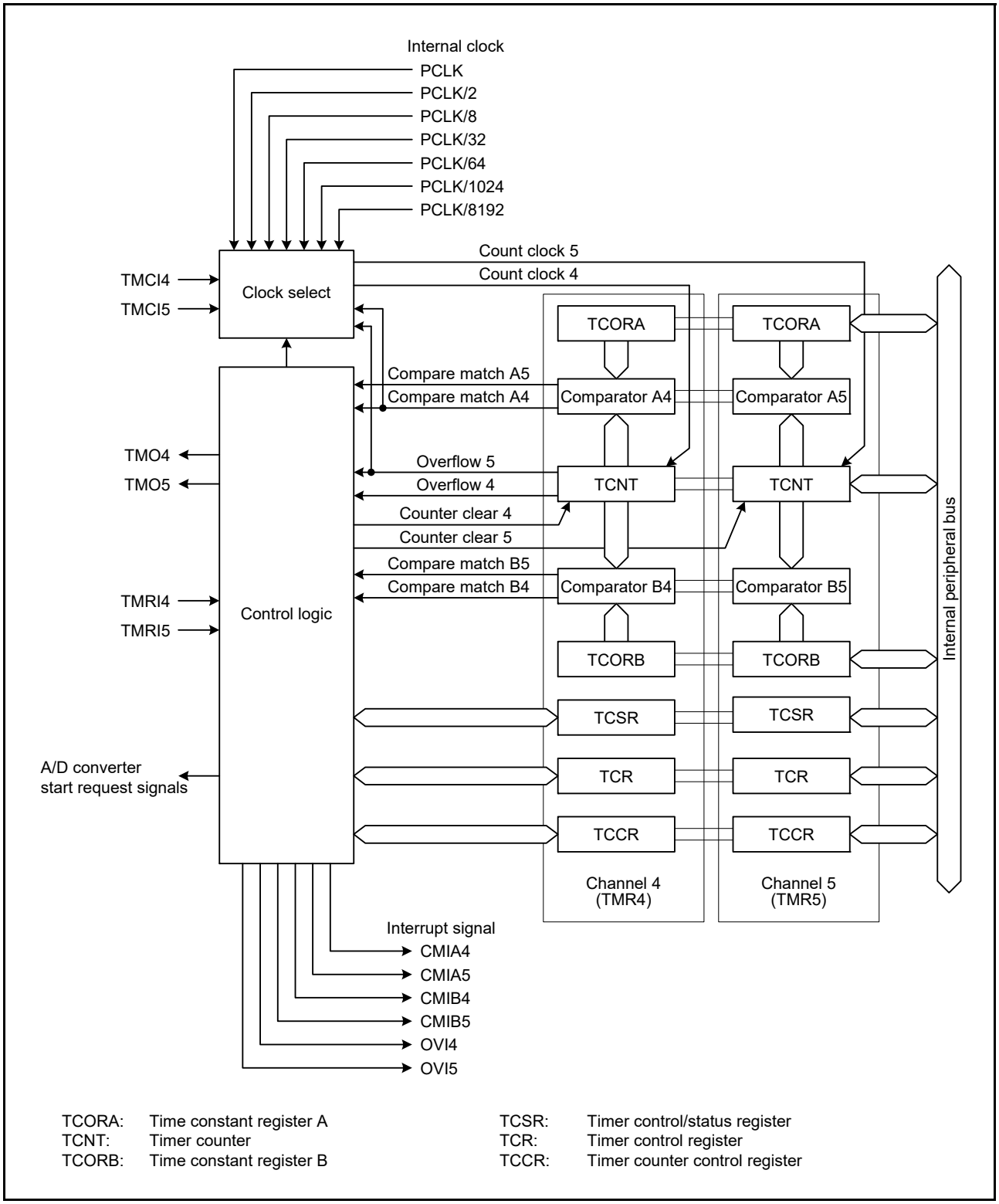


Figure 27.3 Block Diagram of TMR (Unit 2)

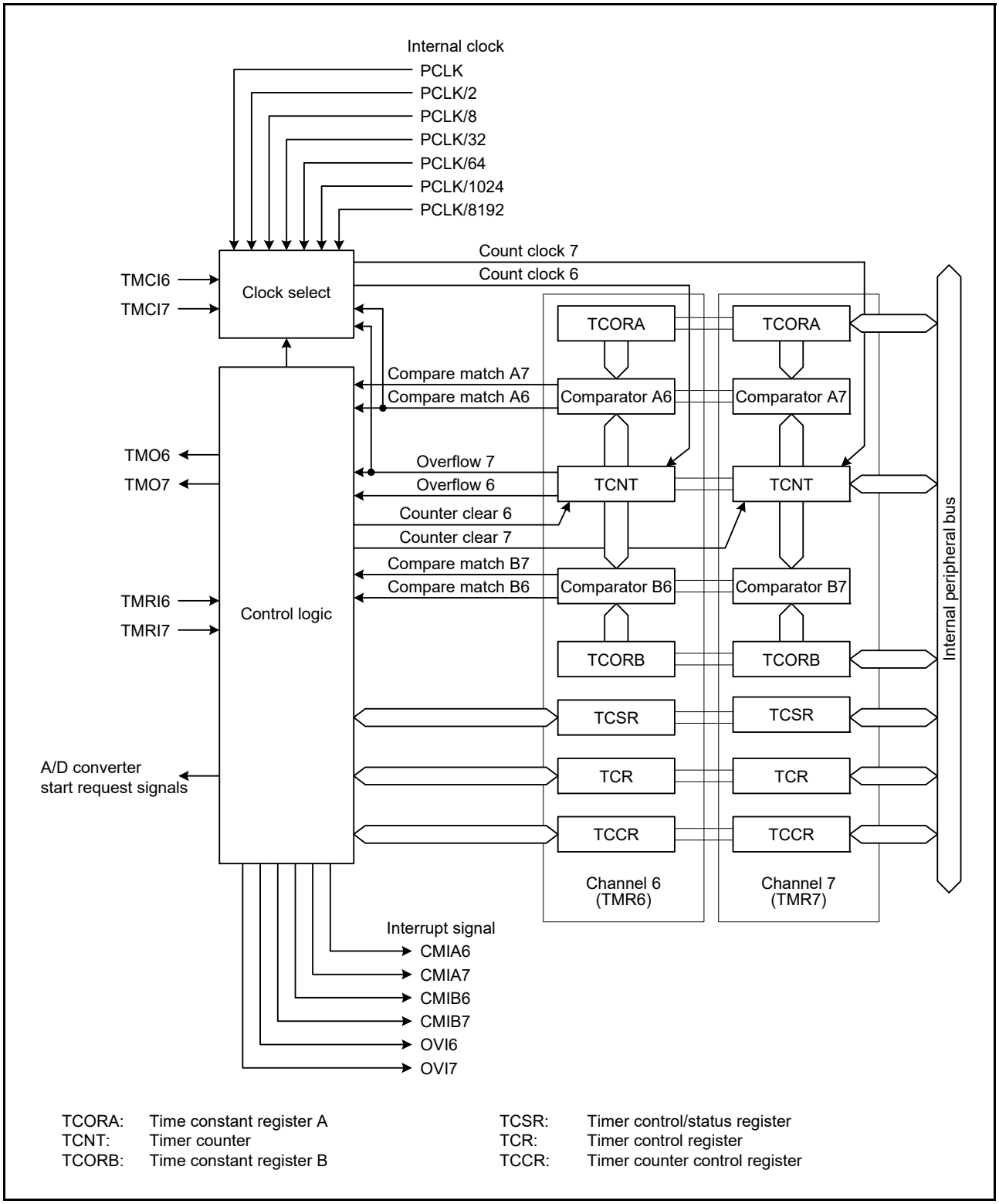


Figure 27.4 Block Diagram of TMR (Unit 3)

Table 27.4 lists the I/O pins of the TMR.

Table 27.4 Pin Configuration of TMR

Unit	Channel	Pin Name	I/O	Description
0	TMR0	TMO0	Output	Outputs compare match
		TMCI0	Input	Inputs external count clock
		TMRI0	Input	Inputs external counter reset
	TMR1	TMO1	Output	Outputs compare match
		TMCI1	Input	Inputs external count clock
		TMRI1	Input	Inputs external counter reset
1	TMR2	TMO2	Output	Outputs compare match
		TMCI2	Input	Inputs external count clock
		TMRI2	Input	Inputs external counter reset
	TMR3	TMO3	Output	Outputs compare match
		TMCI3	Input	Inputs external count clock
		TMRI3	Input	Inputs external counter reset
2	TMR4	TMO4	Output	Outputs compare match
		TMCI4	Input	Inputs external count clock
		TMRI4	Input	Inputs external counter reset
	TMR5	TMO5	Output	Outputs compare match
		TMCI5	Input	Inputs external count clock
		TMRI5	Input	Inputs external counter reset
3	TMR6	TMO6	Output	Outputs compare match
		TMCI6	Input	Inputs external count clock
		TMRI6	Input	Inputs external counter reset
	TMR7	TMO7	Output	Outputs compare match
		TMCI7	Input	Inputs external count clock
		TMRI7	Input	Inputs external counter reset

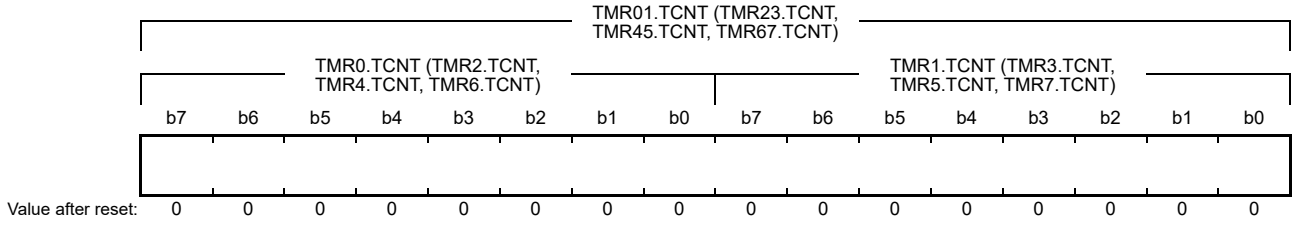
27.2 Register Descriptions

Table 27.5 Register Allocation for 16-Bit Access

Address	Register	Upper 8 Bits	Lower 8 Bits
0008 8208h	TMR01.TCNT	TMR0.TCNT	TMR1.TCNT
0008 8204h	TMR01.TCORA	TMR0.TCORA	TMR1.TCORA
0008 8206h	TMR01.TCORB	TMR0.TCORB	TMR1.TCORB
0008 820Ah	TMR01.TCCR	TMR0.TCCR	TMR1.TCCR
0008 8218h	TMR23.TCNT	TMR2.TCNT	TMR3.TCNT
0008 8214h	TMR23.TCORA	TMR2.TCORA	TMR3.TCORA
0008 8216h	TMR23.TCORB	TMR2.TCORB	TMR3.TCORB
0008 821Ah	TMR23.TCCR	TMR2.TCCR	TMR3.TCCR
0008 8228h	TMR45.TCNT	TMR4.TCNT	TMR5.TCNT
0008 8224h	TMR45.TCORA	TMR4.TCORA	TMR5.TCORA
0008 8226h	TMR45.TCORB	TMR4.TCORB	TMR5.TCORB
0008 822Ah	TMR45.TCCR	TMR4.TCCR	TMR5.TCCR
0008 8238h	TMR67.TCNT	TMR6.TCNT	TMR7.TCNT
0008 8234h	TMR67.TCORA	TMR6.TCORA	TMR7.TCORA
0008 8236h	TMR67.TCORB	TMR6.TCORB	TMR7.TCORB
0008 823Ah	TMR67.TCCR	TMR6.TCCR	TMR7.TCCR

27.2.1 Timer Counter (TCNT)

Address(es): TMR0.TCNT 0008 8208h, TMR1.TCNT 0008 8209h, TMR2.TCNT 0008 8218h, TMR3.TCNT 0008 8219h, TMR4.TCNT 0008 8228h, TMR5.TCNT 0008 8229h, TMR6.TCNT 0008 8238h, TMR7.TCNT 0008 8239h, TMR01.TCNT 0008 8208h, TMR23.TCNT 0008 8218h, TMR45.TCNT 0008 8228h, TMR67.TCNT 0008 8238h



TCNT is an 8-bit readable/writable up-counter.

TMR0.TCNT and TMR1.TCNT (TMR2.TCNT and TMR3.TCNT, TMR4.TCNT and TMR5.TCNT, TMR6.TCNT and TMR7.TCNT) comprise a single 16-bit counter (TMR01.TCNT, TMR23.TCNT, TMR45.TCNT, TMR67.TCNT) so they can be accessed together in 16-bit units.

The TCCR.CSS[1:0] and CKS[2:0] bits are used to select a count clock.

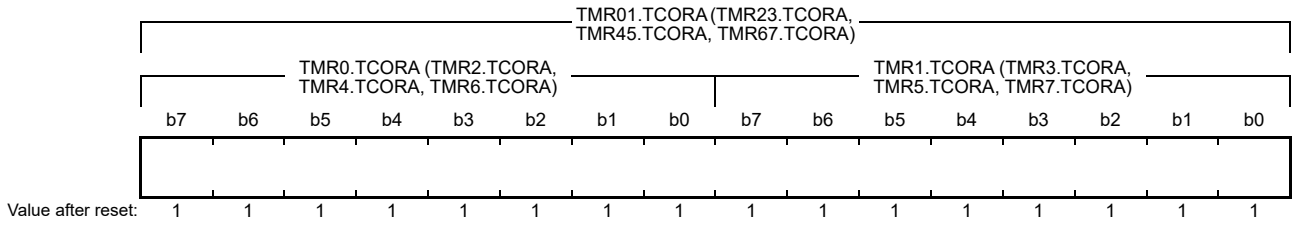
TCNT can be cleared by an external counter reset signal, compare match A, or compare match B. Which compare match to be used for clearing is selected by the TCR.CCLR[1:0] bits.

When TCNT overflows (its value changes from FFh to 00h), an overflow interrupt is output provided the interrupt request is enabled by the TCR.OVIE bit.

For details on the corresponding interrupt vector number, refer to section 14, Interrupt Controller (ICUG), and Table 27.7, TMR Interrupt Sources.

27.2.2 Time Constant Register A (TCORA)

Address(es): TMR0.TCORA 0008 8204h, TMR1.TCORA 0008 8205h, TMR2.TCORA 0008 8214h, TMR3.TCORA 0008 8215h, TMR4.TCORA 0008 8224h, TMR5.TCORA 0008 8225h, TMR6.TCORA 0008 8234h, TMR7.TCORA 0008 8235h, TMR01.TCORA 0008 8204h, TMR23.TCORA 0008 8214h, TMR45.TCORA 0008 8224h, TMR67.TCORA 0008 8234h



TCORA is an 8-bit readable/writable register.

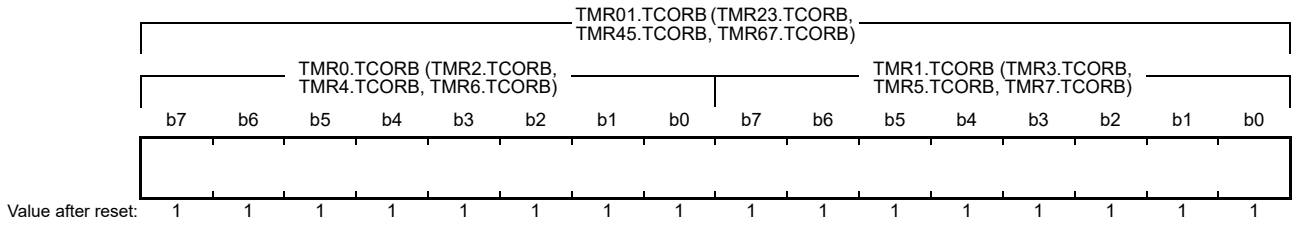
TMR0.TCORA and TMR1.TCORA (TMR2.TCORA and TMR3.TCORA, TMR4.TCORA and TMR5.TCORA, TMR6.TCORA and TMR7.TCORA) comprise a single 16-bit register (TMR01.TCORA, TMR23.TCORA, TMR45.TCORA, TMR67.TCORA) so they can be accessed together in 16-bit units.

The value in TCORA is continually compared with the value in TCNT. When a match is detected, the corresponding compare match A is generated, and a compare match A interrupt is output provided the interrupt request is enabled by the TCR.CMIEA bit.

However, comparison is not performed during writing to TCORA. The timer output from the TMOn pin can be freely controlled by this compare match A and the settings of the TCSR.OSA[1:0] bits.

27.2.3 Time Constant Register B (TCORB)

Address(es): TMR0.TCORB 0008 8206h, TMR1.TCORB 0008 8207h, TMR2.TCORB 0008 8216h, TMR3.TCORB 0008 8217h, TMR4.TCORB 0008 8226h, TMR5.TCORB 0008 8227h, TMR6.TCORB 0008 8236h, TMR7.TCORB 0008 8237h, TMR01.TCORB 0008 8206h, TMR23.TCORB 0008 8216h, TMR45.TCORB 0008 8226h, TMR67.TCORB 0008 8236h



TCORB is an 8-bit readable/writable register.

TMR0.TCORB and TMR1.TCORB (TMR2.TCORB and TMR3.TCORB, TMR4.TCORB and TMR5.TCORB, TMR6.TCORB and TMR7.TCORB) comprise a single 16-bit register (TMR01.TCORB, TMR23.TCORB, TMR45.TCORB, TMR67.TCORB) so they can be accessed together in 16-bit units.

The value in TCORB is continually compared with the value in TCNT. When a match is detected, the corresponding compare match B is generated, and a compare match B interrupt is output provided the interrupt request is enabled by the TCR.CMIEB bit.

However, comparison is not performed during writing to TCORB. The timer output from the TMOn pin can be freely controlled by this compare match B and the settings of the TCSR.OSB[1:0] bits.

27.2.4 Timer Control Register (TCR)

Address(es): TMR0.TCR 0008 8200h, TMR1.TCR 0008 8201h, TMR2.TCR 0008 8210h, TMR3.TCR 0008 8211h, TMR4.TCR 0008 8220h, TMR5.TCR 0008 8221h, TMR6.TCR 0008 8230h, TMR7.TCR 0008 8231h

	b7	b6	b5	b4	b3	b2	b1	b0
	CMIEB	CMIEA	OVIE	CCLR[1:0]		—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4, b3	CCLR[1:0]	Counter Clear	b4 b3 0 0: Clearing is disabled 0 1: Cleared by compare match A 1 0: Cleared by compare match B 1 1: Cleared by the external counter reset signal*1 (Select edge or level by the TMRIS bit in TCCR.)	R/W
b5	OVIE	Overflow Interrupt Enable	0: Overflow interrupt requests (OVIn) are disabled 1: Overflow interrupt requests (OVIn) are enabled	R/W
b6	CMIEA	Compare Match A Interrupt Enable	0: Compare match A interrupt requests (CMIAn) are disabled 1: Compare match A interrupt requests (CMIAn) are enabled	R/W
b7	CMIEB	Compare Match B Interrupt Enable	0: Compare match B interrupt requests (CMIBn) are disabled 1: Compare match B interrupt requests (CMIBn) are enabled	R/W

Note 1. To use an external counter reset signal, set the corresponding pin function. For details, refer to section 20, I/O Ports and section 21, Multi-Function Pin Controller (MPC).

CCLR[1:0] Bits (Counter Clear)

Select the condition by which TCNT is cleared.

OVIE Bit (Overflow Interrupt Enable)

Selects whether overflow interrupt requests (OVIn) issued by TCNT are enabled or disabled.

CMIEA Bit (Compare Match A Interrupt Enable)

Selects whether compare match A interrupt requests (CMIAn) that are issued when the value of TCORA corresponds to that of TCNT are enabled or disabled.

CMIEB Bit (Compare Match B Interrupt Enable)

Selects whether compare match B interrupt requests (CMIBn) that are issued when the value of TCORB corresponds to that of TCNT are enabled or disabled.

27.2.5 Timer Counter Control Register (TCCR)

Address(es): TMR0.TCCR 0008 820Ah, TMR1.TCCR 0008 820Bh, TMR2.TCCR 0008 821Ah, TMR3.TCCR 0008 821Bh, TMR4.TCCR 0008 822Ah, TMR5.TCCR 0008 822Bh, TMR6.TCCR 0008 823Ah, TMR7.TCCR 0008 823Bh, TMR01.TCCR 0008 820Ah, TMR23.TCCR 0008 821Ah, TMR45.TCCR 0008 822Ah, TMR67.TCCR 0008 823Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	TMRIS	—	—	CSS[1:0]		CKS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CKS[2:0]	Clock Select*1	See Table 27.6.	R/W
b4, b3	CSS[1:0]	Clock Source Select	See Table 27.6.	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	TMRIS	Timer Reset Detection Condition Select	0: Cleared at rising edge of the external counter reset signal 1: Cleared when the external counter reset signal is high	R/W

Note 1. To use an external count clock, set the corresponding pin function. For details, refer to section 20, I/O Ports and section 21, Multi-Function Pin Controller (MPC).

TCCR register is a 8-bit register used to configure the basic operation of the counter. Two TCCR registers can be accessed simultaneously by accessing the address of the even channel TCCR register in 16-bit units.

CKS[2:0] Bits (Clock Select)
CSS[1:0] Bits (Clock Source Select)

The CKS[2:0] and CSS[1:0] bits select a clock. For details, see Table 27.6.

TMRIS Bit (Timer Reset Detection Condition Select)

This bit is enabled when the TCR.CCLR[1:0] bits are 11b (cleared by external counter reset signal) and selects the condition for detecting counter reset (level or edge).

Table 27.6 Clock Input to TCNT and Count Condition

Channel	TCCR Register CSS[1:0] b4	b3	CKS[2:0] b2	b1	b0	Description
TMR0 (TMR2, TMR4, TMR6)	0	0	—	0	0	Clock input prohibited
					1	Uses external count clock. Counts at rising edge*1.
				1	0	Uses external count clock. Counts at falling edge*1.
					1	Uses external count clock. Counts at both rising and falling edges*1.
	0	1	0	0	0	Uses internal clock. Counts at PCLK.
					1	Uses internal clock. Counts at PCLK/2.
				1	0	Uses internal clock. Counts at PCLK/8.
					1	Uses internal clock. Counts at PCLK/32.
			1	0	0	Uses internal clock. Counts at PCLK/64.
					1	Uses internal clock. Counts at PCLK/1024.
				1	0	Uses internal clock. Counts at PCLK/8192.
					1	Clock input prohibited
	1	0	—	—	—	Setting prohibited
	1	1	—	—	—	Counts at TMR1.TCNT (TMR3.TCNT, TMR5.TCNT, TMR7.TCNT) overflow signal*2.
TMR1 (TMR3, TMR5, TMR7)	0	0	—	0	0	Clock input prohibited
					1	Uses external count clock. Counts at rising edge*1.
				1	0	Uses external count clock. Counts at falling edge*1.
					1	Uses external count clock. Counts at both rising and falling edges*1.
	0	1	0	0	0	Uses internal clock. Counts at PCLK.
					1	Uses internal clock. Counts at PCLK/2.
				1	0	Uses internal clock. Counts at PCLK/8.
					1	Uses internal clock. Counts at PCLK/32.
			1	0	0	Uses internal clock. Counts at PCLK/64.
					1	Uses internal clock. Counts at PCLK/1024.
				1	0	Uses internal clock. Counts at PCLK/8192.
					1	Clock input prohibited
	1	0	—	—	—	Setting prohibited
	1	1	—	—	—	Counts at TMR0.TCNT (TMR2.TCNT, TMR4.TCNT, TMR6.TCNT) compare match A*2.

Note 1. To use an external count clock, set the corresponding pin function. For details, refer to section 20, I/O Ports and section 21, Multi-Function Pin Controller (MPC).

Note 2. If the clock input of TMR0 (TMR2, TMR4, TMR6) is the overflow signal of the TMR1.TCNT (TMR3.TCNT, TMR5.TCNT, TMR7.TCNT) counter and that of TMR1 (TMR3, TMR5, TMR7) is the compare match signal of the TMR0.TCNT (TMR2.TCNT, TMR4.TCNT, TMR6.TCNT) counter, no TCNT count clock is generated. Do not use this setting.

27.2.6 Timer Control/Status Register (TCSR)

- TMR0.TCSR, TMR2.TCSR, TMR4.TCSR, TMR6.TCSR

Address(es): TMR0.TCSR 0008 8202h, TMR2.TCSR 0008 8212h, TMR4.TCSR 0008 8222h, TMR6.TCSR 0008 8232h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	ADTE	OSB[1:0]		OSA[1:0]	
Value after reset:	x	x	x	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b1, b0	OSA[1:0]	Output Select A*1	b1 b0 0 0: No change 0 1: Low is output 1 0: High is output 1 1: Output is inverted (toggle output)	R/W
b3, b2	OSB[1:0]	Output Select B*1	b3 b2 0 0: No change 0 1: Low is output 1 0: High is output 1 1: Output is inverted (toggle output)	R/W
b4	ADTE	A/D Trigger Enable	0: A/D conversion start request in response to compare match A is disabled. 1: A/D conversion start request in response to compare match A is enabled.	R/W
b7 to b5	—	Reserved	These bits are read as an undefined value. The write value should be 1.	R/W

Note 1. When all OSA[1:0] and OSB[1:0] bits are 0, the output enable signal corresponding to the TMOn pin is negated and a request for high-impedance output is issued to the I/O port. Timer output pin is driven low until the first compare-match occurs after a reset when at least one of the OSA[1:0] and OSB[1:0] bits is 1.

OSA[1:0] Bits (Output Select A)

These bits select a method of TMOn pin output when compare match A of TCORA and TCNT occurs.

OSB[1:0] Bits (Output Select B)

These bits select a method of TMOn pin output when compare match B of TCORB and TCNT occurs.

- TMR1.TCSR, TMR3.TCSR, TMR5.TCSR, TMR7.TCSR

Address(es): TMR1.TCSR 0008 8203h, TMR3.TCSR 0008 8213h, TMR5.TCSR 0008 8223h, TMR7.TCSR 0008 8233h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	OSB[1:0]		OSA[1:0]	
Value after reset:	x	x	x	1	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b1, b0	OSA[1:0]	Output Select A*1	b1 b0 0 0: No change 0 1: Low is output 1 0: High is output 1 1: Output is inverted (toggle output)	R/W
b3, b2	OSB[1:0]	Output Select B*1	b3 b2 0 0: No change 0 1: Low is output 1 0: High is output 1 1: Output is inverted (toggle output)	R/W
b4	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b7 to b5	—	Reserved	These bits are read as an undefined value. The write value should be 1.	R/W

Note 1. When all OSA[1:0] and OSB[1:0] bits are 0, the output enable signal corresponding to the TMOn pin is negated and a request for high-impedance output is issued to the I/O port. Timer output pin is driven low until the first compare match occurs after a reset when at least one of the OSA[1:0] and OSB[1:0] bits is 1.

OSA[1:0] Bits (Output Select A)

These bits select a method of TMOn pin output when compare match A of TCORA and TCNT occurs.

OSB[1:0] Bits (Output Select B)

These bits select a method of TMOn pin output when compare match B of TCORB and TCNT occurs.

27.2.7 Timer Counter Start Register (TCSTR)

Address(es): TMR0.TCSTR 0008 820Ch, TMR1.TCSTR 0008 820Dh, TMR2.TCSTR 0008 821Ch, TMR3.TCSTR 0008 821Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	TCS
Value after reset:	x	x	x	x	x	x	x	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	TCS	Timer Counter Status	0: Count stopped state in response to ELC. 1: Count start state in response to ELC.	R/W
b7 to b1	—	Reserved	These bits are read as an undefined value. The write value should be 0.	R/W

TCS Bit (Timer Counter Status)

The TCS bit is used to check the state of the timer count in response to ELC.

When this bit is read as 1, it shows the timer start state in response to ELC. When this bit is read as 0, it shows the timer stopped state in response to ELC.

This bit is cleared by writing 0. Do not write 1 to this bit.

The TCS bit is valid only when the count start operation is selected by the ELOPD register of the event controller (ELC). For details, refer to section 27.7, Link Operation by ELC, or section 19, Event Link Controller (ELC).

27.3 Operation

27.3.1 Pulse Output

Figure 27.5 shows an example of the 8-bit timer being used to generate a pulse output with a desired duty cycle.

1. Set the TCR.CCLR[1:0] bits to 01b (cleared by compare match A) so that TCNT is cleared at a compare match of TCORA.
2. Set the TCSR.OSA[1:0] bits to 10b (high is output) and TCSR.OSB[1:0] bits to 01b (low is output), causing the output to change to high at a compare match of TCORA and to low at a compare match of TCORB.

With these settings, the 8-bit timer provides pulses output at a cycle determined by TCORA with a pulse width determined by TCORB. No software intervention is required.
The timer output pin is low after the TCSR.OSA[1:0] or TCSR.OSB[1:0] bits are set until the first compare match occurs after a reset.

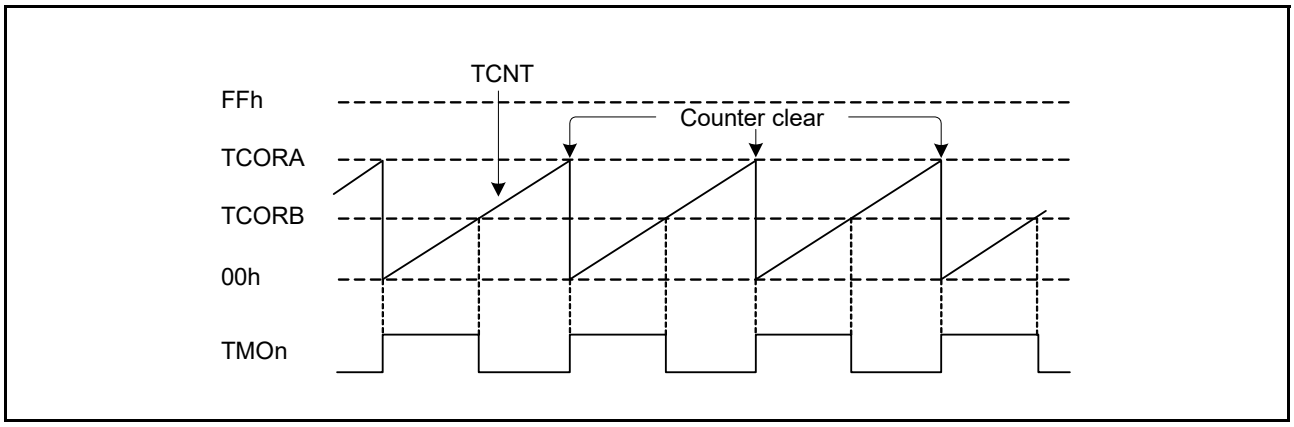


Figure 27.5 Example of Pulse Output (n = 0 to 7)

27.3.2 External Counter Reset Input

Figure 27.6 shows an example of the 8-bit timer being used to generate a pulse which is output after a desired delay time from a TMRIn input.

1. Set the TCR.CCLR[1:0] bits to 11b (cleared by external counter reset signal) and set the TMRIS bit in TCCR to 1 (cleared when the external counter reset signal is high) so that TCNT is cleared at the high level input of the TMRIn signal.
2. Set the TCSR.OSA[1:0] bits to 10b (high output) and the TCSR.OSB[1:0] bits to 01b (low output), causing the output to change to high at a compare match of TCORA and to low at a compare match of TCORB.

With these settings, the 8-bit timer provides pulses output at a desired delay time from a TMRIn input determined by TCORA and with a pulse width determined by TCORB and TCORA.

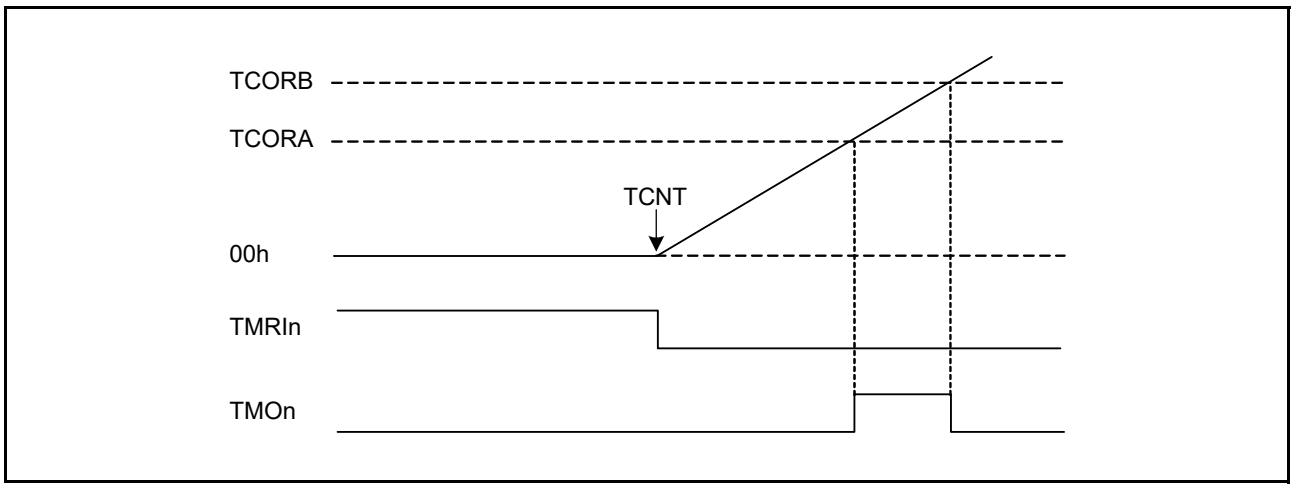


Figure 27.6 Example of External Counter Reset Signal Input (n = 0 to 7)

27.4 Operation Timing

27.4.1 TCNT Count Timing

Figure 27.7 shows the count timing of TCNT for internal clock. Figure 27.8 shows the count timing of TCNT for external clock.

Note that the external clock pulse width must be at least 1.5 PCLK cycles for increment at a single edge, and at least 2.5 PCLK cycles for increment at both edges. The counter will not increment correctly if the pulse width is less than these values.

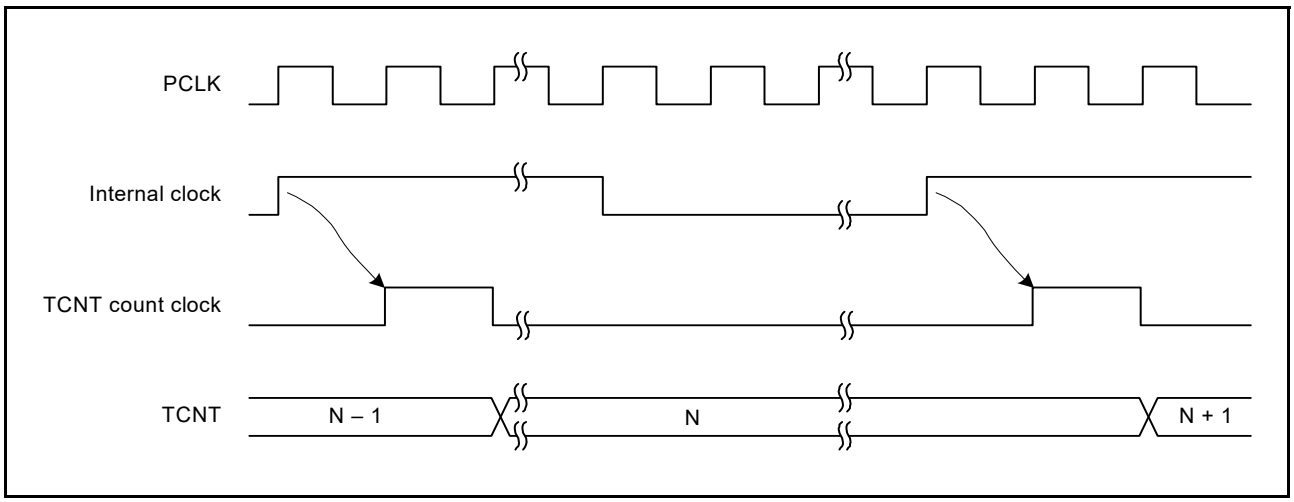


Figure 27.7 Count Timing for Internal Clock

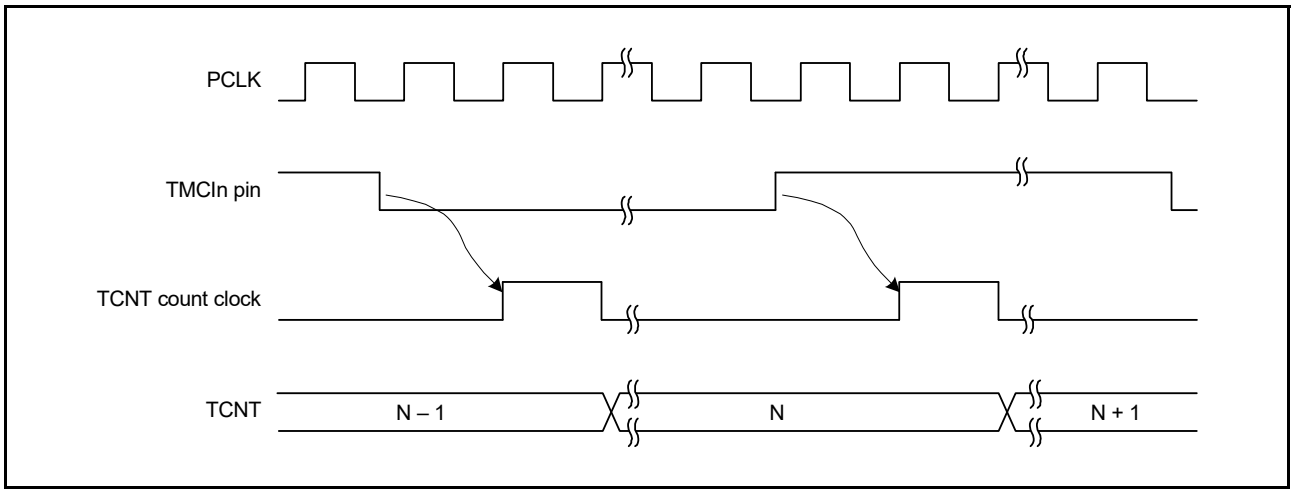


Figure 27.8 Count Timing for External Clock (at Both Edges)

27.4.2 Timing of Interrupt Signal Output on a Compare Match

A compare match refers to a match between the value of the TCORA or TCORB register and the TCNT, and a compare match interrupt signal is output at this time if the interrupt request is enabled. The compare match is generated in the last cycle in which the values match (at the time at which the value counted by TCNT to produce the match is updated). Accordingly, after a match between TCNT and the TCORA or TCORB register is detected, the compare match is not actually generated until the next cycle of the TCNT count clock. Figure 27.9 shows the timing of output of the interrupt signal.
For the corresponding interrupt vector number, refer to section 14, Interrupt Controller (ICUG) and Table 27.7.

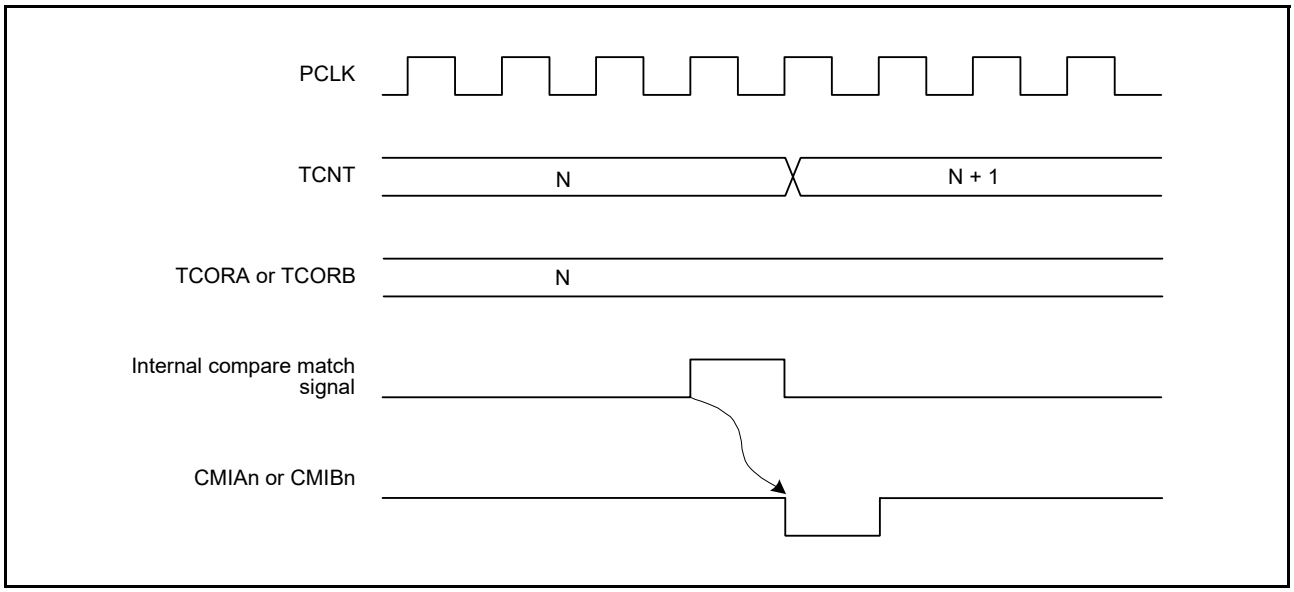


Figure 27.9 Timing of Interrupt Flag Setting to 1 at Compare Match (n = 0 to 7)

27.4.3 Timing of Timer Output Signal at Compare Match

When a compare match signal is generated, the output value specified by the TCSR.OSA[1:0] and OSB[1:0] bits is output on the timer output pin (TMOn).
Figure 27.10 shows the timing when the timer output is toggled by the compare match A signal.

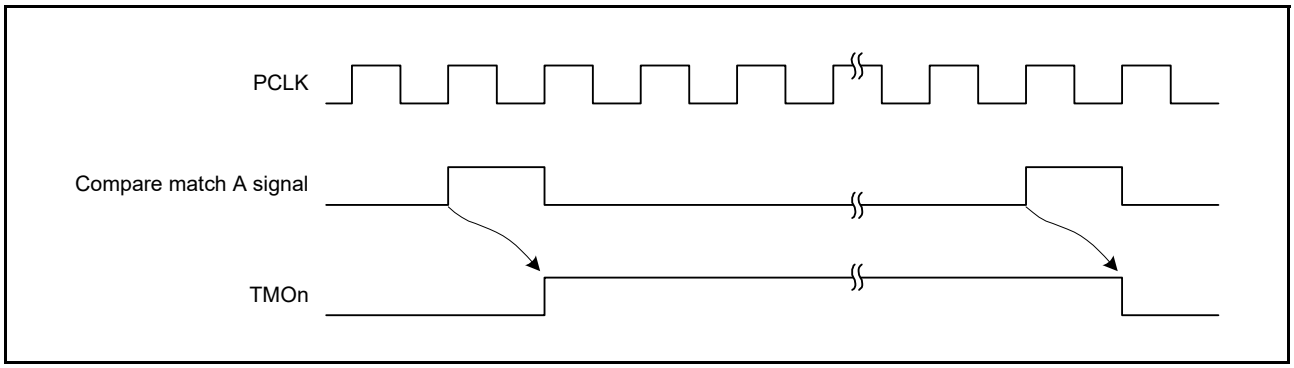


Figure 27.10 Timing of Timer Output Signal at Compare Match A Signal (n = 0 to 7)

27.4.4 Timing of Counter Clear by Compare Match

TCNT is cleared when compare match A or B occurs, depending on the settings of the TCR.CCLR[1:0] bits.
Figure 27.11 shows the timing of this operation.

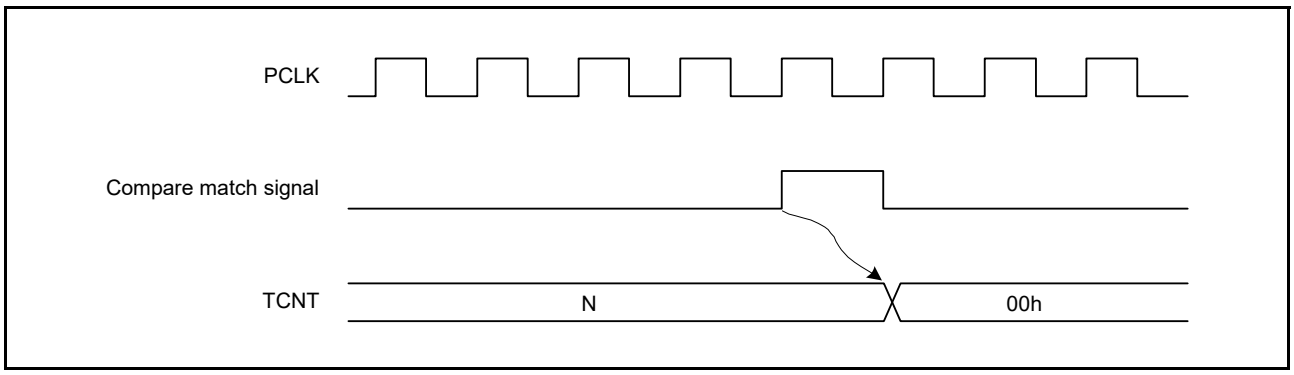


Figure 27.11 Timing of Counter Clear by Compare Match

27.4.5 Timing of the External Reset for TCNT

TCNT is cleared at the rising edge or high level of an external counter reset signal, depending on the settings of the TCR.CCLR[1:0] bits. At least 2 PCLK cycles are required from a reset input to clearing of TCNT.
Figure 27.12 and Figure 27.13 show the timing of this operation.

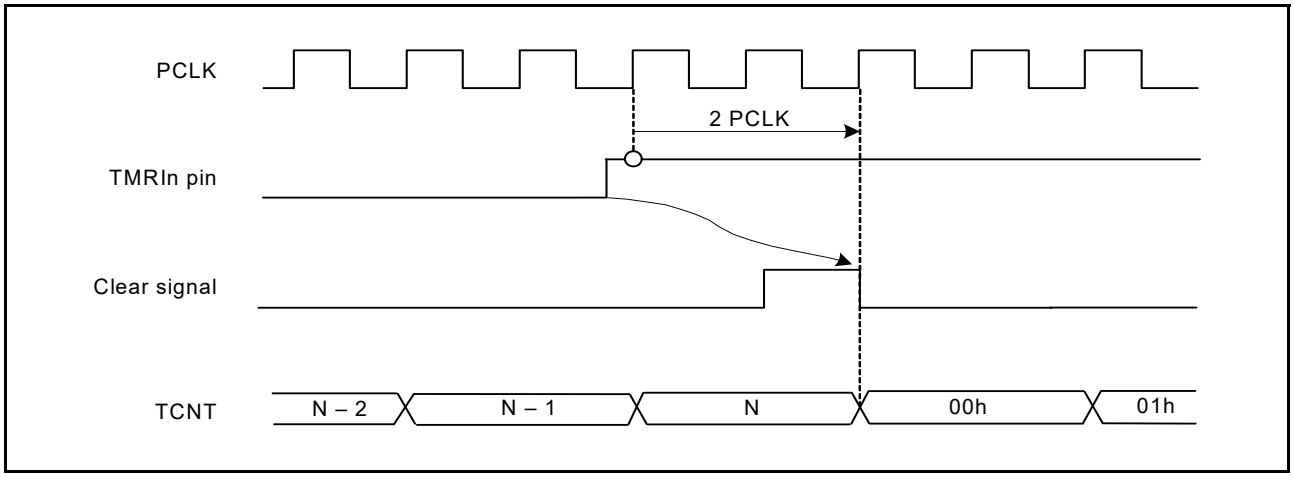


Figure 27.12 Clear Timing by External Counter Reset Signal (Rising Edge)

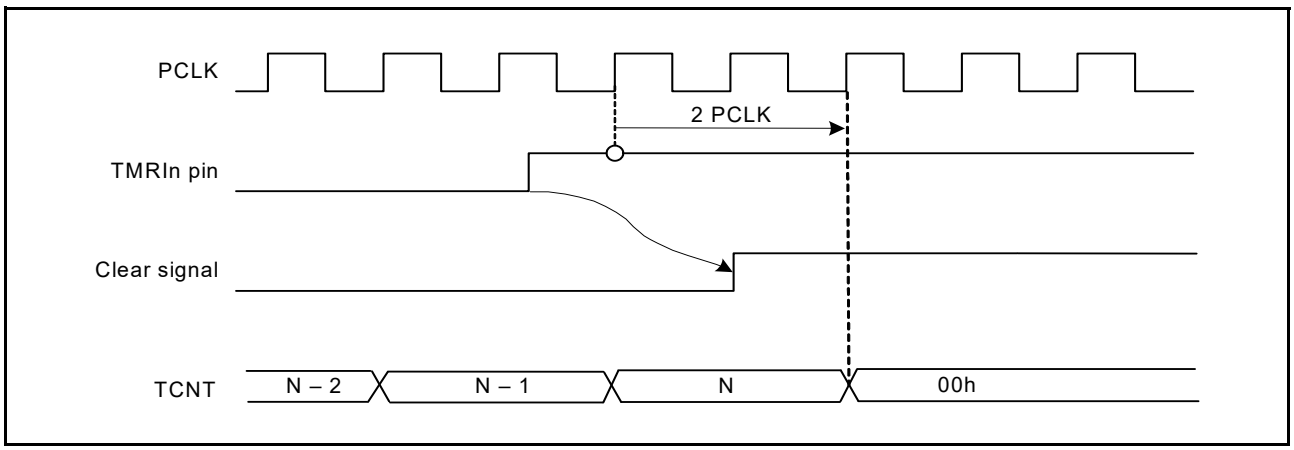


Figure 27.13 Clear Timing by External Counter Reset Signal (High Level)

27.4.6 Timing of Interrupt Signal Output on an Overflow

When TCNT overflows (changes from FFh to 00h), an overflow interrupt signal is output if this interrupt request is enabled.

Figure 27.14 shows the timing of output of the interrupt signal.

For the corresponding interrupt vector number, refer to section 14, Interrupt Controller (ICUG) and Table 27.7.

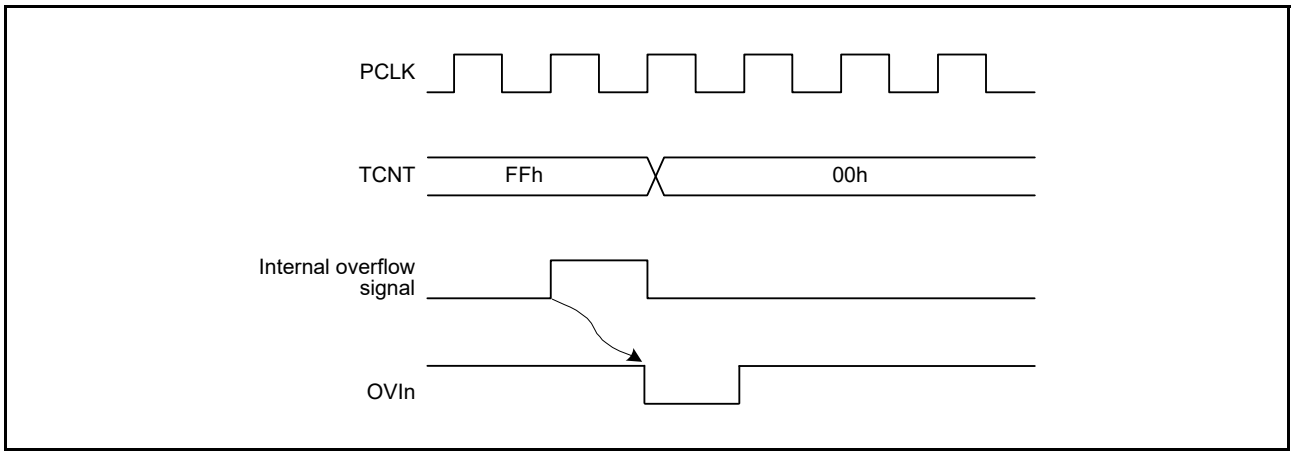


Figure 27.14 Timing of Overflow Interrupt Flag Setting to 1 (n = 0 to 7)

27.5 Operation with Cascaded Connection

If the CSS[1:0] bits in either TMR0.TCCR or TMR1.TCCR are set to 11b, the TMR of the two channels are cascaded. With this configuration, a single 16-bit timer could be used (16-bit counter mode) or compare matches of TMR0 could be counted by TMR1 (compare match count mode).
This section describes unit 0. The operation of unit 1, unit 2, and unit 3 with cascaded connection is the same as unit 0.

27.5.1 16-Bit Count Mode

When the TMR0.TCCR.CSS[1:0] bits are set to 11b, the timer functions as a single 16-bit timer with TMR0 occupying the upper 8 bits and TMR1 occupying the lower 8 bits.

(1) Counter Clear Specification

- The settings of the TMR0.TCR.CCLR[1:0] bits become effective for the 16-bit counter. If the TMR0.TCR.CCLR[1:0] bits have been set for counter clear at compare match, the 16-bit counter (TMR0.TCNT and TMR1.TCNT together) is cleared when a 16-bit compare match event occurs. The 16-bit counter (TMR0.TCNT and TMR1.TCNT together) is cleared even if counter clear by the TMRI0 pin has been set.
- The settings of the TMR1.TCR.CCLR[1:0] bits are ignored.

(2) Pin Output

- Control of output from the TMO0 pin by the TMR0.TCSR.OSA[1:0] and OSB[1:0] bits is in accordance with the 16-bit compare match conditions.
- Control of output from the TMO1 pin by the TMR1.TCSR.OSA[1:0] and OSB[1:0] bits is in accordance with the lower 8-bit compare match conditions.

27.5.2 Compare Match Count Mode

When the TMR1.TCCR.CSS[1:0] bits are set to 11b, TMR1.TCNT counts the number of occurrences of compare match A for TMR0. TMR0 and TMR1 are controlled independently. Conditions such as generation of interrupts, output from the TMOn pin (n = 0, 1), and counter clear are in accordance with the settings for each channel.

27.6 Interrupt Sources

27.6.1 Interrupt Sources and DTC Activation

There are three interrupt sources for TMRn: CMIAn, CMIBn, and OVIn. Their interrupt sources and priorities are listed in Table 27.7.
It is also possible to activate the DTC by means of CMIAn and CMIBn interrupts.

Table 27.7 TMR Interrupt Sources

Name	Interrupt Sources	DTC Activation
CMIA0	TMR0.TCORA compare match	Possible
CMIB0	TMR0.TCORB compare match	Possible
OVI0	TMR0.TCNT overflow	Not possible
CMIA1	TMR1.TCORA compare match	Possible
CMIB1	TMR1.TCORB compare match	Possible
OVI1	TMR1.TCNT overflow	Not possible
CMIA2	TMR2.TCORA compare match	Possible
CMIB2	TMR2.TCORB compare match	Possible
OVI2	TMR2.TCNT overflow	Not possible
CMIA3	TMR3.TCORA compare match	Possible
CMIB3	TMR3.TCORB compare match	Possible
OVI3	TMR3.TCNT overflow	Not possible
CMIA4	TMR4.TCORA compare match	Possible
CMIB4	TMR4.TCORB compare match	Possible
OVI4	TMR4.TCNT overflow	Not possible
CMIA5	TMR5.TCORA compare match	Possible
CMIB5	TMR5.TCORB compare match	Possible
OVI5	TMR5.TCNT overflow	Not possible
CMIA6	TMR6.TCORA compare match	Possible
CMIB6	TMR6.TCORB compare match	Possible
OVI6	TMR6.TCNT overflow	Not possible
CMIA7	TMR7.TCORA compare match	Possible
CMIB7	TMR7.TCORB compare match	Possible
OVI7	TMR7.TCNT overflow	Not possible

27.6.2 Startup of the A/D Converter

The compare match A of TMR0, TMR2, TMR4, and TMR6 allows the A/D converter to be started.
An A/D conversion start request is issued to the A/D converter in response to a generation of compare match A when the TMRn.TCSR.ADTE bit is 1 (i.e., when an A/D conversion request in response to compare match A is enabled). In this case, the conversion trigger for the 8-bit timer should be selected in the A/D converter to start A/D conversion.

Table 27.8 Startup of A/D Converter

A/D Converter	TMR Unit No.	Target	A/D Conversion Start Request
S12AD, S12AD1, S12AD2 (12-bit A/D converter)	0	Compare match between TMR0.TCORA and TMR0.TCNT	TMTRG0AN_0
	1	Compare match between TMR2.TCORA and TMR2.TCNT	TMTRG0AN_1
	2	Compare match between TMR4.TCORA and TMR4.TCNT	TMTRG0AN_2
	3	Compare match between TMR6.TCORA and TMR6.TCNT	TMTRG0AN_3

27.7 Link Operation by ELC

27.7.1 Event Signal Output to ELC

The TMR uses the event link controller (ELC) to perform link operation to the previously specified module using the interrupt request signal as the event signal. The TMR outputs compare match A, compare match B, and overflow signals as event signals. Channels that can be used in this way are TMR0 to TMR3.
The event signal can be output regardless of the setting of the corresponding interrupt request enable bits (TMRn.TCR.OVIE, TMRn.TCR.CMIEA, and TMRn.TCR.CMIEB (n = 0 to 3)). For details, refer to section 19, Event Link Controller (ELC).
The event output function can be used for the cascaded operation.

27.7.2 TMR Operation when Receiving an Event Signal from ELC

The TMR can perform either of the following operations upon the event previously specified by the ELSRn register of the ELC. However, the ELC does not support the cascaded operation.

(1) Count Start

When the TMR count start operation is selected by the ELOPD register of the ELC and the event specified by ELSRn occurs, the TCSTR.TCS bit is set to 1, starting the TMR count operation. After the TMR count start operation is selected by the ELOPD register of the ELC, use the TCCR.CKS[2:0] and CSS[1:0] bits to select the count source.
If the specified event occurs while the TCS bit is 1, the event is ignored.
Write 0 to the TCSTR.TCS bit to stop counting.
When the count start event is input in the count stopped state, the TMR starts counting again according to the CKS[2:0] and CSS[1:0] bits.
The TCS bit is valid only when the ELOPD.TMR0MD[1:0], ELOPD.TMR1MD[1:0], ELOPD.TMR2MD[1:0], and ELOPD.TMR3MD[1:0] bits of the ELC select the count start operation.

(2) Event Count

When the TMR event count operation is selected by the ELOPD register of the ELC and the event specified by ELSRn occurs, the events are counted as the count source regardless of the TCCR.CKS[2:0] and CSS[1:0] bit settings. Reading the counter value returns the number of events that have been actually input.

(3) Count Restart

When the TMR count restart operation is selected by the ELOPD register of the ELC and the event specified by ELSRn occurs, the TCNT counter value is modified to the initial value. If the CKS[2:0] and CSS[1:0] bit settings are not disabling the clock input, the count operation is continued.

27.7.3 Notes on Operating TMR According to an Event Signal from ELC

The following describes the notes on operating the TMR using the event link feature.

(1) Count Start

When the event specified by ELSRn occurs during the write cycle to the TCSTR.TCS bit, the cycle is not completed; setting 1 according to the event occurrence takes priority.

(2) Event Count

When the event specified by ELSRn occurs during the write cycle to the TCNT, the cycle is not completed; event count operation according to the event occurrence takes priority.

(3) Count Restart

When the event specified by ELSRn occurs during the write cycle to the TCNT, the cycle is not completed; count value initialization according to the event occurrence takes priority.

27.8 Usage Notes

27.8.1 Module Stop State Setting

Operation of the TMR can be disabled or enabled by using the module stop control registers. The initial setting is for halting of TMR operation. Register access becomes possible after release from the module stop state. For details, refer to section 11, Low Power Consumption.

27.8.2 Notes on Setting Cycle

If the compare match is selected for counter clear, TCNT is cleared at the last PCLK in the cycle in which the value of TCNT matches with that of TCORA or TCORB. TCNT updates the counter value at this last state. Therefore, the counter frequency is obtained by the following formula (f: Counter frequency, PCLK: Operating frequency, N: TCORA and TCORB register setting value).

$$f = PCLK / (N + 1)$$

27.8.3 Conflict between TCNT Write and Counter Clear

If a counter clear signal is generated concurrently with CPU write to TCNT, the clear takes priority and the write is not performed as shown in Figure 27.15.

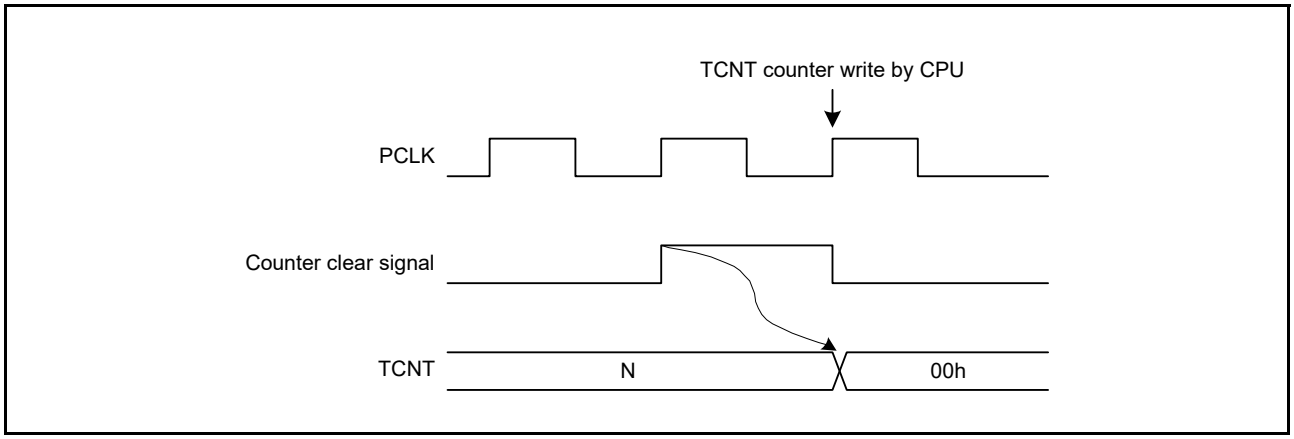


Figure 27.15 Conflict between TCNT Write and Counter Clear

27.8.4 Conflict between TCNT Write and Increment

Even if a counting-up signal is generated concurrently with CPU write to TCNT, the counting-up is not performed and the write takes priority as shown in Figure 27.16.

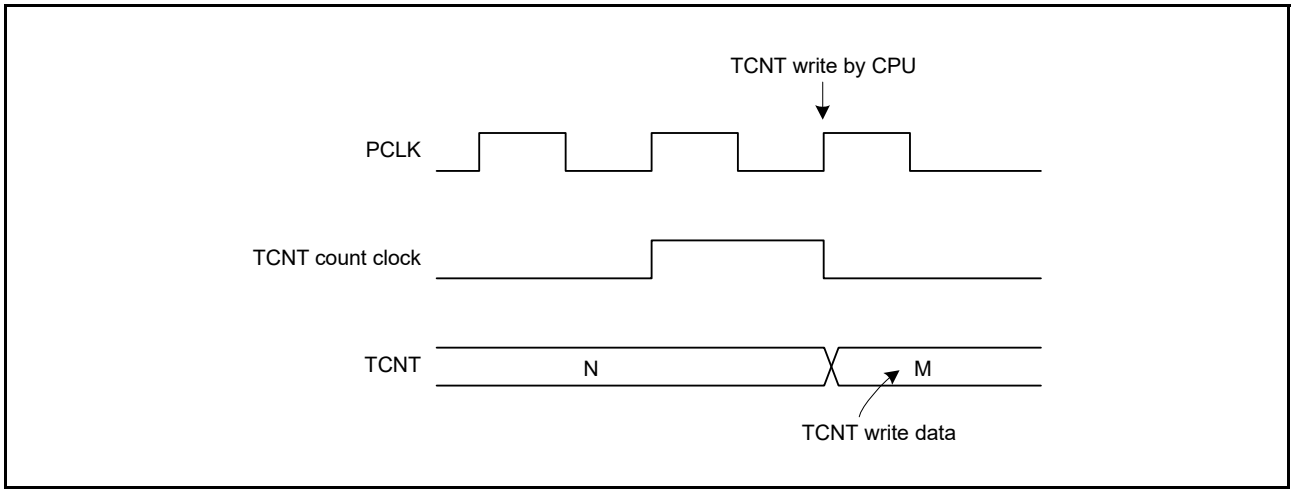


Figure 27.16 Conflict between TCNT Write and Increment

27.8.5 Conflict between TCORA or TCORB Write and Compare Match

Even if a compare match signal is generated simultaneously with CPU write to TCORA or TCORB as shown in Figure 27.17, the write takes priority and the compare match signal does not reach High level.

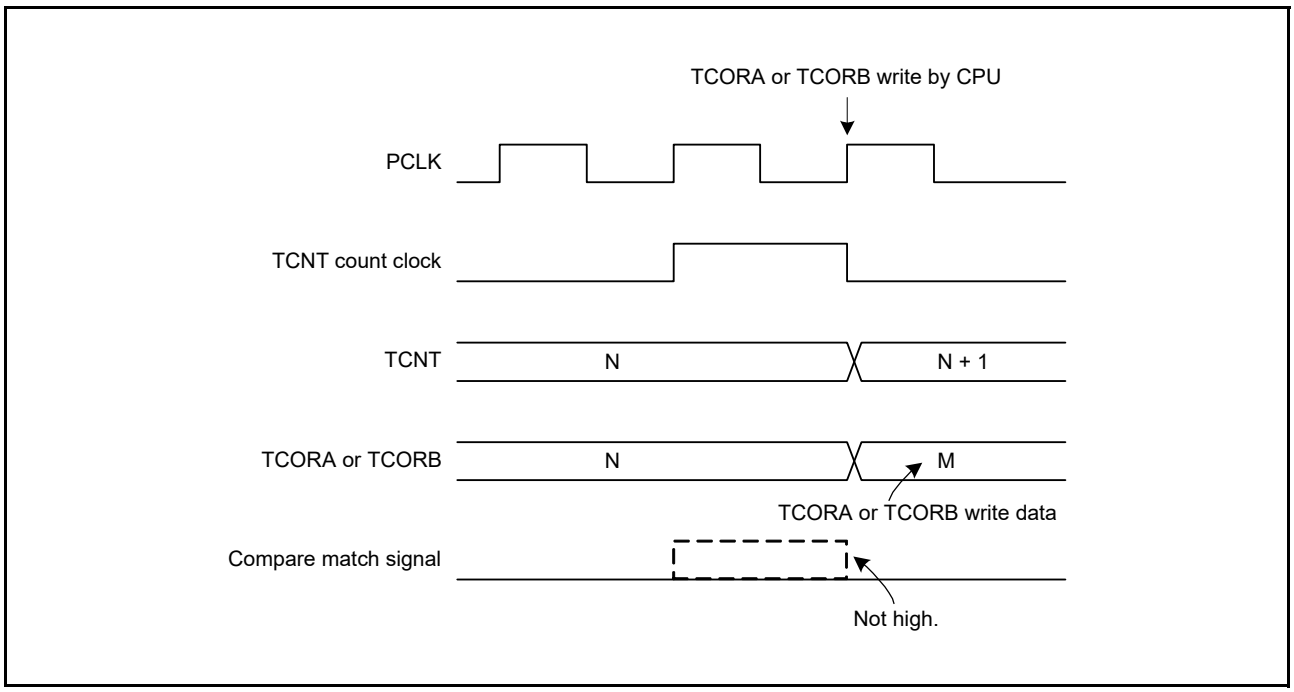


Figure 27.17 Conflict between TCORA or TCORB Write and Compare Match

27.8.6 Conflict between Compare Matches A and B

If compare match events A and B occur at the same time, the 8-bit timer operates in accordance with the priorities for the output methods high for compare match A and compare match B, as listed in Table 27.9.

Table 27.9 Timer Output Priorities

Output Setting	Priority
Toggle output	High
High output	↑
Low output	
No change	Low

27.8.7 Switching of Internal Clocks and TCNT Operation

TCNT may be incremented erroneously depending on when the internal clock is switched. Table 27.10 lists the relationship between the timing at which the internal clock is switched (by writing to the TCCR.CKS[2:0] bits) and the operation of TCNT.

When TCNT count clock is generated from an internal clock, the rising edge of the internal clock pulse are always monitored. If the signal levels of the clocks before and after switching change from low to high as shown in No. 2 in Table 27.10, the change is considered as an edge. Therefore, a TCNT count clock is generated and TCNT is incremented.

The erroneous increment of TCNT can also happen when switching between internal and internal clocks.

Table 27.10 Switching of Internal Clocks and TCNT Operation (1/2)

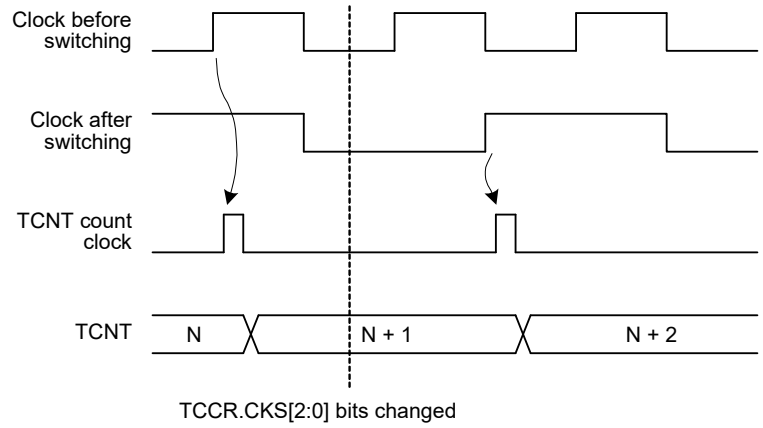
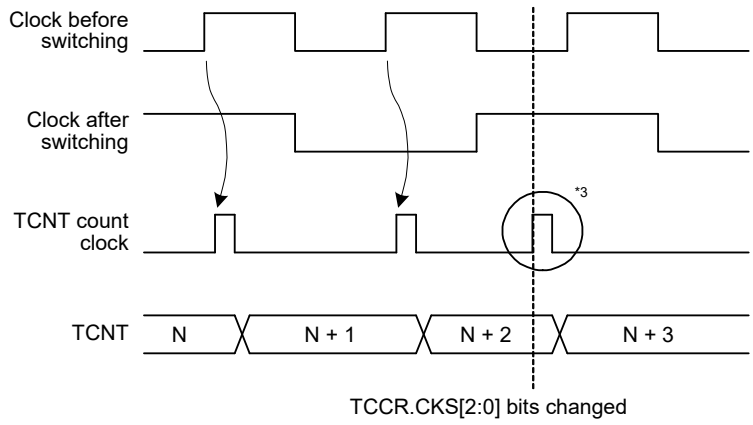
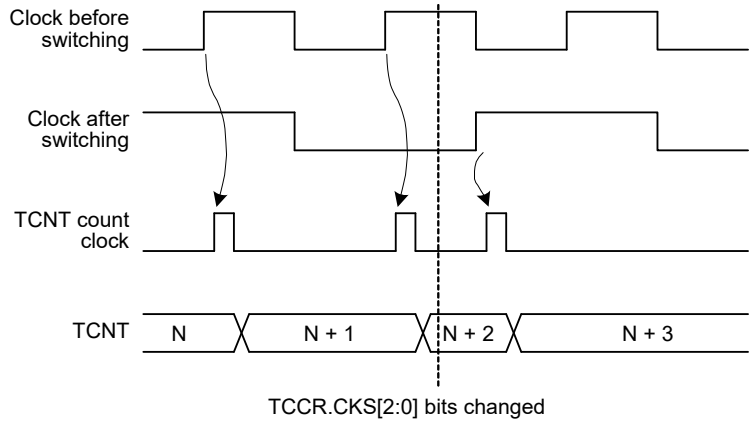
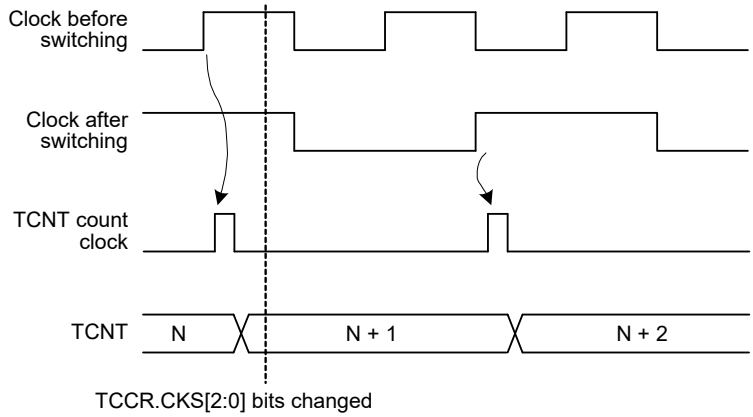
No.	Timing to Change the TCCR.CKS[2:0] Bits	TCNT Counter Operation
1	Switching from low to low*1	

Table 27.10 Switching of Internal Clocks and TCNT Operation (2/2)

No.	Timing to Change the TCCR.CKS[2:0] Bits	TCNT Counter Operation
2	Switching from low to high*2	
3	Switching from high to low*4	
4	Switching from high to high	

Note 1. Includes switching from low to stop, and from stop to low.
Note 2. Includes switching from stop to high.
Note 3. Generated because the change of the signal levels is considered as an edge; TCNT counter is incremented.
Note 4. Includes switching from high to stop.

27.8.8 Clock Source Setting with Cascaded Connection

If 16-bit counter mode and compare match count mode are specified at the same time, count clocks for TMR0.TCNT and TMR1.TCNT (TMR2.TCNT and TMR3.TCNT, TMR4.TCNT and TMR5.TCNT, TMR6.TCNT and TMR7.TCNT) are not generated, and the counter stops. Do not specify 16-bit counter mode and compare match count mode simultaneously.

27.8.9 Continuous Output of Compare Match Interrupt Signal

When TCORA or TCORB is set to 00h, PCLK/1 is set as the internal clock, and compare match is set as the counter clear source, the TCNT counter remains 00h and is not updated, and a compare match interrupt signal is output continuously to form a flat signal level.
At this time, the interrupt controller cannot detect the second and subsequent interrupts.
Figure 27.18 shows operation timing when the compare match interrupt signal is continuously output.

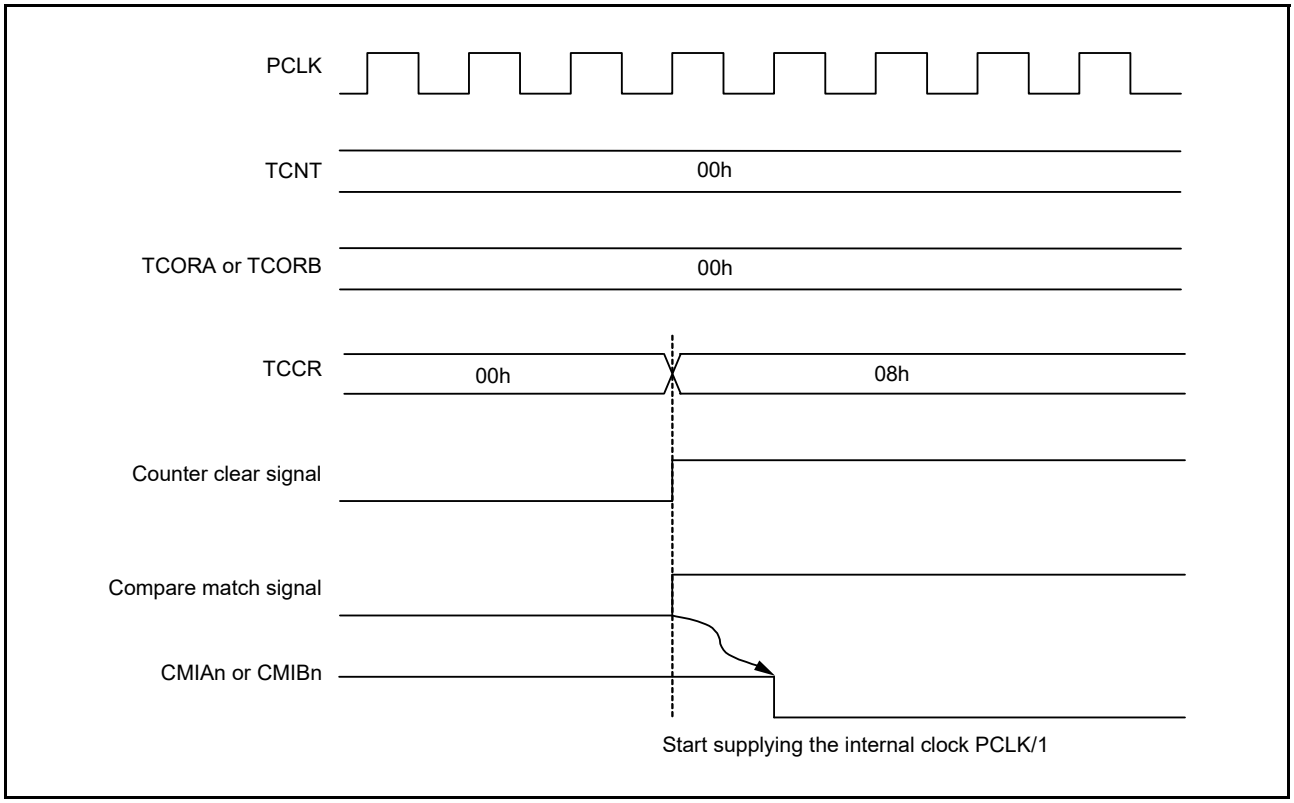


Figure 27.18 Continuous Output of Compare Match Interrupt Signal (n = 0 to 7)

28. Compare Match Timer (CMT)

This MCU has two on-chip compare match timer (CMT) units (unit 0 and unit 1), each consisting of a two-channel 16-bit timer (i.e., a total of four channels). The CMT has a 16-bit counter, and can generate interrupts at set intervals.
In this section, “PCLK” is used to refer to PCLKB.

28.1 Overview

Table 28.1 lists the specifications for the CMT.
Figure 28.1 shows a block diagram of the CMT (unit 0). A two-channel CMT constitutes a unit. Unit 0 and unit 1 are the same in terms of specifications. Compare match timer start register 0 (CMSTR0) and compare match interrupts (CMI0 and CMI1) of unit 0 correspond to compare match timer start register 1 (CMSTR1) and compare match interrupts (CMI2 and CMI3) of unit 1.

Table 28.1 CMT Specifications

Item	Description
Count clocks	• Four frequency dividing clocks One clock from PCLK/8, PCLK/32, PCLK/128, and PCLK/512 can be selected for each channel.
Interrupt	A compare match interrupt can be requested for each channel.
Event link function (output)	An event signal is output upon a CMT1 compare match.
Event link function (input)	Linking to the specified module is possible. CMT1 count start, event counter, or count restart operation is possible.
Low power consumption function	Each unit can be placed in a module stop state.

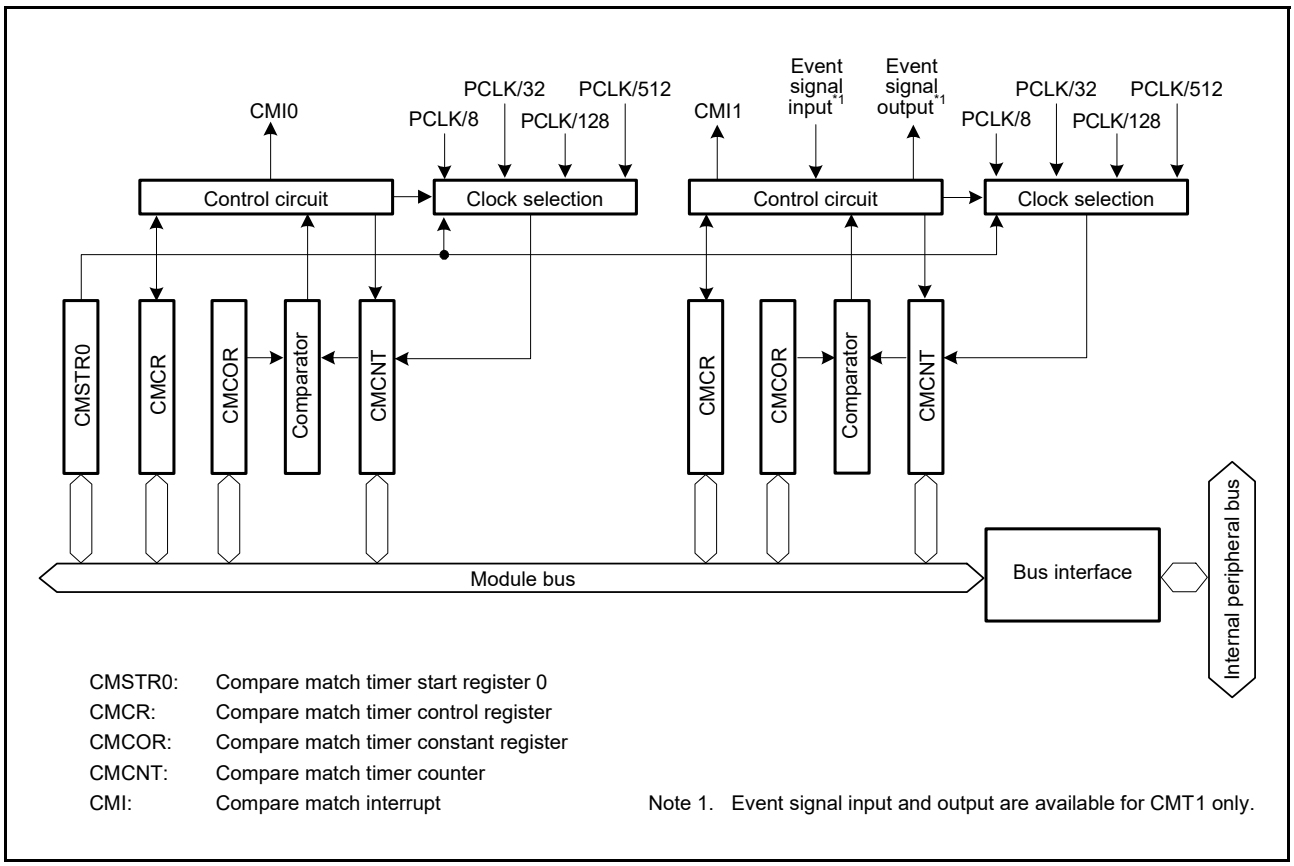


Figure 28.1 CMT (Unit 0) Block Diagram

28.2 Register Descriptions

28.2.1 Compare Match Timer Start Register 0 (CMSTR0)

Address(es): 0008 8000h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STR1	STR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	STR0	Count Start 0	0: CMT0.CMCNT count is stopped. 1: CMT0.CMCNT count is started.	R/W
b1	STR1	Count Start 1	0: CMT1.CMCNT count is stopped. 1: CMT1.CMCNT count is started.	R/W
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

28.2.2 Compare Match Timer Start Register 1 (CMSTR1)

Address(es): 0008 8010h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STR3	STR2
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	STR2	Count Start 2	0: CMT2.CMCNT count is stopped. 1: CMT2.CMCNT count is started.	R/W
b1	STR3	Count Start 3	0: CMT3.CMCNT count is stopped. 1: CMT3.CMCNT count is started.	R/W
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

28.2.3 Compare Match Timer Control Register (CMCR)

Address(es): CMT0.CMCR 0008 8002h, CMT1.CMCR 0008 8008h, CMT2.CMCR 0008 8012h, CMT3.CMCR 0008 8018h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	CMIE	—	—	—	—	CKS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	x	0	0	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CKS[1:0]	Clock Select	b1 b0 0 0: PCLK/8 0 1: PCLK/32 1 0: PCLK/128 1 1: PCLK/512	R/W
b5 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	CMIE	Compare Match Interrupt Enable	0: Compare match interrupt (CMIn) disabled 1: Compare match interrupt (CMIn) enabled	R/W
b7	—	Reserved	This bit is read as undefined. The write value should be 1.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CKS[1:0] Bits (Clock Select)

These bits select the count source from four frequency dividing clocks obtained by dividing the peripheral module clock (PCLK).

When the CMSTRm.STRn (m = 0, 1; n = 0 to 3) bit is set to 1, the CMCNT counter starts counting up on the clock selected with the CKS[1:0] bits.

CMIE Bit (Compare Match Interrupt Enable)

The CMIE bit enables or disables compare match interrupt (CMIn) (n = 0 to 3) generation when the CMCNT counter and the CMCOR register values match.

28.2.4 Compare Match Counter (CMCNT)

Address(es): CMT0.CMCNT 0008 8004h, CMT1.CMCNT 0008 800Ah, CMT2.CMCNT 0008 8014h, CMT3.CMCNT 0008 801Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The CMCNT counter is a readable/writable up-counter.
When an frequency dividing clock is selected by the CMCR.CKS[1:0] bits and the CMSTRm.STRn (m = 0, 1; n = 0 to 3) bit is set to 1, the CMCNT counter starts counting up using the selected clock.
When the value in the CMCNT counter and the value in the CMCOR register match, the CMCNT counter is set to 0000h. At the same time, a compare match interrupt (CMIn) (n = 0 to 3) is generated.

28.2.5 Compare Match Constant Register (CMCOR)

Address(es): CMT0.CMCOR 0008 8006h, CMT1.CMCOR 0008 800Ch, CMT2.CMCOR 0008 8016h, CMT3.CMCOR 0008 801Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The CMCOR register is a readable/writable register to set a value for compare match with the CMCNT counter.

28.3 Operation

28.3.1 Periodic Count Operation

When an frequency dividing clock is selected by the CMCR.CKS[1:0] bits and the CMSTRm.STRn (m = 0, 1; n = 0 to 3) bit is set to 1, the CMCNT counter starts counting up using the selected clock.
When the value in the counter and the value in the register match, a compare match interrupt (CMIn) (n = 0 to 3) is generated. The CMCNT counter then starts counting up again from 0000h. Figure 28.2 shows the operation of the CMCNT counter.

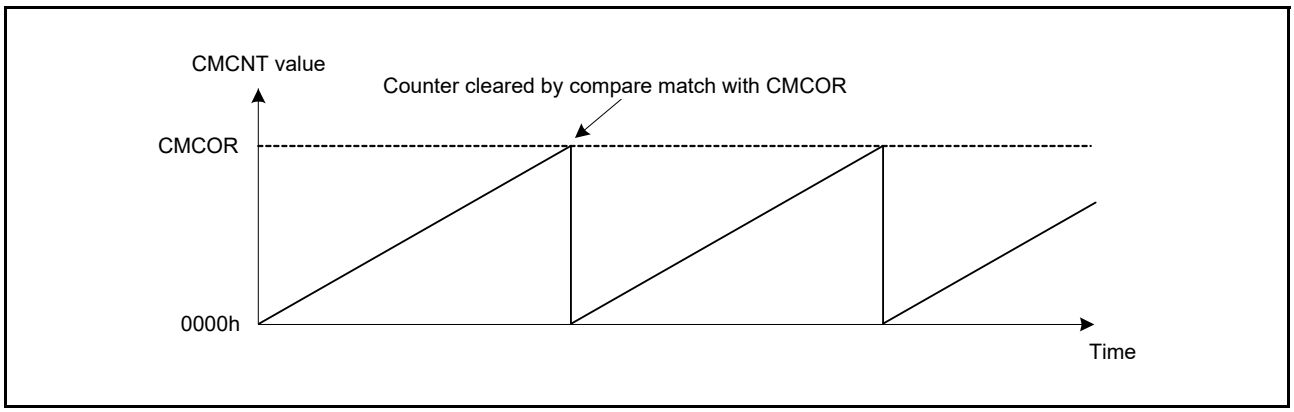


Figure 28.2 CMCNT Counter Operation

28.3.2 CMCNT Count Timing

As the count clock to be input to the CMCNT counter, one of four frequency dividing clocks (PCLK/8, PCLK/32, PCLK/128, and PCLK/512) obtained by dividing the peripheral module clock (PCLK) can be selected with the CMCR.CKS[1:0] bits. Figure 28.3 shows the timing of the CMCNT counter.

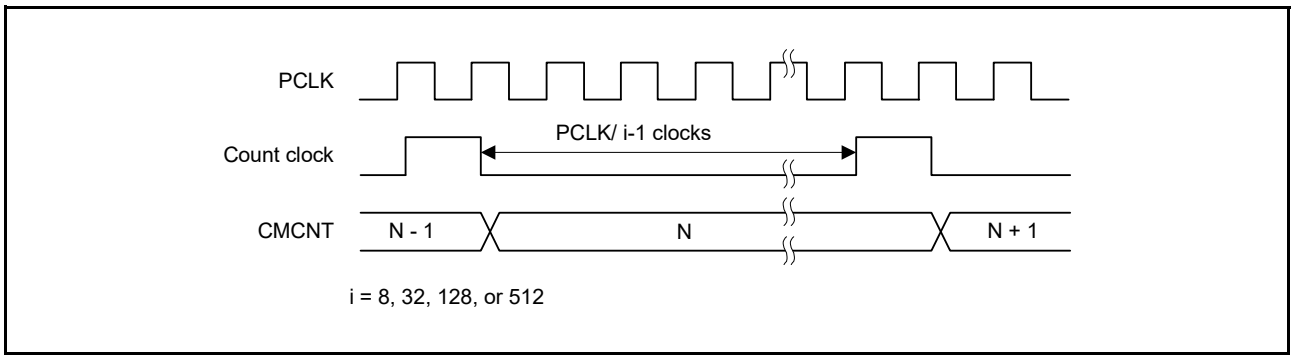


Figure 28.3 CMCNT Count Timing

28.4 Interrupts

28.4.1 Interrupt Sources

The CMT has channels and each of them to which a different vector address is allocated has a compare match interrupt (CMIn) (n = 0 to 3). When a compare match interrupt occurs, the corresponding interrupt request is output.
When the interrupt request is used to generate a CPU interrupt, the priority of channels can be changed by the interrupt controller settings. For details, see section 14, Interrupt Controller (ICUG).

Table 28.2 CMT Interrupt Sources

Name	Interrupt Sources	DTC Activation	DMAC Activation
CMI0	Compare match in CMT0	Possible	Possible
CMI1	Compare match in CMT1	Possible	Possible
CMI2	Compare match in CMT2	Possible	Possible
CMI3	Compare match in CMT3	Possible	Possible

28.4.2 Timing of Compare Match Interrupt Generation

When the CMCNT counter and the CMCOR register match, a compare match interrupt (CMIn) (n = 0 to 3) is generated. A compare match signal is generated at the last state in which the values match (the timing when the CMCNT counter updates the matched count value). That is, after a match between the CMCOR register and the CMCNT counter, the compare match signal is not generated until the next the CMCNT counter input clock.
Figure 28.4 shows the timing of a compare match interrupt.

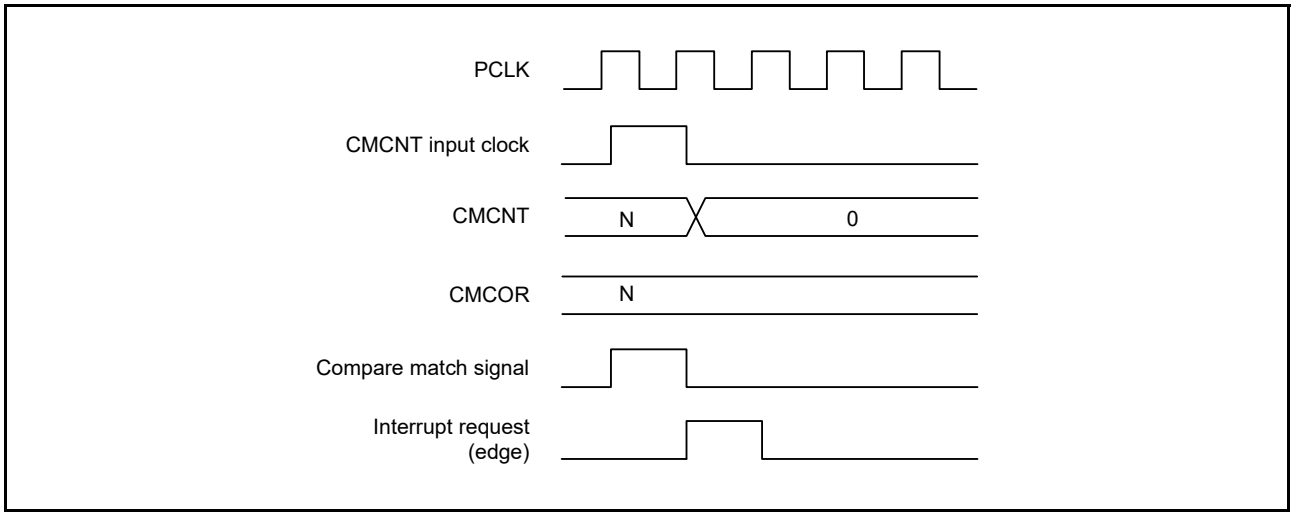


Figure 28.4 Timing of a Compare Match Interrupt

28.5 Link Operations by ELC

28.5.1 Event Signal Output to ELC

The CMT uses the event link controller (ELC) to perform link operation to a preset module using the interrupt request signal as the event signal. The CMT outputs the event signal upon a CMT1 compare match.
The event signal can be output regardless of the setting of the corresponding interrupt request enable bit (CMTn.CMCR.CMIE).

28.5.2 CMT Operation When Receiving an Event Signal from ELC

The CMT can perform either of the following operations upon the event preset by the ELSR7 register of the ELC.

(1) Count Start

When the CMT count start operation is selected by the ELOPC register of the ELC and the event specified by ELSR7 occurs, the CMSTR0.STR1 bit is set to 1, starting the CMT count operation.
However, if the specified event occurs while the CMSTR0.STR1 bit is 1, the event is ignored.

(2) Event Count

When the CMT event count operation is selected by the ELOPC register of the ELC and the event specified by ELSR7 occurs with the CMSTR0.STR1 bit being 1, the events are counted as the count source regardless of the CMT1.CMCR.CKS[1:0] bit setting. Reading the counter value returns the number of events that have been actually input.

(3) Count Restart

When the CMT count restart operation is selected by the ELOPC register of the ELC and the event specified by ELSR7 occurs, the CMT1.CMCNT counter value is modified to the initial value. If the CMSTR0.STR1 bit is 1 here, the count operation can be continued.

28.5.3 Notes on Operating CMT According to an Event Signal from ELC

The following describes the notes on operating the CMT using the event link feature.

(1) Count Start

When the event specified by ELSR7 occurs during the write cycle to the CMSTR0.STR1 bit, the cycle is not completed; setting 1 according to the event occurrence takes priority.

(2) Event Count

When the event specified by ELSR7 occurs during the write cycle to the CMT1.CMCNT, the cycle is not completed; event count operation according to the event occurrence takes priority.

(3) Count Restart

When the event specified by ELSR7 occurs during the write cycle to the CMT1.CMCNT, the cycle is not completed; count value initialization according to the event occurrence takes priority.

28.6 Usage Notes

28.6.1 Setting the Module Stop Function

The CMT can be enabled or disabled using the module stop control register. After a reset, the CMT is in the module stop state. The registers can be accessed by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

28.6.2 Conflict between CMCNT Counter Writing and Compare Match

When the compare match signal is generated while writing to the CMCNT counter, clearing the CMCNT counter has priority over writing to it. In this case, the CMCNT counter is not written to. Figure 28.5 shows the timing to clear the CMCNT counter.

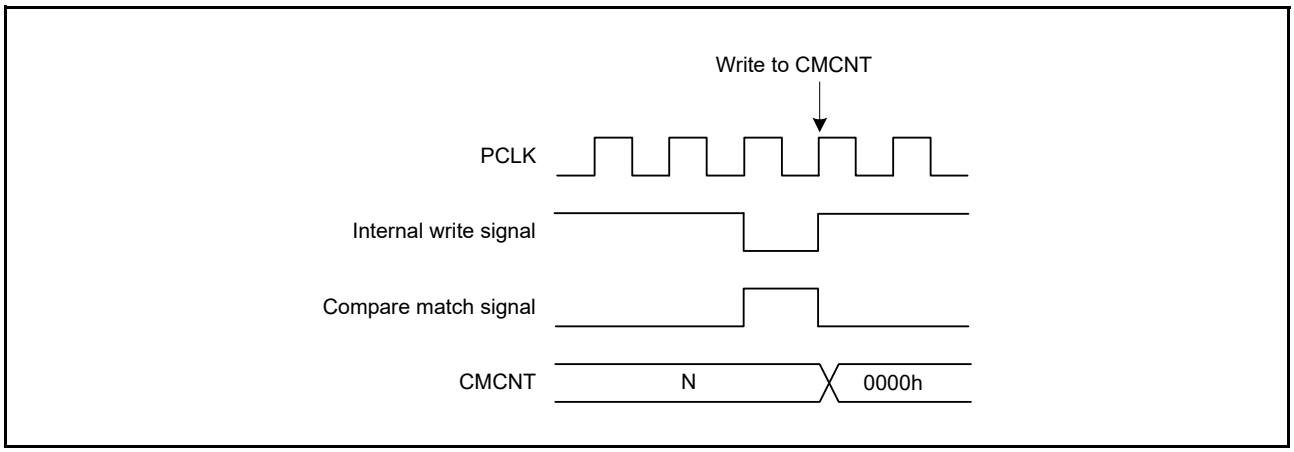


Figure 28.5 Conflict between CMCNT Counter Writing and Compare Match

28.6.3 Conflict between CMCNT Counter Writing and Incrementing

If writing to the counter and the incrementing conflict, the writing has priority over the incrementing. Figure 28.6 shows the timing to write the CMCNT counter.

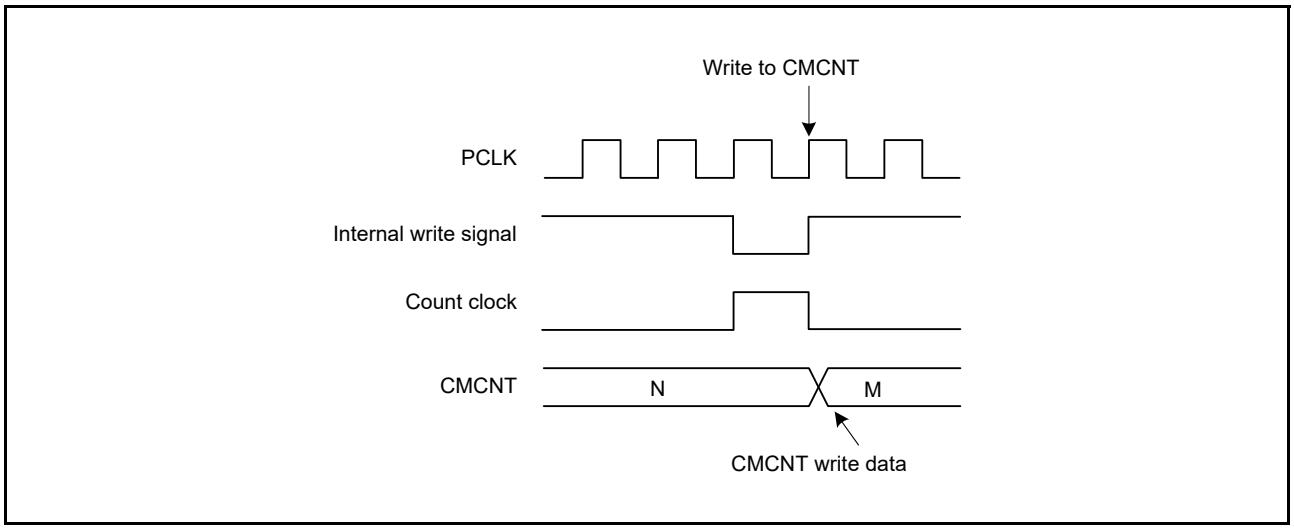


Figure 28.6 Conflict between CMCNT Counter Writing and Incrementing

29. Compare Match Timer W (CMTW)

This MCU includes two units (unit 0 and unit 1) with one channel of 32-bit compare match timer W (CMTW).
CMTW has a 32-bit counter and can generate interrupts each time a set period elapses.
In this section, “PCLK” is used to refer to PCLKB.

29.1 Overview

Table 29.1 shows the specifications of the CMTW.
Figure 29.1 shows a block diagram of the CMTW0 and Figure 29.2 shows a block diagram of the CMTW1.

Table 29.1 CMTW Specifications

Item	Function
Number of channels	Two channels (unit 0, unit 1)
Timer counter	16-bit/32-bit selectable up-counter The counter returns to 0000 0000h after a compare match.
Prescaler	Four dividing clocks are output. Selectable from any of PCLK/8, PCLK/32, PCLK/128, and PCLK/512
Input capture	Up to two input capture input signals available.
Output compare	Up to two output compare output signals available.
Compare match	One compare match available (no output compare output pin used).
Interrupts	Compare match interrupt Input capture 0 and 1 interrupts Output compare 0 and 1 interrupts
Event link function (output) (Unit 0)	Compare match
Event link function (input) (Unit 0)	One of the following operations is enabled after an event is accepted: • Count start operation • Event count operation • Count restart operation
Low power consumption function	Each unit can be placed in the module stop state.

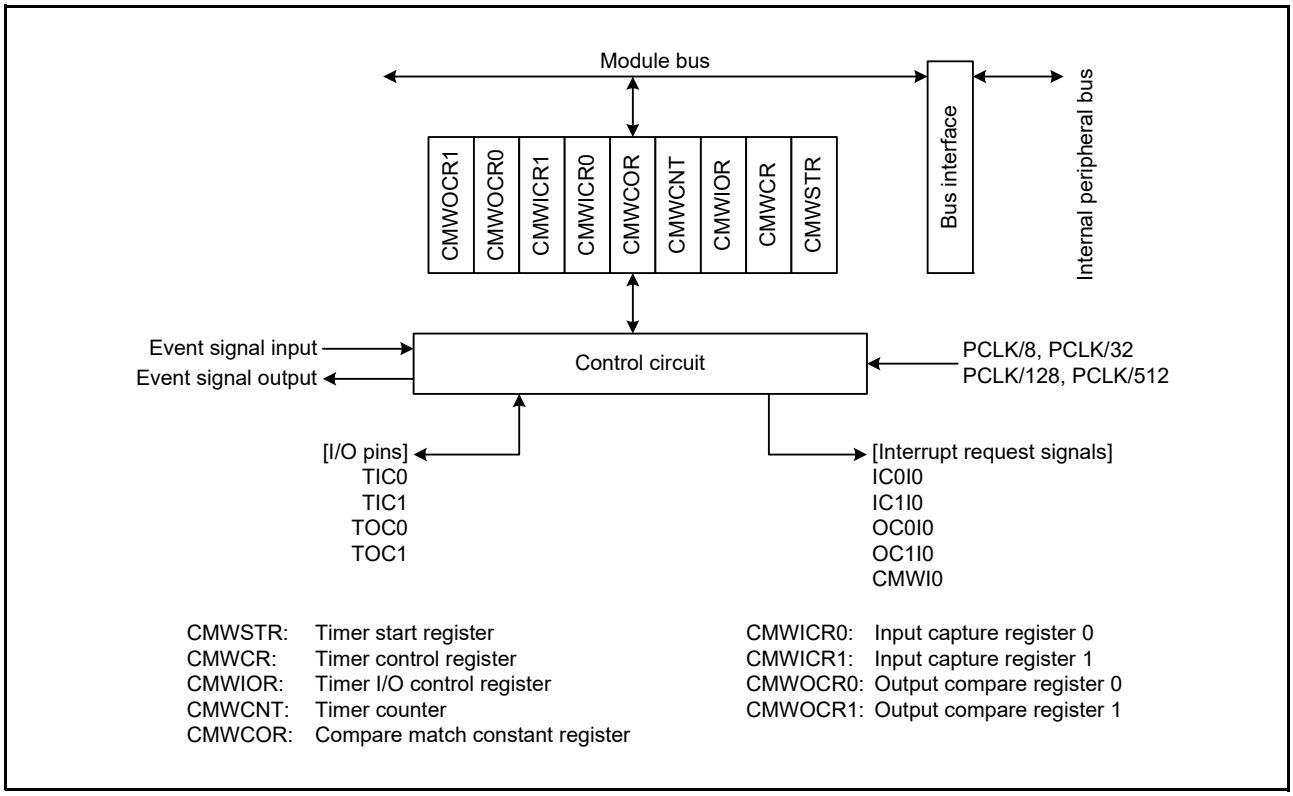


Figure 29.1 CMTW0 Block Diagram

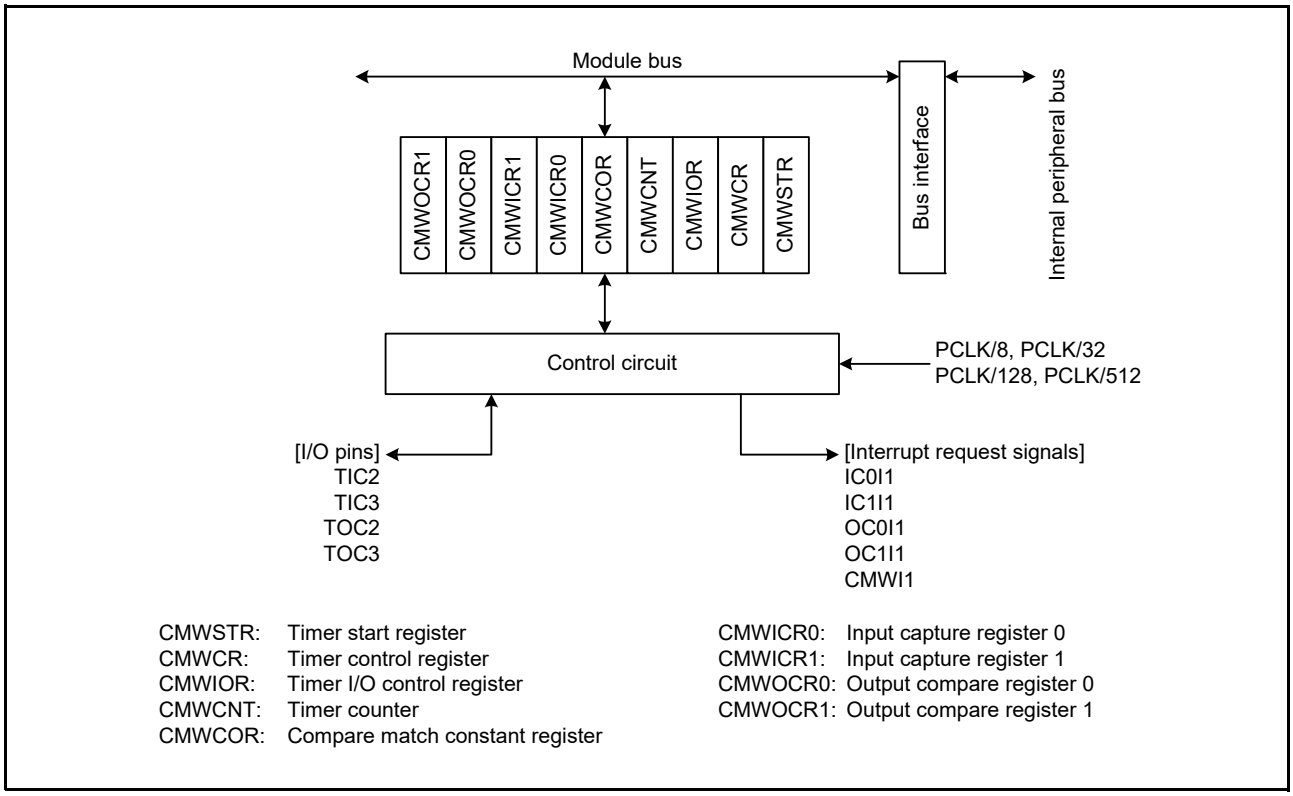


Figure 29.2 CMTW1 Block Diagram

Table 29.2 shows the CMTW pin configuration.

Table 29.2 CMTW Pin Configuration

Unit	Pin Name	I/O	Description
CMTW0	TIC0	Input	Input capture input for the CMTW0.CMWICR0 register
	TIC1	Input	Input capture input for the CMTW0.CMWICR1 register
	TOC0	Output	Output compare output for the CMTW0.CMWOCR0 register
	TOC1	Output	Output compare output for the CMTW0.CMWOCR1 register
CMTW1	TIC2	Input	Input capture input for the CMTW1.CMWICR0 register
	TIC3	Input	Input capture input for the CMTW1.CMWICR1 register
	TOC2	Output	Output compare output for the CMTW1.CMWOCR0 register
	TOC3	Output	Output compare output for the CMTW1.CMWOCR1 register

29.2 Register Descriptions

29.2.1 Timer Start Register (CMWSTR)

Address(es): CMTW0.CMWSTR 0009 4200h, CMTW1.CMWSTR 0009 4280h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	STR	Counter Start	0: CMWCNT counter count is stopped. (The value immediately before count operation stops is retained and the count operation is stopped.) 1: CMWCNT counter count is started.	R/W
b15 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

STR Bit (Counter Start)

Specifies whether the timer counter operates or is stopped. The relevant prescaler operates or is stopped according to the settings of the STR bit.

29.2.2 Timer Control Register (CMWCR)

Address(es): CMTW0.CMWCR 0009 4204h, CMTW1.CMWCR 0009 4284h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CCLR[2:0]			—	—	—	CMS	—	OC1IE	OC0IE	IC1IE	IC0IE	CMWIE	—	CKS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CKS[1:0]	Clock Select	b1 b0 0 0: PCLK/8 0 1: PCLK/32 1 0: PCLK/128 1 1: PCLK/512	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	CMWIE	Compare Match Interrupt Request Enable	0: Interrupt request (CMWI) disabled 1: Interrupt request (CMWI) enabled	R/W
b4	IC0IE	Input Capture 0 Interrupt Request Enable	0: Interrupt request (IC0I) disabled 1: Interrupt request (IC0I) enabled	R/W
b5	IC1IE	Input Capture 1 Interrupt Request Enable	0: Interrupt request (IC1I) disabled 1: Interrupt request (IC1I) enabled	R/W
b6	OC0IE	Output Compare 0 Interrupt Request Enable	0: Interrupt request (OC0I) disabled 1: Interrupt request (OC0I) enabled	R/W
b7	OC1IE	Output Compare 1 Interrupt Request Enable	0: Interrupt request (OC1I) disabled 1: Interrupt request (OC1I) enabled	R/W
b8	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b9	CMS	Timer Counter Size	0: 32 bits 1: 16 bits	R/W
b12 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b13	CCLR[2:0]	Counter Clear	b15 b13 0 0 0: CMWCNT counter cleared by CMWCOR register compare match 0 0 1: Clearing of CMWCNT counter disabled 0 1 0: Clearing of CMWCNT counter disabled 0 1 1: Clearing of CMWCNT counter disabled 1 0 0: CMWCNT counter cleared by CMWICR0 register input capture 1 0 1: CMWCNT counter cleared by CMWICR1 register input capture 1 1 0: CMWCNT counter cleared by CMWOCR0 register compare match 1 1 1: CMWCNT counter cleared by CMWOCR1 register compare match	R/W

The CMWCR register should be set while the timer counter (CMWCNT) operation is stopped.

CKS[1:0] Bits (Clock Select)

Select the clock to be input to the CMWCNT counter among four internal clocks obtained by dividing the peripheral module clock (PCLK). When the CMWSTR.STR bit is set to 1, the CMWCNT counter starts counting based on the clock selected with the CMWCR.CKS[1:0] bits.

CMWIE Bit (Compare Match Interrupt Request Enable)

Enables or disables compare match interrupt request (CMWI) generation when the CMWCNT counter and the CMWCOR register values match.

IC0IE Bit (Input Capture 0 Interrupt Request Enable)

Enables or disables input capture 0 interrupt request (IC0I) generation when input capture is generated in the CMWICR0 register.

IC1IE Bit (Input Capture 1 Interrupt Request Enable)

Enables or disables input capture 1 interrupt request (IC1I) generation when input capture is generated in the CMWICR1 register.

OC0IE Bit (Output Compare 0 Interrupt Request Enable)

Enables or disables output compare 0 interrupt request (OC0I) generation when the CMWCNT counter and the CMWOCR0 register values match.

OC1IE Bit (Output Compare 1 Interrupt Request Enable)

Enables or disables output compare 1 interrupt request (OC1I) generation when the CMWCNT counter and CMWOCR1 register values match.

CMS Bit (Timer Counter Size)

Selects either 16 or 32 bits as the size of the timer counter (CMWCNT). The size selected with the CMS bit is valid in the CMWCOR, CMWICR0, CMWICR1, CMWOCR0, and CMWOCR1 registers.

CCLR[2:0] Bits (Counter Clear)

Select the CMWCNT counter clearing source.

29.2.3 Timer I/O Control Register (CMWIOR)

Address(es): CMTW0.CMWIOR 0009 4208h, CMTW1.CMWIOR 0009 4288h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CMWE	—	OC1E	OC0E	OC1[1:0]		OC0[1:0]		—	—	IC1E	IC0E	IC1[1:0]		IC0[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	IC0[1:0]	Input Capture 0 Control	b1 b0 0 0: Input capture at the rising edge 0 1: Input capture at the falling edge 1 0: Input capture at both edges 1 1: Setting prohibited	R/W
b3, b2	IC1[1:0]	Input Capture 1 Control	b3 b2 0 0: Input capture at the rising edge 0 1: Input capture at the falling edge 1 0: Input capture at both edges 1 1: Setting prohibited	R/W
b4	IC0E	Input Capture 0 Enable	0: Input capture 0 operation disabled 1: Input capture 0 operation enabled	R/W
b5	IC1E	Input Capture 1 Enable	0: Input capture 1 operation disabled 1: Input capture 1 operation enabled	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	OC0[1:0]	Output Compare 0 Control	b9 b8 0 0: Retains the output value.*1 0 1: Initially outputs low and toggles the output value upon compare match. 1 0: Initially outputs high and toggles the output value upon compare match. 1 1: Setting prohibited	R/W
b11, b10	OC1[1:0]	Output Compare 1 Control	b11 b10 0 0: Retains the output value.*1 0 1: Initially outputs low and toggles the output value upon compare match. 1 0: Initially outputs high and toggles the output value upon compare match. 1 1: Setting prohibited	R/W
b12	OC0E	Output Compare 0 Enable	0: Output compare 0 operation disabled 1: Output compare 0 operation enabled	R/W
b13	OC1E	Output Compare 1 Enable	0: Output compare 1 operation disabled 1: Output compare 1 operation enabled	R/W
b14	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b15	CMWE	Compare Match Enable	0: Compare match operation disabled 1: Compare match operation enabled	R/W

Note 1. After reset, low is output until the CMWIOR register is set.

29.2.4 Timer Counter (CMWCNT)

Address(es): CMTW0.CMWCNT 0009 4210h, CMTW1.CMWCNT 0009 4290h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The CMWCNT counter is a readable/writable up-counter.
Before starting count operation, the timer control register (CMWCR) should be set. When the 16-bit counter size is selected with the CMS bit in the timer control register (CMWCR), bits 15 to 0 in this register are valid. When writing to the CMWCNT counter, write data in 32-bit units with the upper bits set to 0000h. The CMWCNT counter can only be accessed in longword units.
When the CMWSTR.STR bit is set to 1, the CMWCNT counter starts counting. When the CMWSTR.STR bit is set to 0, the CMWCNT counter retains the value immediately before a stop of counting and stops counting.

29.2.5 Compare Match Constant Register (CMWCOR)

Address(es): CMTW0.CMWCOR 0009 4214h, CMTW1.CMWCOR 0009 4294h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The CMWCOR register is a readable/writable register that specifies the time up to a compare match between the timer counter (CMWCNT) value and CMWCOR value. When the 16-bit counter size is selected with the CMS bit in the timer control register (CMWCR), bits 15 to 0 in this register are valid. When writing to the CMWCOR register, write data in 32-bit units with the upper bits set to 0000h. The CMWCOR register can only be accessed in longword units. To detect an overflow, set the CMWCOR register value to FFFF FFFFh (32-bit count operation) or 0000 FFFFh (16-bit count operation). When the CMWCNT counter is set to 0, a compare match interrupt request (CMWI) can be used as an overflow detection signal.

29.2.6 Input Capture Register n (CMWICRn) (n = 0, 1)

Address(es): CMTW0.CMWICR0 0009 4218h, CMTW0.CMWICR1 0009 421Ch, CMTW1.CMWICR0 0009 4298h, CMTW1.CMWICR1 0009 429Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The CMWICRn register is a read-only register in which the CMWCNT value is stored when an input capture is generated.
When the 16-bit counter size is selected with the CMWCR.CMS bit, bits 15 to 0 in these registers are valid. Writing to these registers is invalid. The CMWICRn register can only be accessed in longword units.

29.2.7 Output Compare Register n (CMWOCRn) (n = 0, 1)

Address(es): CMTW0.CMWOCR0 0009 4220h, CMTW0.CMWOCR1 0009 4224h, CMTW1.CMWOCR0 0009 42A0h, CMTW1.CMWOCR1 0009 42A4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The CMWOCRn register is a readable/writable register that set the value to be compared when an output compare is generated.
When the 16-bit counter size is selected with the CMWCR.CMS bit, bits 15 to 0 of these registers become valid. When writing to these registers, write data in 32-bit units with the upper bits set to 0000h.
The CMWOCRn register can only be accessed in longword units. The initial value of CMWOCR0 and CMWOCR1 registers is FFFF FFFFh.

29.3 Operation

When the CMWCR register is set and then the STR bit in CMWSTR is set to 1, the CMTW starts count operation. Setting the CMWSTR.STR bit to 0 enables the CMWCNT counter to retain the value immediately before a stop of counting and stop counting. Setting the CMWIOR register enables using the compare match function, input capture input function, and output compare output function.

29.3.1 Period Count Operation

When the counter clock is selected by using the CMWCR.CKS[1:0] bits and the CMWSTR.STR bit is set to 1, the CMWCNT counter starts counting cycles of the selected clock. When clearing of the counter is selected by the CMWCR.CCLR[2:0] bits and the counter clearing source is generated, the CMWCNT counter becomes 0000 0000h and continues counting. When clearing of the counter is not selected, an overflow is generated when FFFF FFFFh changes to 0000 0000h during 32-bit count operation and 0000 FFFFh changes to 0000 0000h during 16-bit count operation, and the CMWCNT counter continues counting.

29.3.2 Compare Match Function

When the values of the CMWCNT counter and CMWCOR register match, the CMWCNT counter becomes 0000 0000h. At this time, a compare match interrupt request (CMWI) is generated. The CMWCNT counter restarts counting from 0000 0000h.

To enable overflow detection, the CMWCOR register value should be set to FFFF FFFFh (when the counter size is 32 bits) or 0000 FFFFh (when the counter size is 16 bits). When the values of the CMWCNT counter and CMWCOR register match, the CMWCNT counter becomes 0000 0000h. In this case, the compare match interrupt request (CMWI) is generated. The CMWCNT counter then restarts counting from 0000 0000h.

Figure 29.3 shows an example of procedure for setting compare match operation.

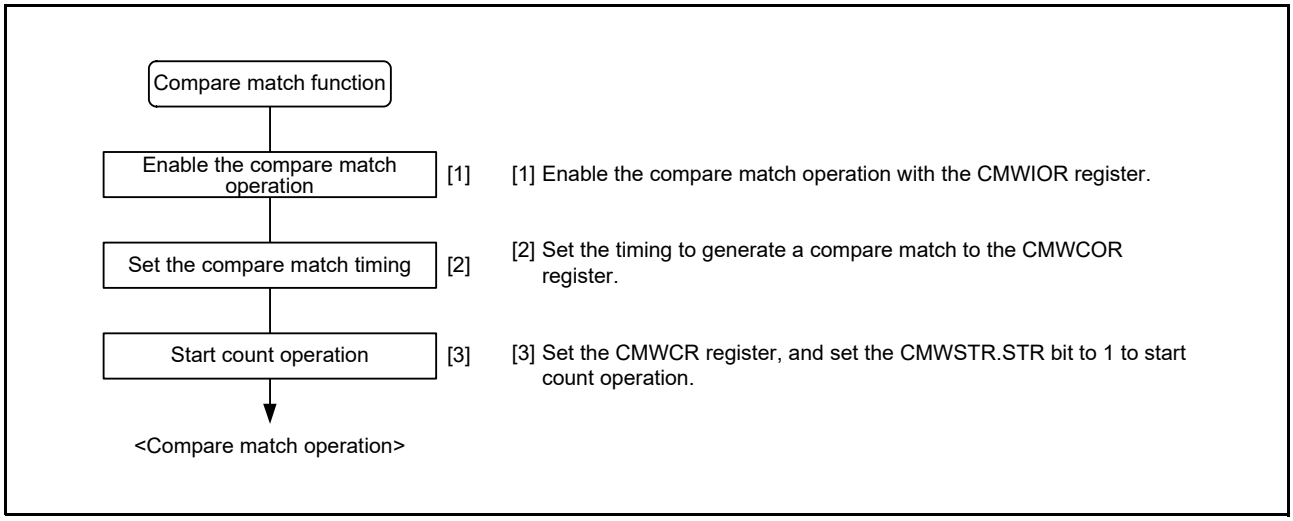


Figure 29.3 Procedure for Setting Compare Match Operation

Figure 29.4 shows an example when compare match with CMWCOR is set as a counter clearing source.

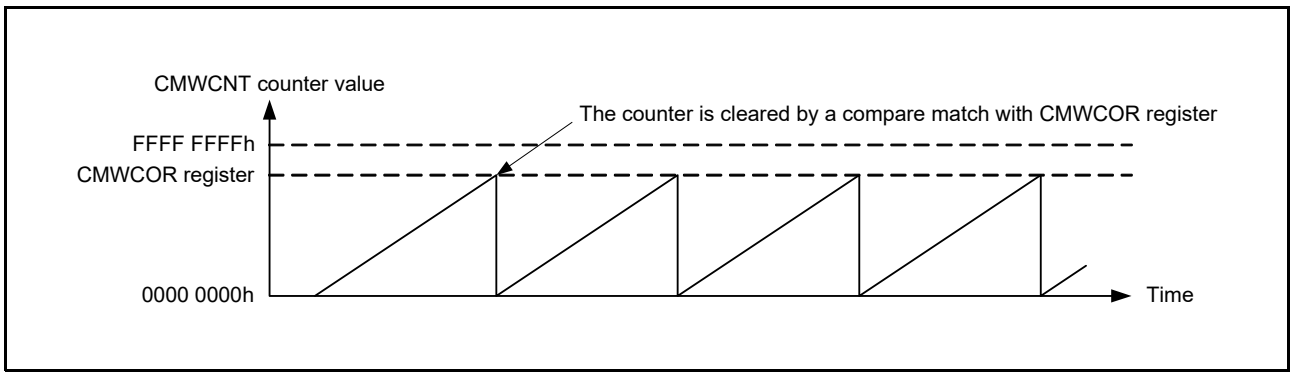


Figure 29.4 Example of Compare Match Operation

Figure 29.5 shows an example when CMWCOR is set to FFFF FFFFh and an overflow is detected.

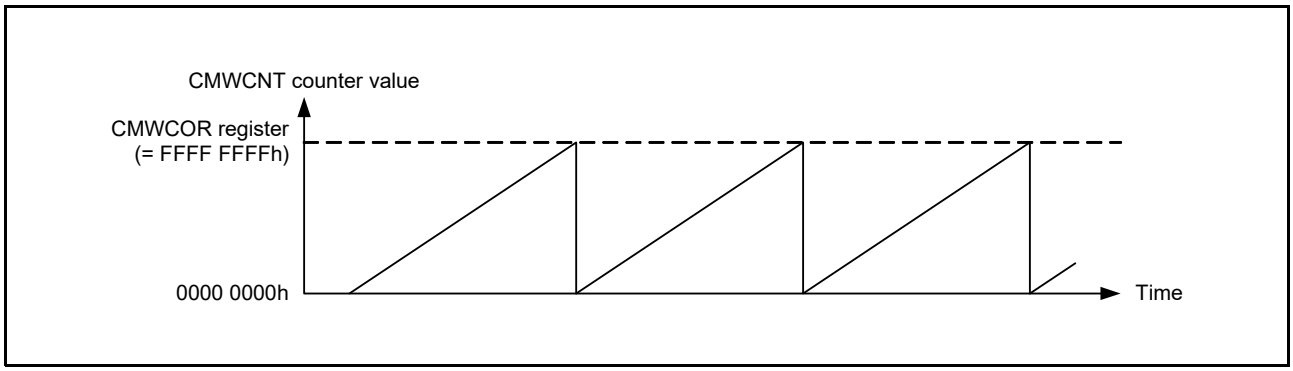


Figure 29.5 Example of Compare Match Operation (Overflow Detected)

29.3.3 Output Compare Function

The output compare function can be used for toggle waveform output. When the CMWCNT counter value matches either of the values of the CMWOCR0 or CMWOCR1 register, the output compare interrupt request (OC0I or OC1I) is generated. Figure 29.6 shows an example of procedure for setting output compare operation.

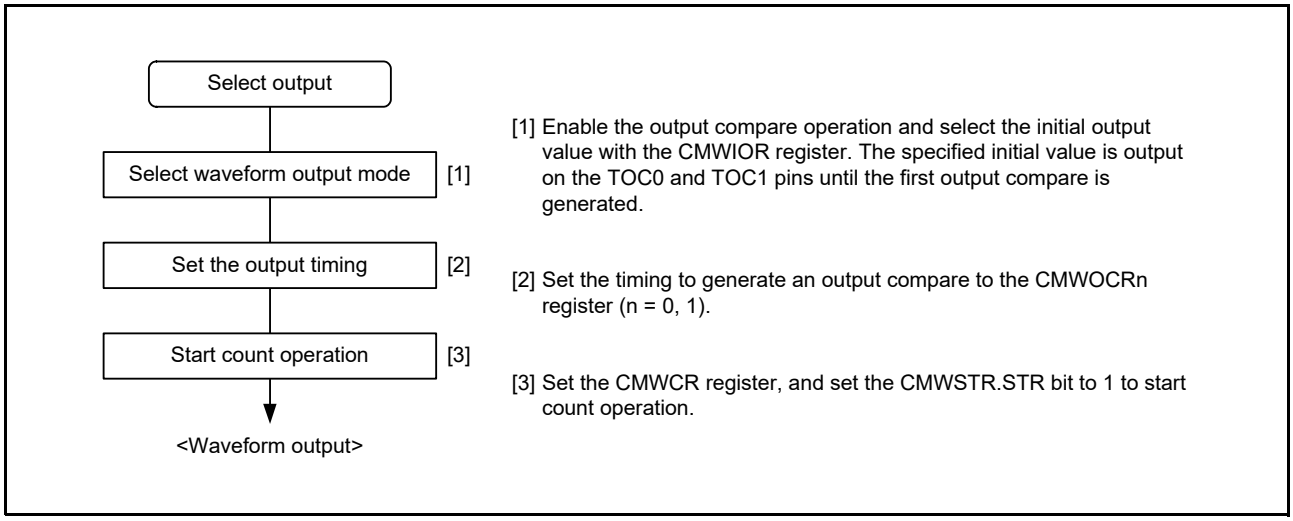


Figure 29.6 Procedure for Setting Output Compare Operation

Figure 29.7 shows an example of toggle waveform output from the TOC0 and TOC1 pins when the counter is set to be cleared by compare match with the CMWOCR1 register.

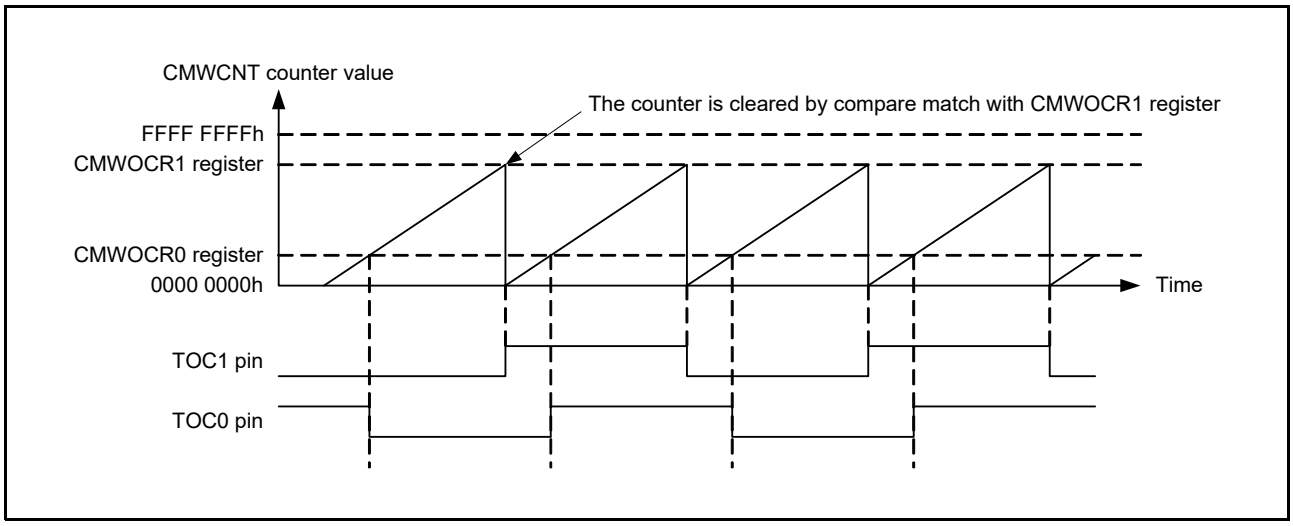


Figure 29.7 Example of Output Compare Operation (Unit 0)

29.3.4 Input Capture Function

Through detecting the edge on the TIC0 and TIC1 pin input, the CMWCNT counter value can be transferred to the CMWICR0 and CMWICR1 registers, respectively. The edges to be detected can be selected from among the rising edge alone, falling edge alone, and both the rising and falling edges. When the CMWCNT counter value is transferred to the CMWICR0 or CMWICR1 register using the input capture operation, an input capture interrupt request (IC0I or IC1I) is generated. Figure 29.8 shows an example of procedure for setting input capture operation.

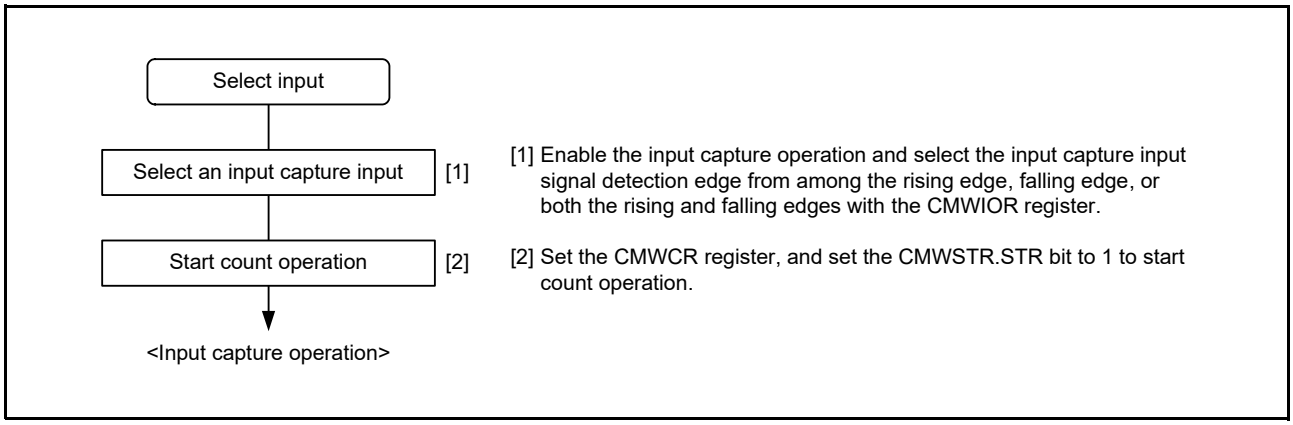


Figure 29.8 Procedure for Setting Input Capture Operation

Figure 29.9 shows an example in which both edges are selected for the TIC0 pin input capture detection edge and the falling edge for the TIC1 pin, and the CMWCNT counter is cleared by a CMWICR1 register input capture.

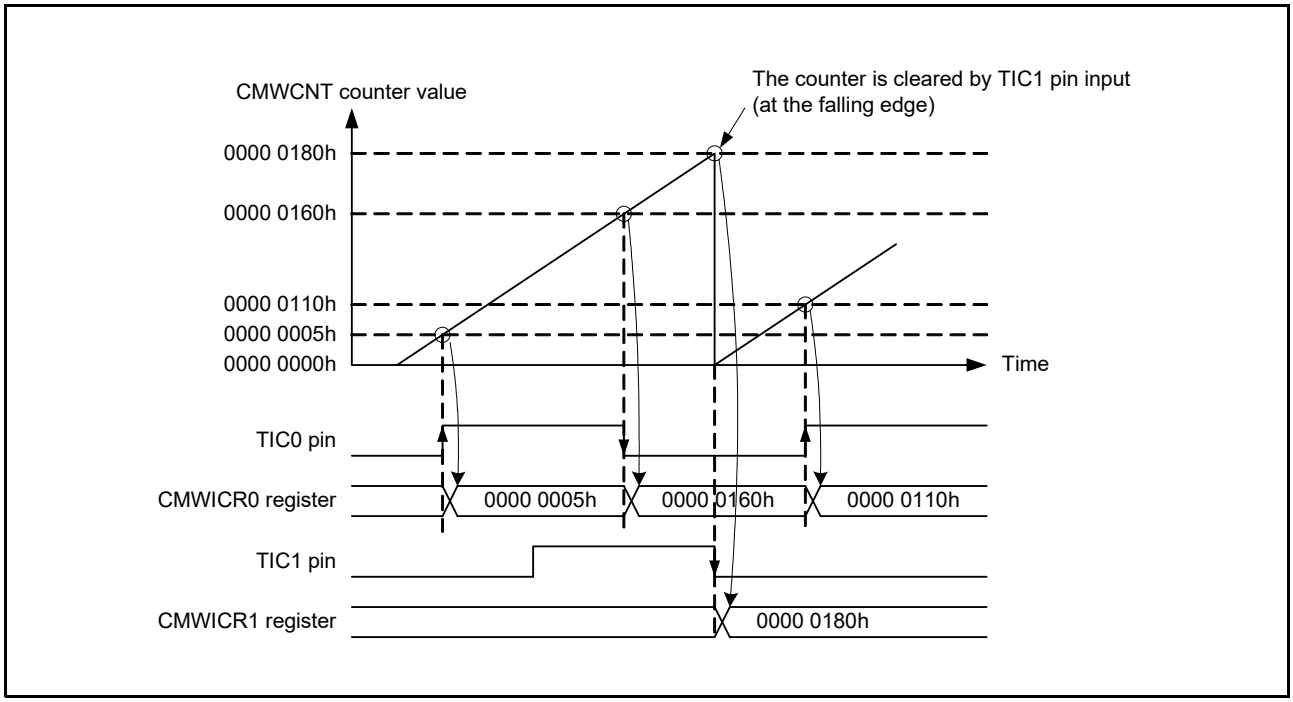


Figure 29.9 Example of Input Capture Operation (Unit 0)

29.3.5 Counter Size

With the CMTW, either 16 or 32 bits can be selected as the counter size by using the CMWCR.CMS bit.
When the counter is used as a 16-bit counter, set the value of the CMWCOR register in 32-bit units with the upper 16 bits set to 0000h. 0000 FFFFh should be set to detect an overflow. Similarly, set the values of the CMWOCR0 and CMWOCR1 registers in 32-bit units with the upper 16 bits set to 0000h. Read the CMWOCR0 and CMWOCR1 registers in 32-bit units. The upper 16 bits can be read as 0000h.

29.3.6 Count Timing of CMWCNT Counter

By setting the CMWCR.CKS[1:0] bits, one of four clocks (PCLK/8, PCLK/32, PCLK/128, and PCLK/512) obtained by dividing the peripheral module clock (PCLK) can be selected as the counter clock to be input to the CMWCNT counter.
Figure 29.10 shows the timing.

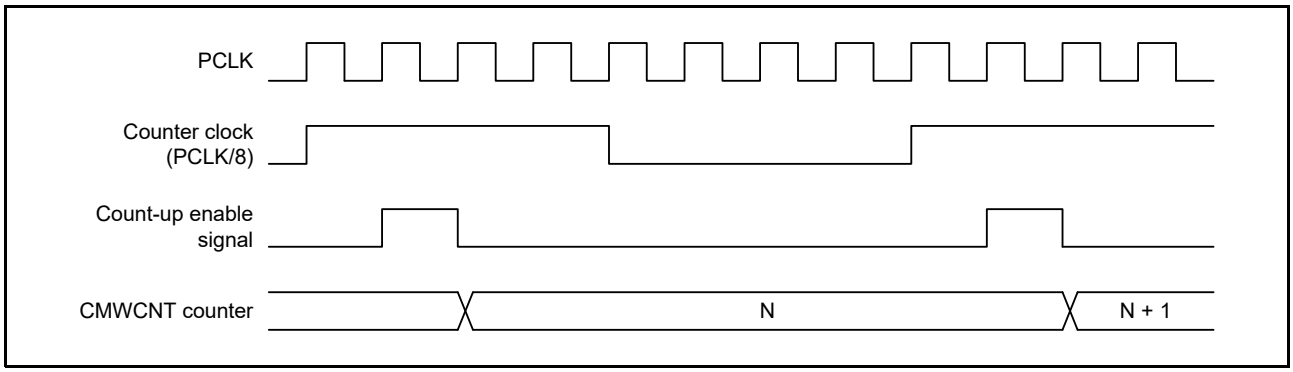


Figure 29.10 Count Timing (PCLK/8)

29.3.7 Output Compare Output Timing

A compare match signal is generated in the last state in which the CMWOCRn register (n = 0, 1) and CMWCNT counter values match (the CMWCNT counter value is updated immediately after the state). The compare match signal is generated if the CMWCNT count-up enable signal is input after a match between the CMWOCRn register and CMWCNT counter values. When a compare match signal is generated, output of the output compare pin (TOC pin) is toggled.
Figure 29.11 shows output compare output timing.

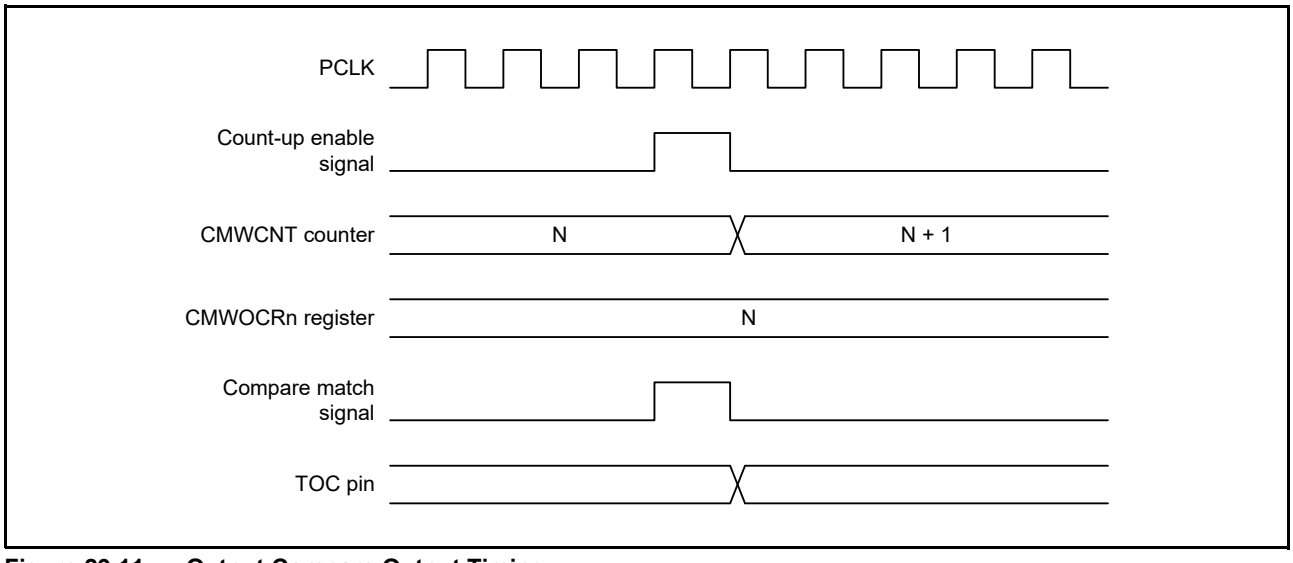


Figure 29.11 Output Compare Output Timing

29.3.8 Input Capture Timing

Figure 29.12 shows the timing of input capture operation at both edges.
When the edge of the TIC0 and TIC1 pins is detected, the CMWCNT counter value is transferred to the CMWICR0 and CMWICR1 registers, respectively.

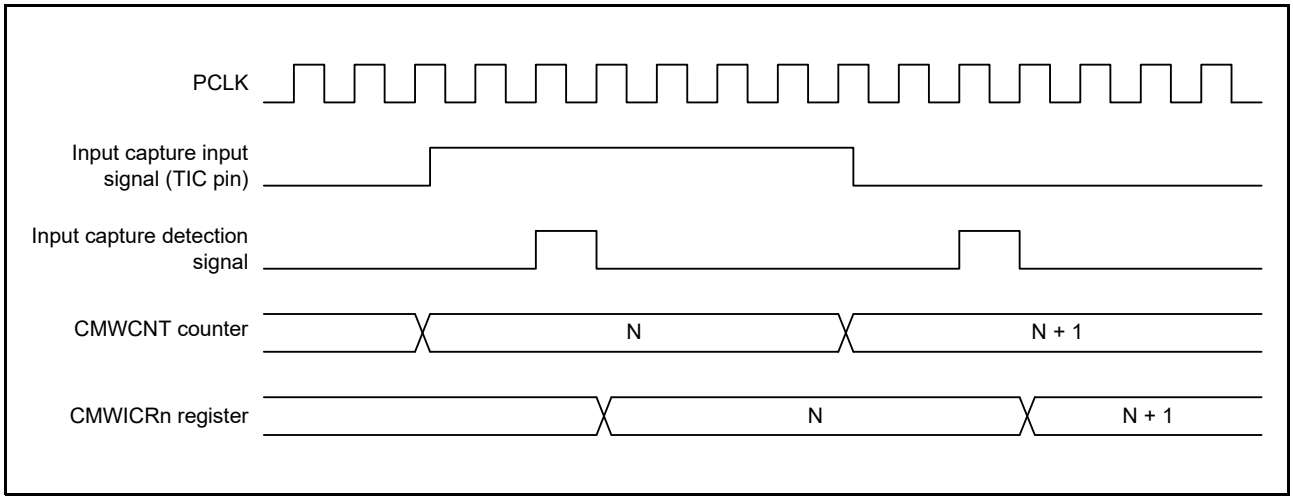


Figure 29.12 Input Capture Timing (Unit 0, Both-Edge Detection) (n = 0, 1)

29.4 Interrupts

29.4.1 CMTW Interrupt Sources and DTC/DMAC Transfer Requests

The CMTW has five interrupt sources: two input capture interrupt requests (IC0I and IC1I), two output compare interrupt requests (OC0I and OC1I), and a compare match interrupt request (CMWI).
Table 29.3 shows the interrupt sources. The interrupt sources can be enabled or disabled using the IC0IE, IC1IE, OC0IE, OC1IE, and CMWIE bits in CMWCR and are separately generated to the interrupt controller.
Each interrupt request can activate the DMAC or DTC. When the DMAC is used for data transfer, an interrupt request is not generated to the CPU. For generating an interrupt request to the CPU during data transfer using the DTC, refer to section 18, Data Transfer Controller (DTCb).

Table 29.3 CMTW Interrupt Sources

Unit	Name	Interrupt Request	Interrupt Request Enable Bit	DMAC/DTC Activation
CMTW0	CMWI0	Compare match of CMTW0.CMWCOR register	CMTW0.CMWCR.CMWIE	Possible
	IC0I0	Input capture of CMTW0.CMWICR0 register	CMTW0.CMWCR.IC0IE	Possible
	IC1I0	Input capture of CMTW0.CMWICR1 register	CMTW0.CMWCR.IC1IE	Possible
	OC0I0	Output compare of CMTW0.CMWOCR0 register	CMTW0.CMWCR.OC0IE	Possible
	OC1I0	Output compare of CMTW0.CMWOCR1 register	CMTW0.CMWCR.OC1IE	Possible
CMTW1	CMWI1	Compare match of CMTW1.CMWCOR register	CMTW1.CMWCR.CMWIE	Possible
	IC0I1	Input capture of CMTW1.CMWICR0 register	CMTW1.CMWCR.IC0IE	Possible
	IC1I1	Input capture of CMTW1.CMWICR1 register	CMTW1.CMWCR.IC1IE	Possible
	OC0I1	Output compare of CMTW1.CMWOCR0 register	CMTW1.CMWCR.OC0IE	Possible
	OC1I1	Output compare of CMTW1.CMWOCR1 register	CMTW1.CMWCR.OC1IE	Possible

29.4.2 Timing of Compare Match Interrupt Generation

When the values of the CMWCNT counter and CMWCOR register match, a compare match interrupt request (CMWI) is generated. The compare match signal is generated at the end of the cycle where the values matched (i.e. when the CMWCNT counter is updated from the matching counter value).The compare match signal, therefore, is not generated until a count-up enable signal is generated after the values of the CMWCNT counter and CMWCOR register have matched. Figure 29.13 shows the timing of compare match interrupt generation.

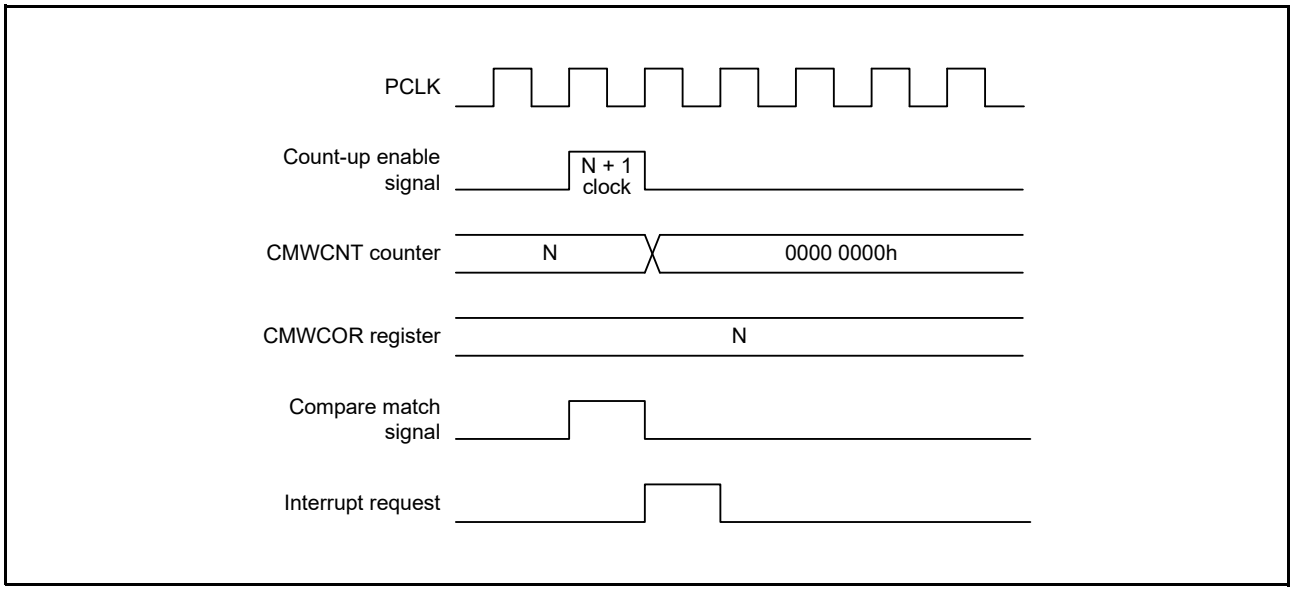


Figure 29.13 Timing of Compare Match Interrupt Generation

(a) Timing of Output Compare Interrupt Generation

Figure 29.14 shows the timing of output compare interrupt generation.

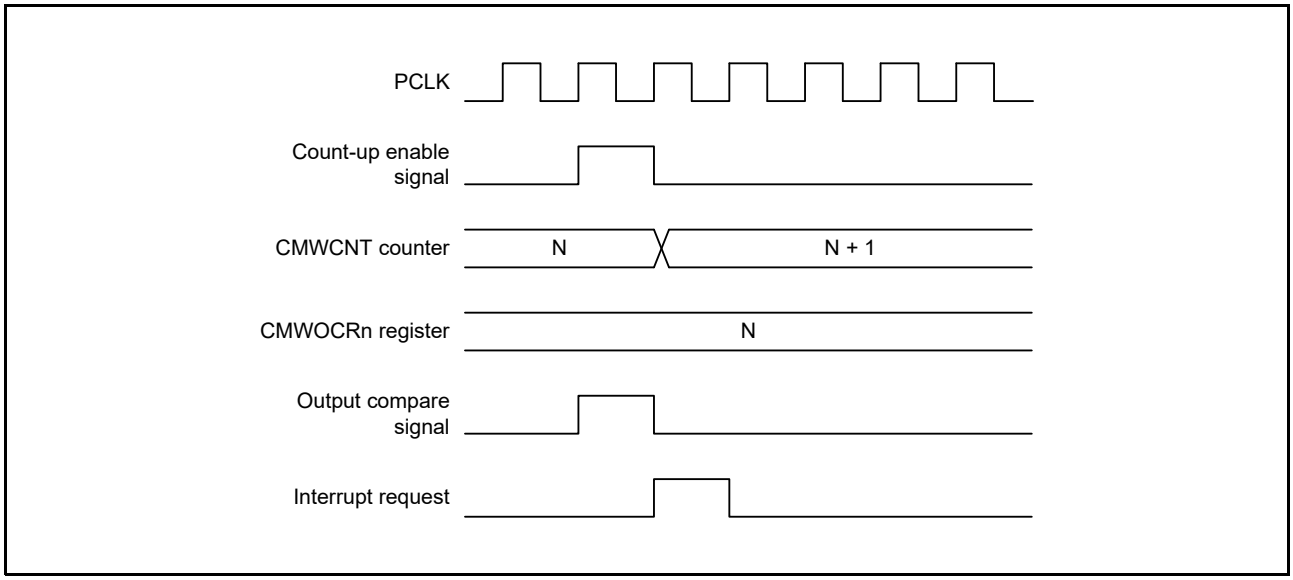


Figure 29.14 Timing of Output Compare Interrupt Generation (n = 0, 1)

(b) Timing of Input Capture Interrupt Generation

Figure 29.15 shows the timing of input capture interrupt generation.

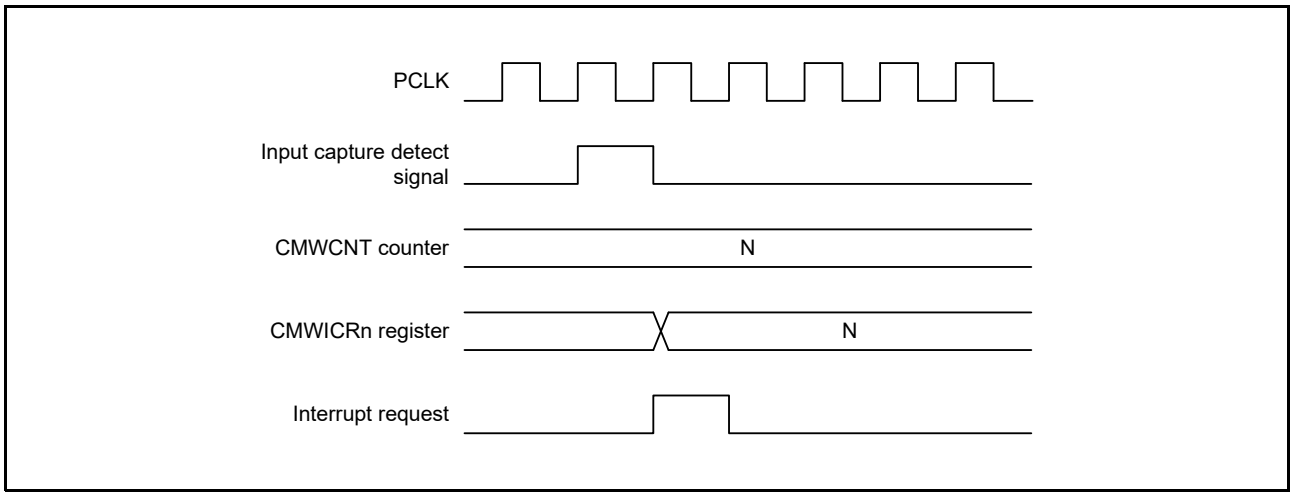


Figure 29.15 Timing of Input Capture Interrupt Generation (n = 0, 1)

29.5 Link Operations by ELC

29.5.1 Event Signal Output to ELC

The CMTW uses the event link controller (ELC) to perform link operation to a preset module using the interrupt request signal as the event signal.
The CMTW outputs the event signal upon a compare match. The corresponding channel is channel 0. The event signal can be output regardless of the corresponding interrupt request enable bit (CMWCR.CMWIE).
For details, refer to section 19, Event Link Controller (ELC).

(1) Compare Match Event

In response to a compare match, the CMTW simultaneously issues an interrupt request and a compare match event signal to the ELC. The event signal is issued regardless of the settings of the corresponding interrupt request enable bit (CMWCR.CMWIE bit).

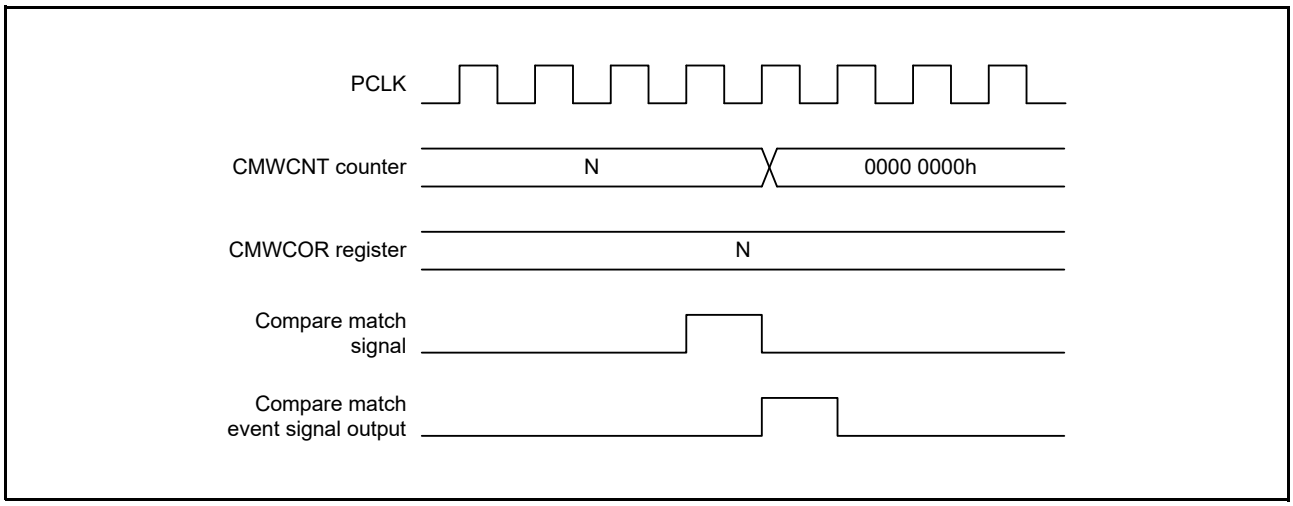


Figure 29.16 Timing of Issuing a Compare Match Event Signal

29.5.2 CMTW Operation When Receiving an Event Signal from ELC

The CMTW can perform any of the following operations according to the event preset by the ELSRn register of the ELC.

(1) Count Start

The CMTW count start operation is selected by the ELOPH register of the ELC. If the event specified by the ELSRn register occurs, the CMWSTR.STR bit is set to 1, starting the CMTW count operation.
However, if the specified event occurs while the CMWSTR.STR bit is 1, the event is ignored.

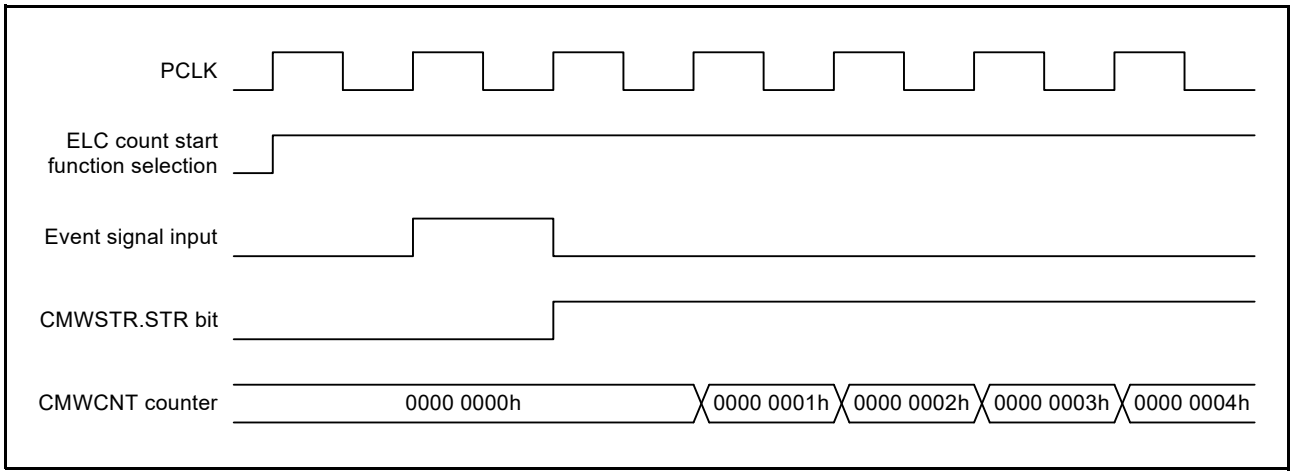


Figure 29.17 Count Start Operation on Acceptance of the Event Signal

(2) Event Count

The CMTW event count operation is selected by the ELOPH register of the ELC. If the event specified by the ELSRn register occurs when the CMWCR.STR bit is 1, the event is counted as the count source regardless of the CMWCR.CKS[1:0] bit setting.
Reading the counter value returns the number of events that have been actually input.

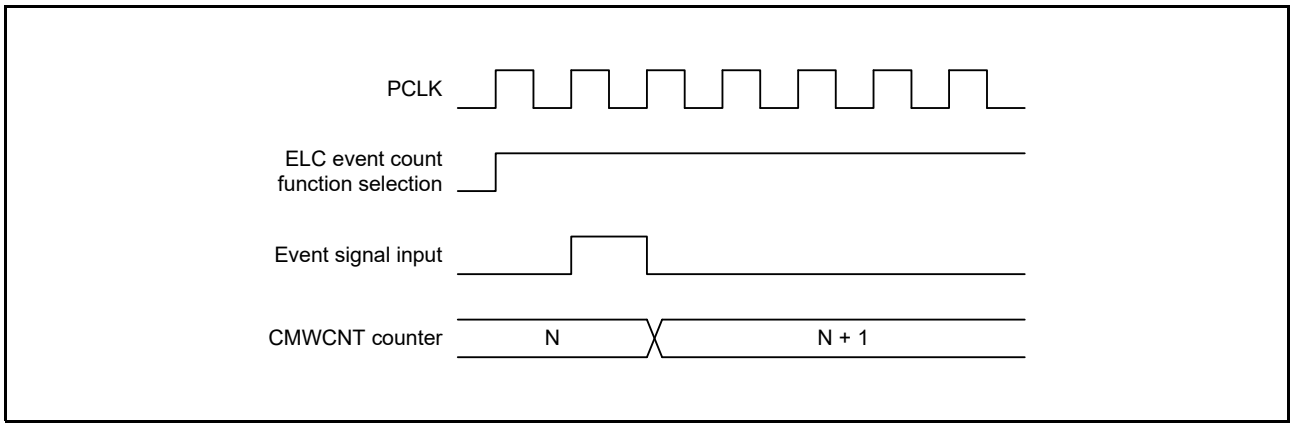


Figure 29.18 Event Count Operation on Acceptance of the Event Signal

(3) Count Restart

The CMTW count restart operation is selected by the ELOPH register of the ELC. If the event specified by the ELSRn register occurs, the CMWCNT counter value becomes 0000 0000h. If the CMWSTR.STR bit is 1, the count operation can be continued.

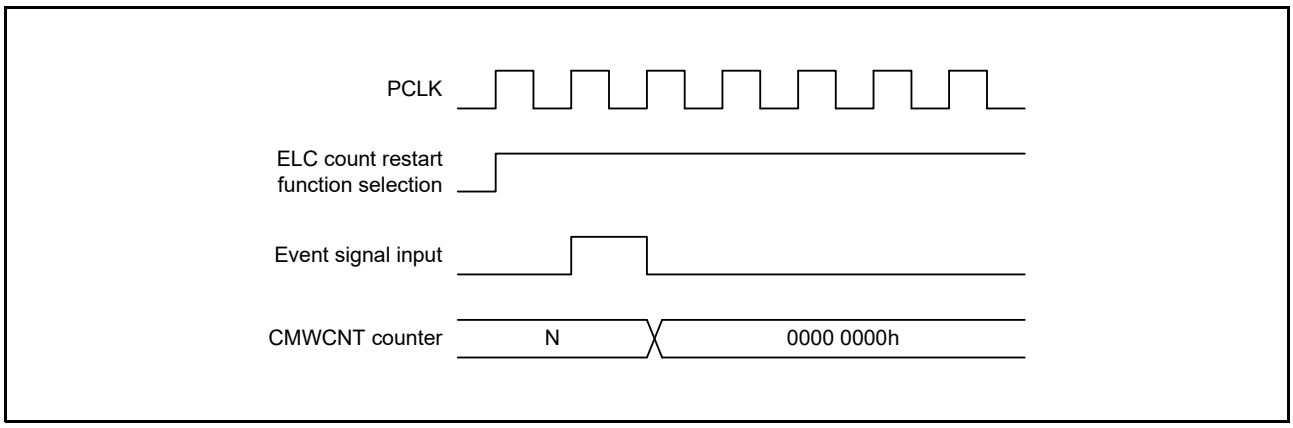


Figure 29.19 Count Restart Operation on Acceptance of the Event Signal

29.5.3 Conflict between Event Link Operation and Register Access

The followings are the notes on using CMTW for event link operations.
Table 29.4 lists count operations when event link operation and register access conflict.

Table 29.4 Count Operations When Event Link Operation and Register Access Conflict (n = 0, 1)

Event Link Operation	Register Access	CMWCNT Counter Status	Operation to be Performed
Count start	Writing to the CMWSTR.STR bit	Stopped state	Count start
		Compare match	Count start
		Counting up	Count start
Event count	Writing to CMWCNT counter	—	Event count
	Writing to CMWCOR register	Compare match	Compare match
Count restart	Writing to CMWCNT counter	Other than compare match	Count restart
	Writing to CMWCNT counter	Compare match	Compare match
	(No access to registers)	Compare match	Compare match
(No events)	Writing to CMWCNT counter	Compare match	Output of compare match interrupt request Writing to CMWCNT counter
		Counting up	Writing to CMWCNT counter
	Writing to CMWCOR register	Compare match	Compare match
	Writing to CMWOCRn register	Compare match	Compare match
	Reading from CMWCNT counter	Counting up	Counting up and reading of the previous value

(1) Count Start

When writing to the STR bit in the CMWSTR register and acceptance of the event signal are in contention, writing to the STR bit does not proceed since setting of the STR bit to 1 in response to the event takes priority.

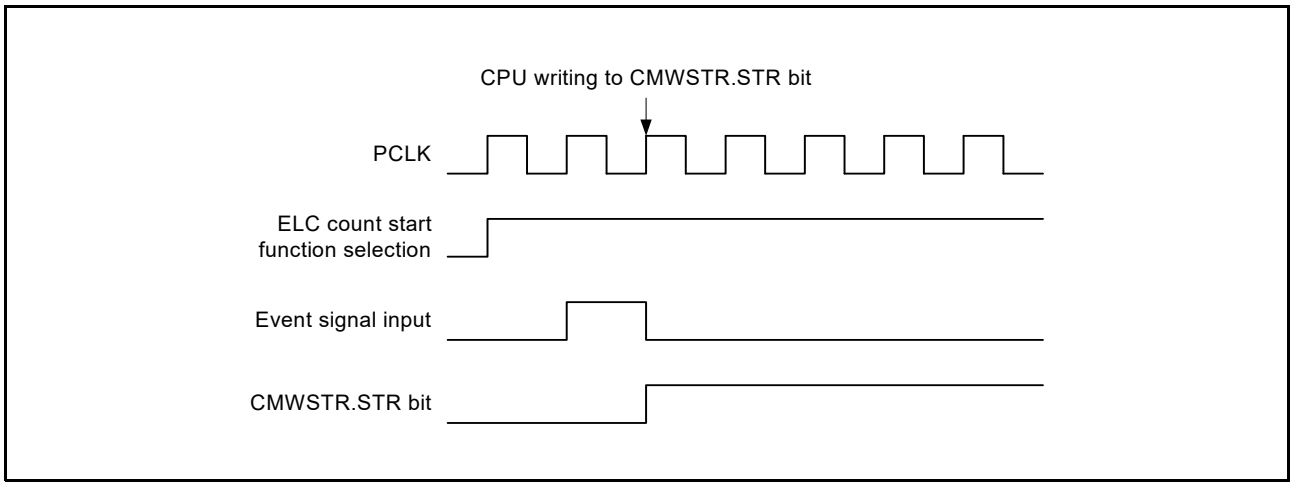


Figure 29.20 Conflict between Event Acceptance and Register Access in Count Start Operation

(2) Event Count

When writing to the CMWCNT counter and acceptance of the event signal are in contention, writing to the CMWCNT counter does not proceed since the count operation in response to the event takes priority.

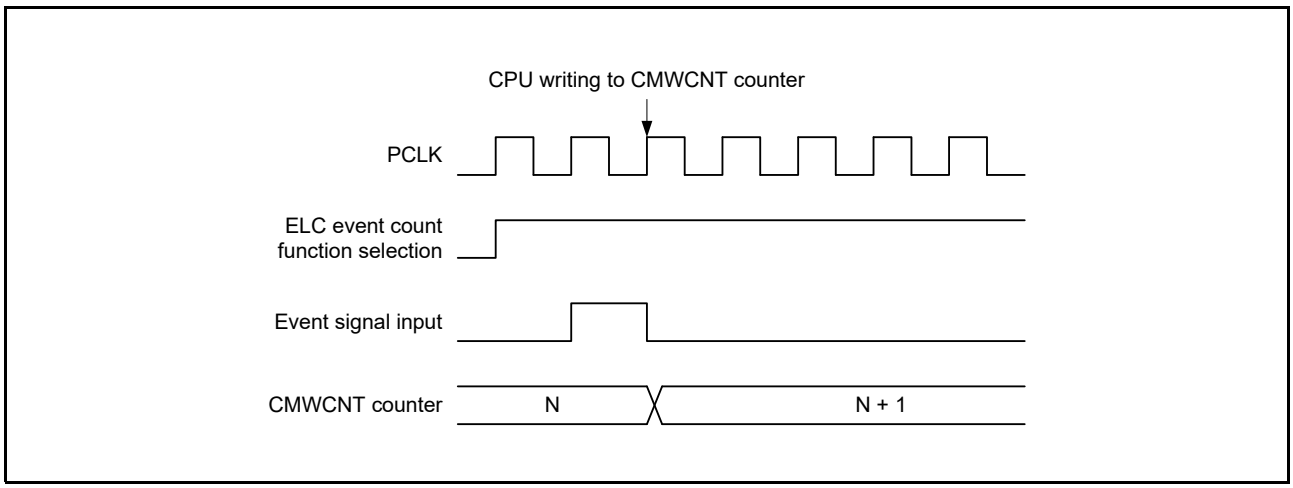


Figure 29.21 Conflict between Event Acceptance and Register Access in Event Counting Operation

(3) Count Restart

When writing to the CMWCNT counter and acceptance of the event signal are in contention, writing to the CMWCNT counter does not proceed since the counter value initialization in response to the event occurrence takes priority.

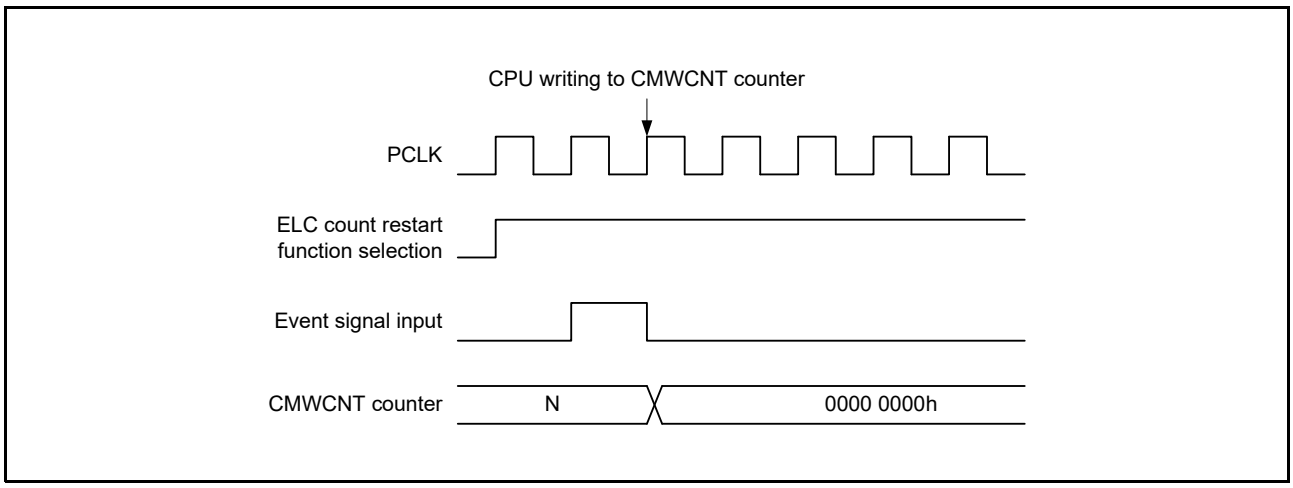


Figure 29.22 Conflict between Event Acceptance and Register Access in Count Restart Operation

29.6 Usage Notes

29.6.1 Setting the Module Stop Function

The CMTW operation can be enabled or disabled using the MSTPCRA register. The CMTW0 and CMTW1 are initially disabled after a reset. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

29.6.2 Conflict between CMWCNT Counter Writing and Compare Match

If the compare match signal is generated during writing to the CMWCNT counter, the compare match request is output but the counter is not cleared since writing to the counter takes priority.
Figure 29.23 shows the timing of conflict between CMWCNT counter writing and compare match.

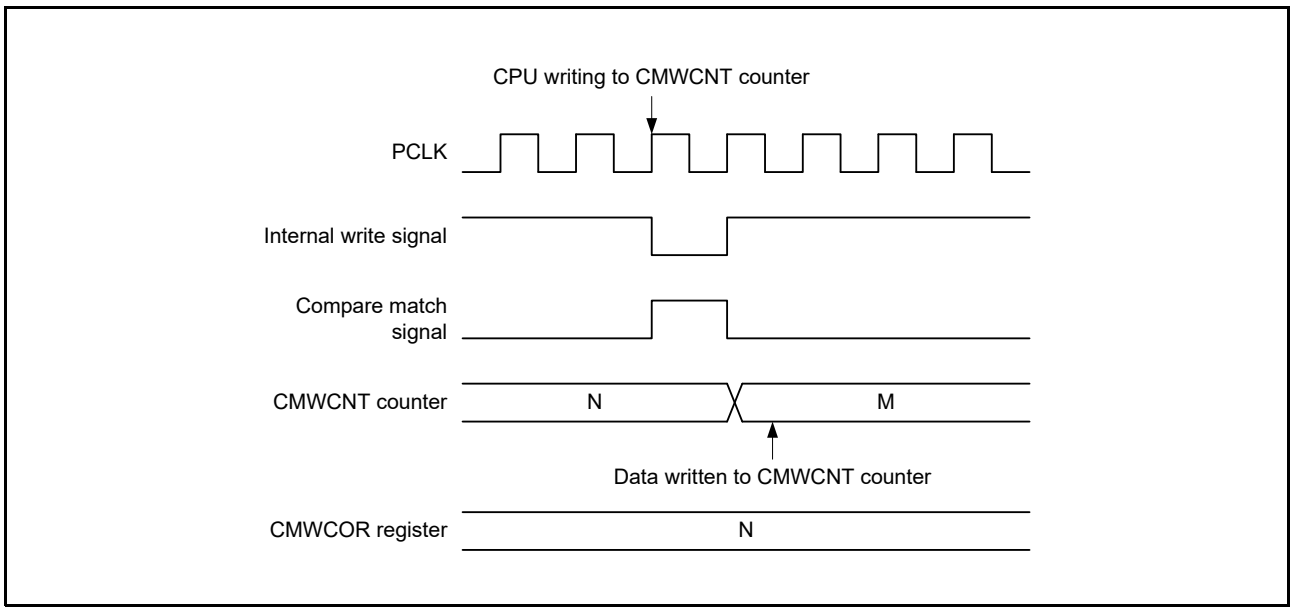


Figure 29.23 Conflict between CMWCNT Counter Writing and Compare Match

29.6.3 Conflict between CMWCNT Counter Writing and Incrementing or Clearing

In case of conflict between incrementation or clearing of the CMWCNT counter and writing to the CMWCNT counter, the counter is not actually incremented or cleared since writing to the CMWCNT counter takes priority.
Figure 29.24 shows the timing in the case of contention between writing to the CMWCNT counter and incrementation or clearing.

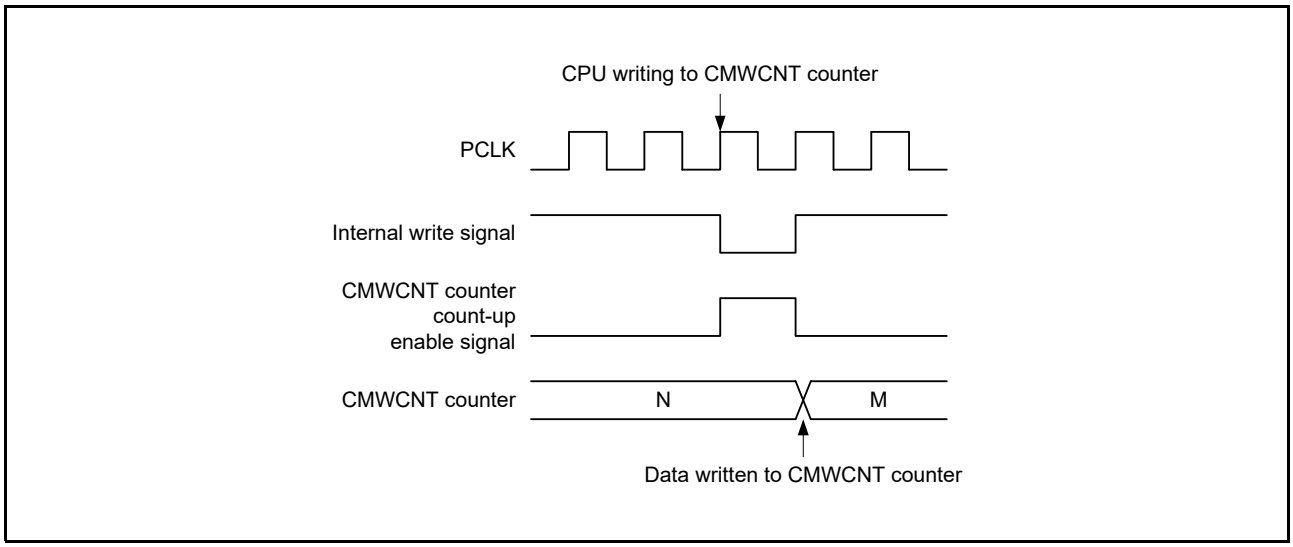


Figure 29.24 Conflict between CMWCNT Counter Writing and Incrementing

29.6.4 Conflict between CMWCOR Register Writing and Compare Match

If the compare match is generated during the CMWCOR register write cycle, the writing to the CMWCOR register takes priority and also the compare match signal is output.
Figure 29.25 shows the timing of conflict between CMWCOR register writing and compare match.

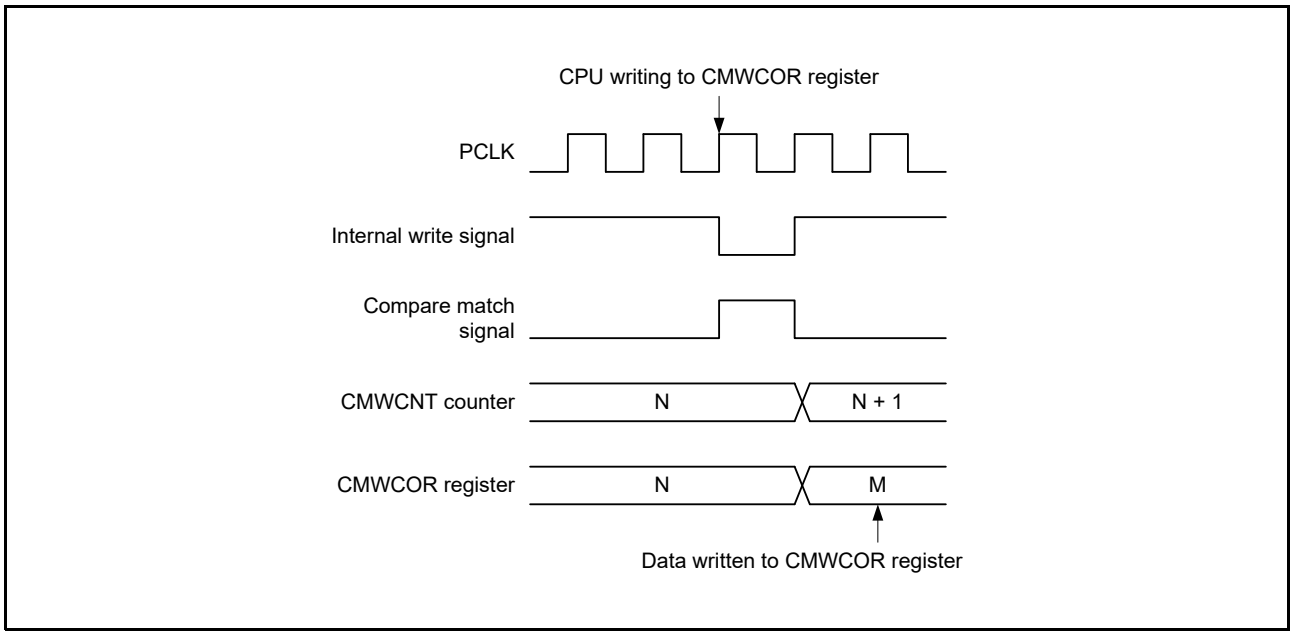


Figure 29.25 Conflict between CMWCOR Register Writing and Compare Match

29.6.5 Conflict between CMWOCRn Register Writing and Compare Match (n = 0, 1)

If the compare match is generated during the CMWOCRn register write cycle, the writing to the CMWOCRn register takes priority and also the compare match signal is output.
Figure 29.26 shows the timing of conflict between CMWOCRn register writing and compare match.

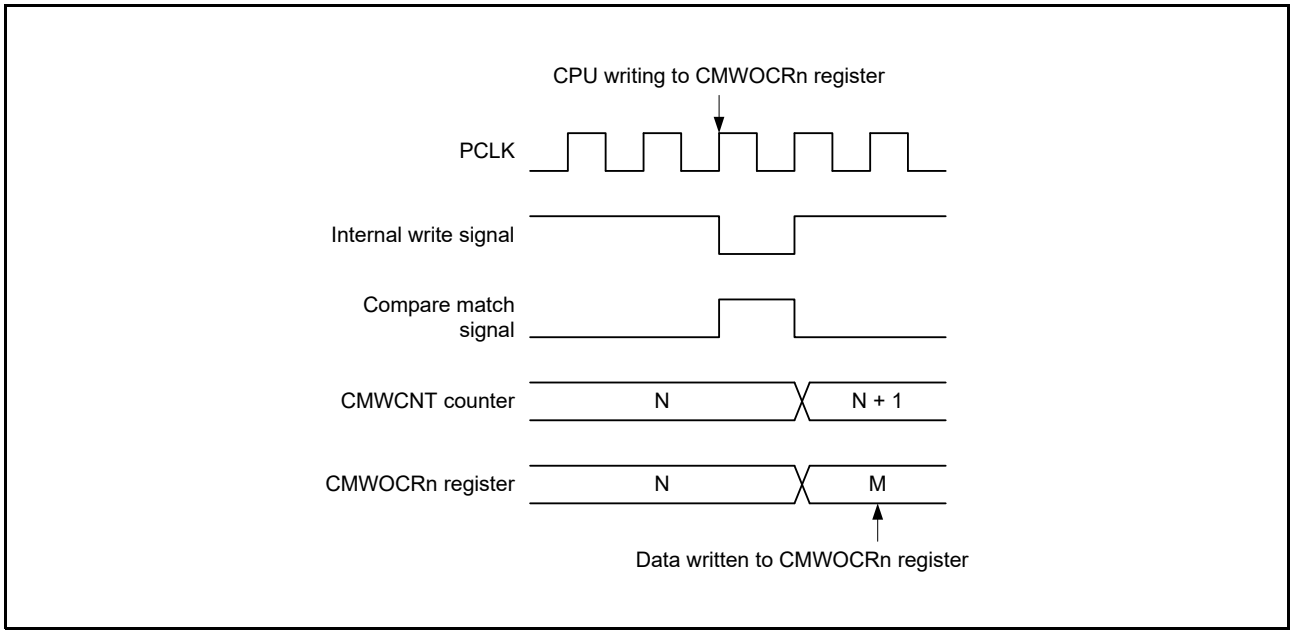


Figure 29.26 Conflict between CMWOCRn Register Writing and Compare Match

29.6.6 Conflict between CMWCNT Counter Reading and Incrementing or Clearing

If the CMWCNT counter incrementing or clearing process occurs at the same time that the data of the CMWCNT counter is read, the value having been in the CMWCNT counter before incremented or cleared is read.
Figure 29.27 shows the timing of conflict between CMWCNT counter reading and incrementing.

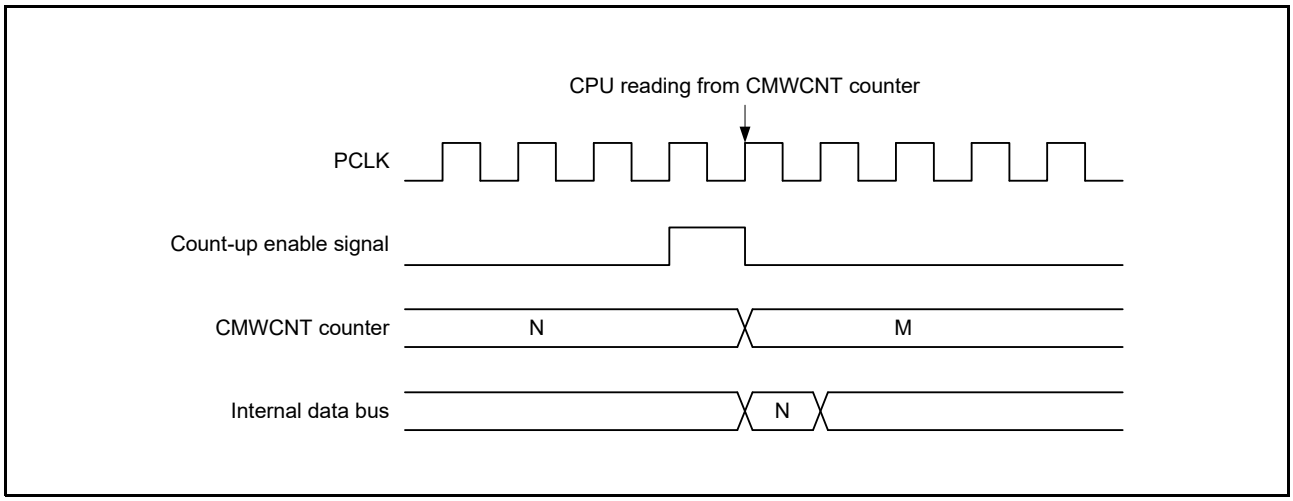


Figure 29.27 Conflict between CMWCNT Counter Reading and Incrementing

29.6.7 Conflict between CMWICRn Register Reading and Input Capture (n = 0, 1)

If the input capture detection signal is generated at the same time that the data of the CMWICRn register is read, the value having been in the CMWICRn register before updated by input capture transfer is read.
Figure 29.28 shows the timing of conflict between CMWICRn register reading and input capture.

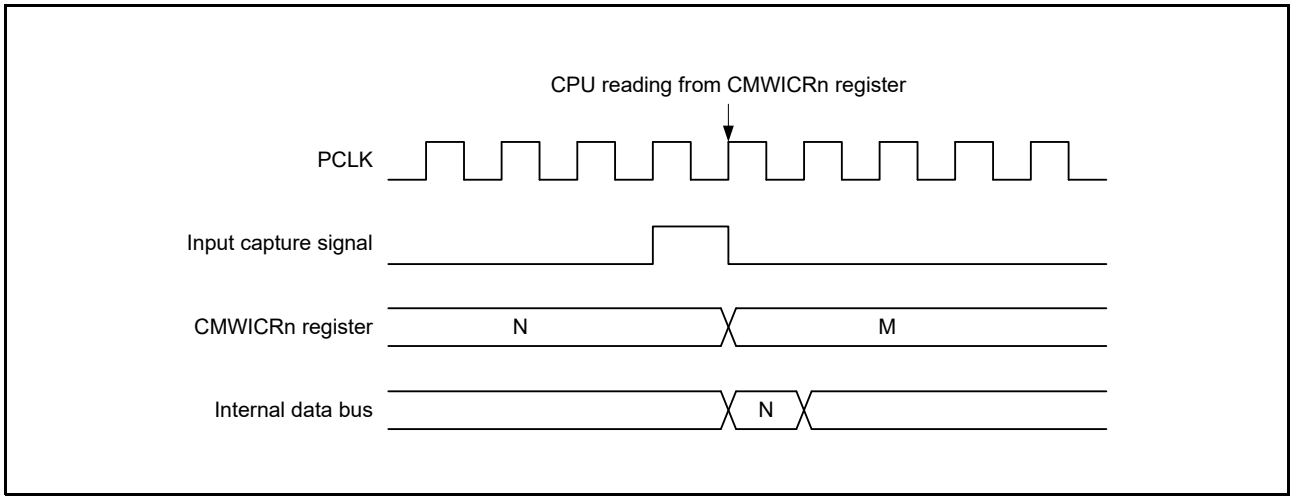


Figure 29.28 Conflict between CMWICRn Register Reading and Input Capture

30. Watchdog Timer (WDTA)

The watchdog timer (WDT) is a 14-bit down-counter. It can be used to reset this MCU when the counter underflows because its value cannot be refreshed due to the system being out of control.
In addition, a non-maskable interrupt can be generated by an underflow.
The refresh-permitted period can be set to refresh the counter and used as the condition to detect when the system runs out of control.
In this section, “PCLK” is used to refer to PCLKB.

30.1 Overview

Table 30.1 lists the specifications of the WDT and Figure 30.1 shows a block diagram of the WDT.

Table 30.1 WDT Specifications

Item	Specifications
Count source	Peripheral module clock (PCLK)
Clock division ratio	Divide by 4, 64, 128, 512, 2048, or 8192
Counter operation	Counting down using a 14-bit down-counter
Conditions for starting the counter	• Auto-start mode: Counting automatically starts after a reset is released • Register start mode: Counting is started by refresh operation (writing 00h and then FFh to the WDTRR register)
Conditions for stopping the counter	• Reset • In low power consumption states • A counter underflows or a refresh error occurs (only in register start mode)
Window function	Window start and end positions can be specified (refresh-permitted and refresh-prohibited periods)
Watchdog timer Reset sources	• Down-counter underflows • Refreshing outside the refresh-permitted period (refresh error)
Non-maskable interrupt/interrupt sources	• Down-counter underflows • Refreshing outside the refresh-permitted period (refresh error)
Reading the counter value	The down-counter value can be read by the WDTSR register.

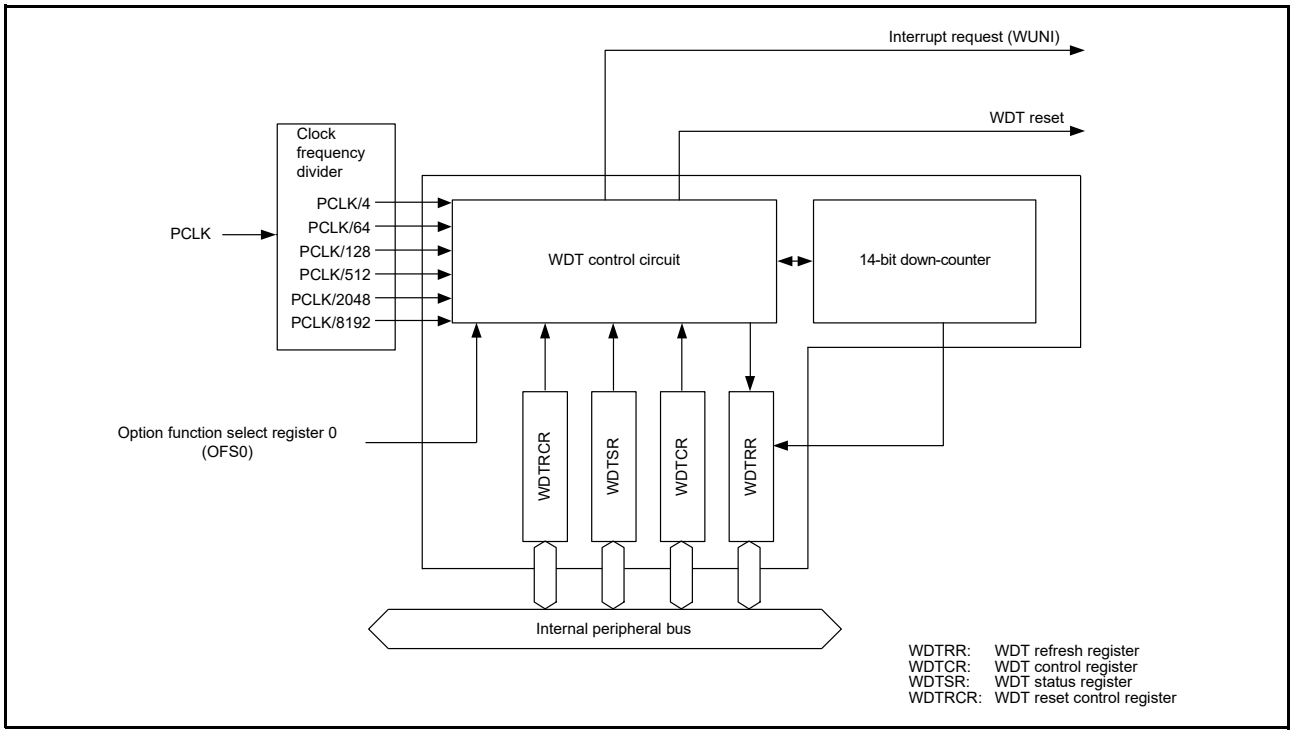
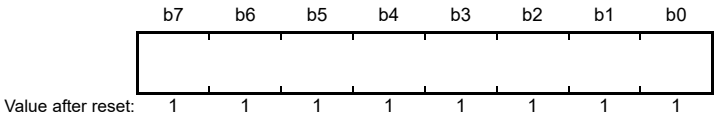


Figure 30.1 WDT Block Diagram

30.2 Register Descriptions

30.2.1 WDT Refresh Register (WDTRR)

Address(es): 0008 8020h



Bit	Description	R/W
b7 to b0	The down-counter is refreshed by writing 00h and then writing FFh to this register	R/W

WDTRR refreshes the down-counter of the WDT.
The down-counter of the WDT is refreshed by writing 00h and then writing FFh to WDTRR (refresh operation) within the refresh-permitted period.
After the down-counter has been refreshed, it starts counting down from the value selected by setting the WDTTOPS[1:0] bits in option function select register 0 (OFS0) in auto-start mode. In register start mode, counting down starts from the value selected by setting the WDTCR.TOPS[1:0] bits.
When 00h is written, the read value is 00h, when a value other than 00h is written, the read value is FFh.
For details of the refresh operation, refer to section 30.3.3, Refresh Operation.

30.2.2 WDT Control Register (WDTCR)

Address(es): 0008 8022h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	RPSS[1:0]		—	—	RPES[1:0]		CKS[3:0]				—	—	TOPS[1:0]	
Value after reset:	0	0	1	1	0	0	1	1	1	1	1	1	0	0	1	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TOPS[1:0]	Timeout Period Selection	b1 b0 0 0: 1024 cycles (03FFh) 0 1: 4096 cycles (0FFFh) 1 0: 8192 cycles (1FFFh) 1 1: 16384 cycles (3FFFh)	R/W
b3, b2	—	Reserved	These bits are read as 0 and cannot be modified.	R
b7 to b4	CKS[3:0]	Clock Division Ratio Selection	b7 b4 0 0 0 1: Divide-by-4 0 1 0 0: Divide-by-64 1 1 1 1: Divide-by-128 0 1 1 0: Divide-by-512 0 1 1 1: Divide-by-2048 1 0 0 0: Divide-by-8192 Setting other than above are prohibited.	R/W
b9, b8	RPES[1:0]	Window End Position Selection	b9 b8 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (window end position is not specified)	R/W
b11, b10	—	Reserved	These bits are read as 0 and cannot be modified.	R
b13, b12	RPSS[1:0]	Window Start Position Selection	b13 b12 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (window start position is not specified)	R/W
b15, b14	—	Reserved	These bits are read as 0 and cannot be modified.	R

There are some restrictions on writing to the WDTCR register. For details, refer to section 30.3.2, Control over Writing to the WDTCR and WDTRCR Registers.
In auto-start mode, the settings in the WDTCR register are disabled, and the settings in option function select register 0 (OFS0) are enabled. The bit setting made to the WDTCR register can also be made in OFS0 register. For details, refer to section 30.3.7, Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers.

TOPS[1:0] Bits (Timeout Period Selection)

These bits select the timeout period (period until the down-counter underflows) from among 1024, 4096, 8192, and 16384 cycles, taking the divided clock specified by the CKS[3:0] bits as one cycle.
After the down-counter is refreshed, the combination of the CKS[3:0] and TOPS[1:0] bits determines the time (number of PCLK cycles) until the counter underflows.
Relations between the CKS[3:0] and TOPS[1:0] bit settings, the timeout period, and the number of PCLK cycles are listed in Table 30.2.

Table 30.2 Timeout Period Settings

CKS[3:0] Bits				TOPS[1:0] Bits		Clock Division Ratio	Timeout Period (Number of Cycles)	Cycles of PCLK Clock
b7	b6	b5	b4	b1	b0			
0	0	0	1	0	0	Divide-by-4	1024	4096
				0	1		4096	16384
				1	0		8192	32768
				1	1		16384	65536
0	1	0	0	0	0	Divide-by-64	1024	65536
				0	1		4096	262144
				1	0		8192	524288
				1	1		16384	1048576
1	1	1	1	0	0	Divide-by-128	1024	131072
				0	1		4096	524288
				1	0		8192	1048576
				1	1		16384	2097152
0	1	1	0	0	0	Divide-by-512	1024	524288
				0	1		4096	2097152
				1	0		8192	4194304
				1	1		16384	8388608
0	1	1	1	0	0	Divide-by-2048	1024	2097152
				0	1		4096	8388608
				1	0		8192	16777216
				1	1		16384	33554432
1	0	0	0	0	0	Divide-by-8192	1024	8388608
				0	1		4096	33554432
				1	0		8192	67108864
				1	1		16384	134217728

CKS[3:0] Bits (Clock Division Ratio Selection)

These bits specify the division ration of the clock used for the down-counter. The division ration can be selected from among the peripheral module clock (PCLK) divided by 4, 64, 128, 512, 2048, and 8192. Combined with the TOPS[1:0] bit setting, a count period between 4,096 and 134,217,728 cycles of the PCLK clock can be selected for the WDT.

RPES[1:0] Bits (Window End Position Selection)

These bits specify the window end position that indicates the refresh-permitted period. 75%, 50%, 25%, or 0% of the timeout period can be selected for the window end position. The selected window end position should be a value smaller than the window start position (window start position > window end position). If the window end position is greater than the window start position, only the window start position setting is enabled.

RPSS[1:0] Bits (Window Start Position Selection)

These bits specify the window start position that indicates the refresh-permitted period. 25%, 50%, 75%, or 100% of the timeout period can be selected for the window end position. The window start position should be set to a value greater the window end position. If the window start position is set to a value smaller than or equal to the window end position, the window end position is set to 0%.

Table 30.3 lists the counter values for the window start and end positions and Figure 30.2 shows the refresh-permitted period set by the RPSS[1:0], RPES[1:0], and TOPS[1:0] bits.

Table 30.3 Relationship between Timeout Period and Window Start and End Counter Values

TOPS[1:0] Bits		Timeout Period		Window Start and End Counter Value			
		Cycles	Counter Value	100%	75%	50%	25%
0	0	1024	03FFh	03FFh	02FFh	01FFh	00FFh
0	1	4096	0FFFh	0FFFh	0BFFh	07FFh	03FFh
1	0	8192	1FFFh	1FFFh	17FFh	0FFFh	07FFh
1	1	16384	3FFFh	3FFFh	2FFFh	1FFFh	0FFFh

RPSS[1:0] Bits		RPES[1:0] Bits		Window	
b13	b12	b9	b8	Start (%)	End (%)
1	1	1	1	100	0
		1	0		25
		0	1		50
		0	0		75
1	0	1	1	75	0
		1	0		25
		0	1		50
		0	0		75
0	1	1	1	50	0
		1	0		25
		0	1		50
		0	0		75
0	0	1	1	25	0
		1	0		25
		0	1		50
		0	0		75

Note: If window end setting ≥ window start setting, the window end setting is set to 0%.

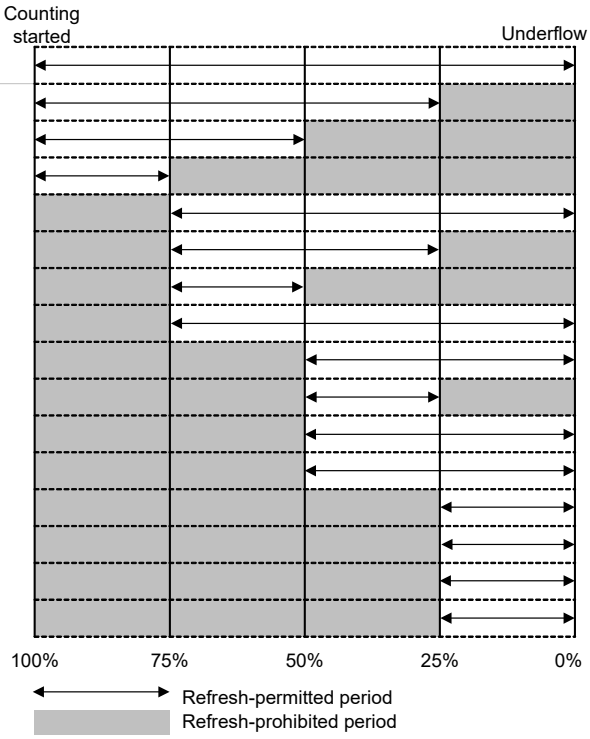


Figure 30.2 RPSS[1:0] and RPES[1:0] Bit Settings and the Refresh-Permitted Period

30.2.3 WDT Status Register (WDTSR)

Address(es): 0008 8024h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	REFEF	UNDFF	CNTVAL[13:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b13 to b0	CNTVAL[13:0]	Down-Counter Value	Value counted by the down-counter	R
b14	UNDFF	Underflow Flag	0: No underflow occurred 1: Underflow occurred	R(/W) *1
b15	REFEF	Refresh Error Flag	0: No refresh error occurred 1: Refresh error occurred	R(/W) *1

Note 1. Only 0 can be written to clear the flag.

CNTVAL[13:0] Bits (Down-Counter Value)

Read these bits to confirm the value of the down-counter, but note that the read value may differ from the actual count by a value of one count.

UNDFF Flag (Underflow Flag)

Read this flag to confirm whether or not an underflow has occurred in the down-counter.

The value 1 indicates that the down-counter has underflowed. The value 0 indicates that the down-counter has not underflowed.

Write 0 to the UNDFF flag to set the value to 0. Writing 1 has no effect.

REFEF Flag (Refresh Error Flag)

Read this flag to confirm whether or not a refresh error (performing a refresh operation during a refresh-prohibited period) has occurred.

The value 1 indicates that a refresh error has occurred. The value 0 indicates that no refresh error has occurred.

Write 0 to the REFEF flag to set the value to 0. Writing 1 has no effect.

30.2.4 WDT Reset Control Register (WDTRCR)

Address(es): 0008 8026h

	b7	b6	b5	b4	b3	b2	b1	b0
	RSTIRQS	—	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0 and cannot be modified.	R
b7	RSTIRQS	Reset Interrupt Request Selection	0: Non-maskable interrupt request or interrupt request output is enabled 1: Reset output is enabled	R/W

There are some restrictions on writing to the WDTRCR register. For details, refer to section 30.3.2, Control over Writing to the WDTCR and WDTRCR Registers.
In auto-start mode, the WDTRCR register settings are disabled, and the settings in option function select register 0 (OFS0) are enabled. The bit setting made to the WDTCR register can also be made in the OFS0 register. For details, refer to section 30.3.7, Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers.

30.2.5 Option Function Select Register 0 (OFS0)

For details on the OFS0 register, refer to section 30.3.7, Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers.

30.3 Operation

30.3.1 Count Operation in Each Start Mode

The WDT has two start modes: auto-start mode, in which counting automatically starts after a reset is released, and register start mode, in which counting is started by refresh operation (writing to the register).
In auto-start mode, counting automatically starts after a reset is released in accordance with the settings in option function select register 0 (OFS0) in the ROM.
In register start mode, counting is started by refresh operation (writing to the register) after the respective registers are set after a reset is released.
Select auto-start mode or register start mode by setting the OFS0.WDTSTRT bit.
When the auto-start mode is selected, the settings in the WDTCR and WDTRCR registers are disabled, and the settings in the OFS0 register are enabled.
On the other hand, when the register start mode is selected, the setting of the OFS0 register is disabled, and the settings of the WDTCR and WDTRCR registers are enabled.

30.3.1.1 Register Start Mode

When the OFS0.WDTSTRT bit is 1, register start mode is selected, and the WDTCR and WDTRCR registers are enabled.
After a reset is released, set the clock division ratio, window start and end positions, and timeout period in the WDTCR register, and the reset output or interrupt request output in the WDTRCR register. Then, refresh the down-counter to start counting down from the value set by the WDTCR.TOPS[1:0] bits.
Thereafter, as long as the counter is refreshed in the refresh-permitted period, the value in the counter is re-set each time the counter is refreshed and counting down continues. The WDT does not output the reset signal as long as this continues. However, if the down-counter underflows because the down-counter cannot be refreshed due to a program runaway, or if a refresh error occurs because the counter was refreshed outside the refresh-permitted period, the WDT outputs a reset signal or a non-maskable interrupt request/interrupt request (WUNI). Reset output or interrupt request output can be selected by setting the WDTRCR.RSTIRQS bit.

Figure 30.3 shows an example of operation under the following conditions.

- Register start mode (OFS0.WDTSTRT = 1)
- Reset output is enabled (WDTRCR.RSTIRQS = 1)
- The window start position is 75% (WDTCR.RPSS[1:0] = 10b)
- The window end position is 25% (WDTCR.RPES[1:0] = 10b)

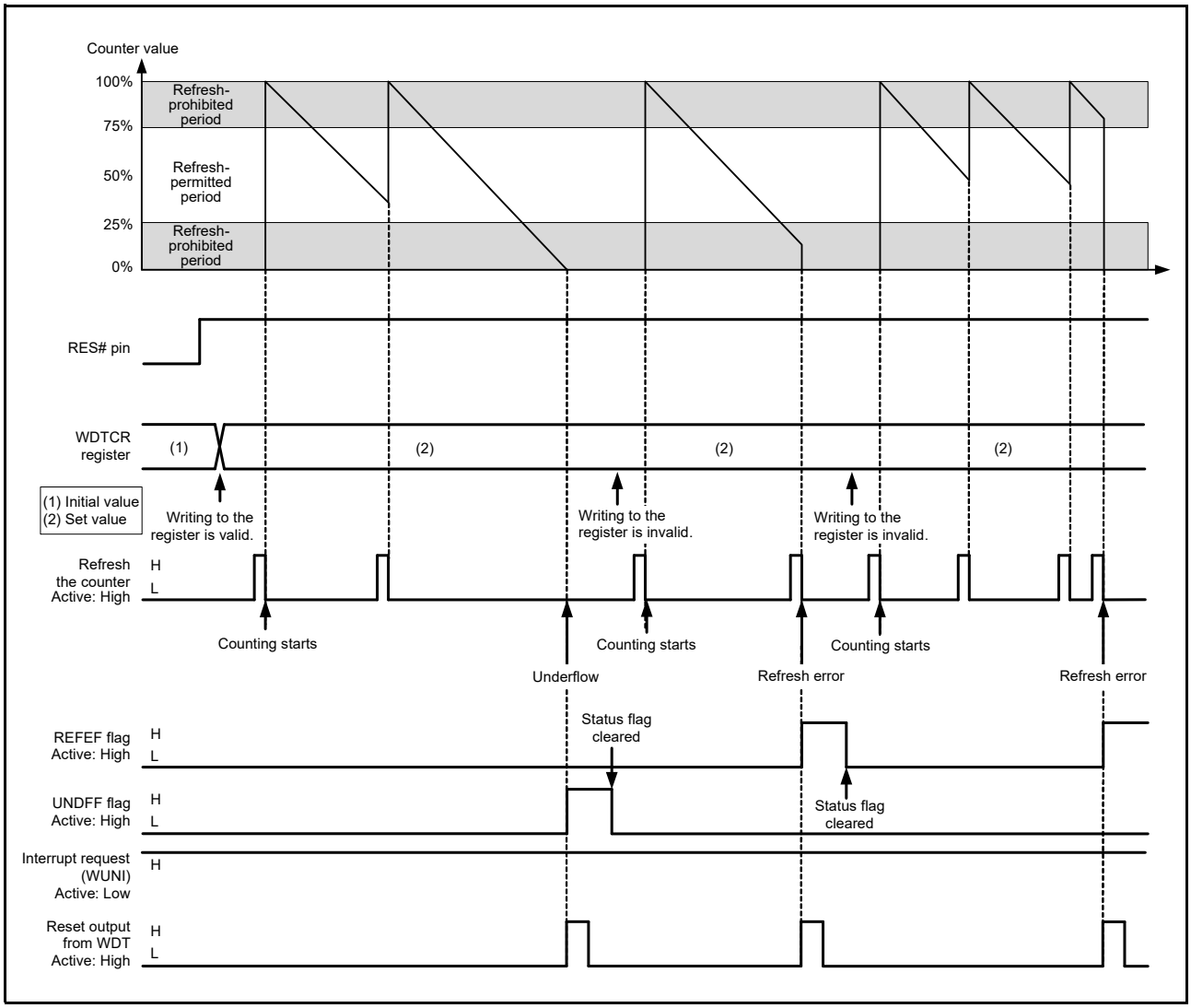


Figure 30.3 Operation Example in Register Start Mode

30.3.1.2 Auto-Start Mode

When the WDTSTRT bit in option function select register 0 (OFS0) is 0, auto-start mode is selected, the WDTCR and WDTRCR registers are disabled, and the settings in the OFS0 register are enabled.
Within the reset state, the setting values (clock division ratio, window start and end positions, timeout period, and reset output or interrupt request) of the OFS0 register are set in the WDT registers.
When the reset is released, the down-counter automatically starts counting down from the value set by the OFS0.WDTTOPS[1:0] bits.
After that, as long as the counter is refreshed in the refresh-permitted period, the value in the counter is re-set each time the counter is refreshed and counting down continues. The WDT does not output the reset signal as long as this continues.
However, if the down-counter underflows because refreshing of the down-counter is not possible due to the program having entered crashed execution or if a refresh error occurs due to refreshing outside the refresh-permitted period, the WDT outputs the reset signal or non-maskable interrupt request/interrupt request (WUNI).
After the reset signal or non-maskable interrupt request/interrupt request is output of for one cycle of counting, the value of the timeout period is set in the down-counter counting is restarted.
Reset output or interrupt request output can be selected by setting the OFS0.WDTRSTIRQS bit.

Figure 30.4 shows an example of operation (non-maskable interrupt) under the following conditions.

- Auto start mode (OFS0.WDTSTRT = 0)
- Non-maskable interrupt request output is enabled (OFS0.WDTRSTIRQS = 0)
- The window start position is 75% (OFS0.WDTRPSS[1:0] = 10b)
- The window end position is 25% (OFS0.WDTRPES[1:0] = 10b)

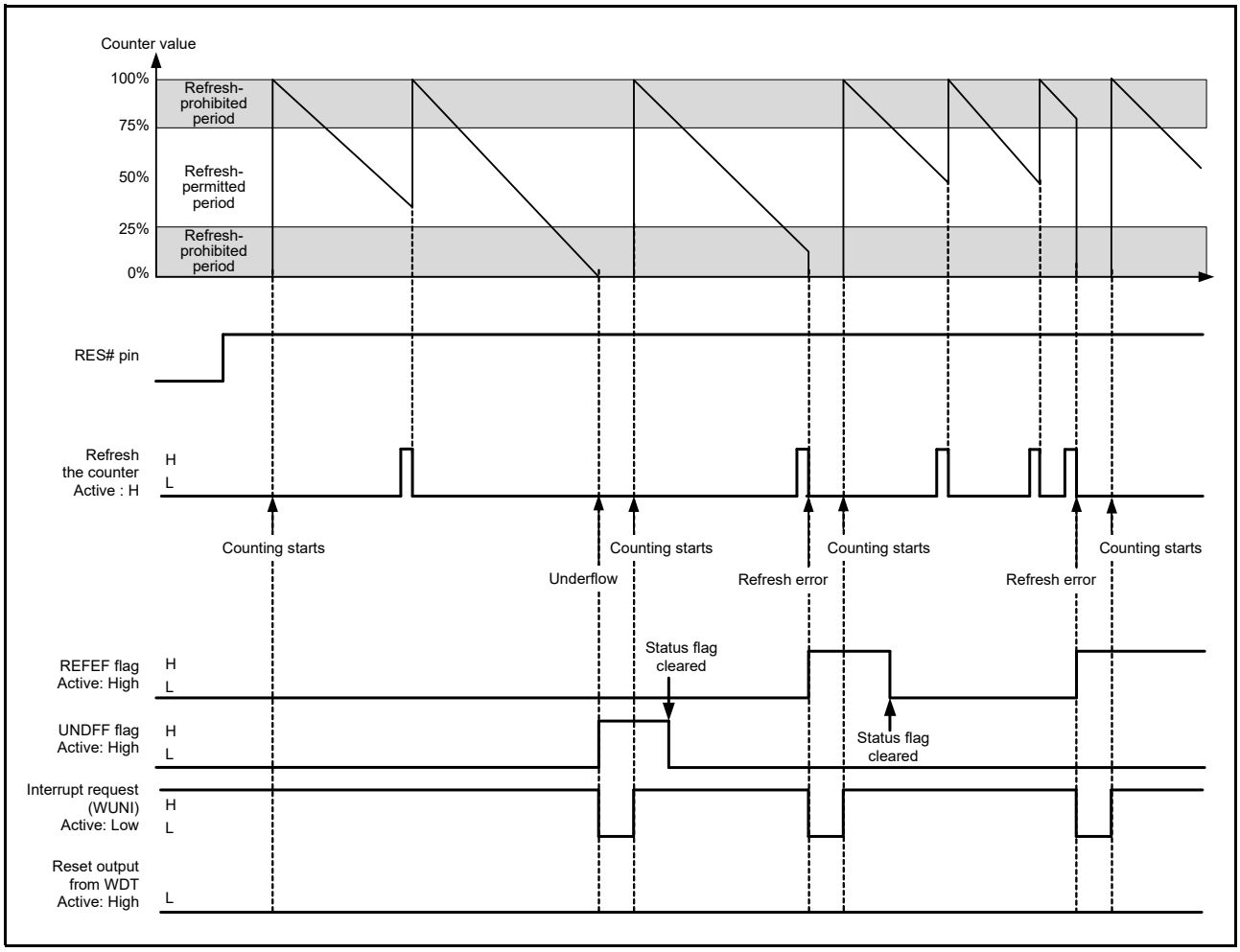


Figure 30.4 Operation Example in Auto-Start Mode

30.3.2 Control over Writing to the WDTCR and WDTRCR Registers

Writing to the WDTCR or WDTRCR register is only possible once between the release from the reset state and the first refresh operation.
After a refresh operation (counting starts) or by writing to the WDTCR or WDTRCR register, the protection signal in the WDT becomes 1 to protect the WDTCR and WDTRCR registers against subsequent attempts at writing.
This protection is released by the reset source of the WDT. With other reset sources, the protection is not released.
Figure 30.5 shows control waveforms produced in response to writing to the WDTCR register.

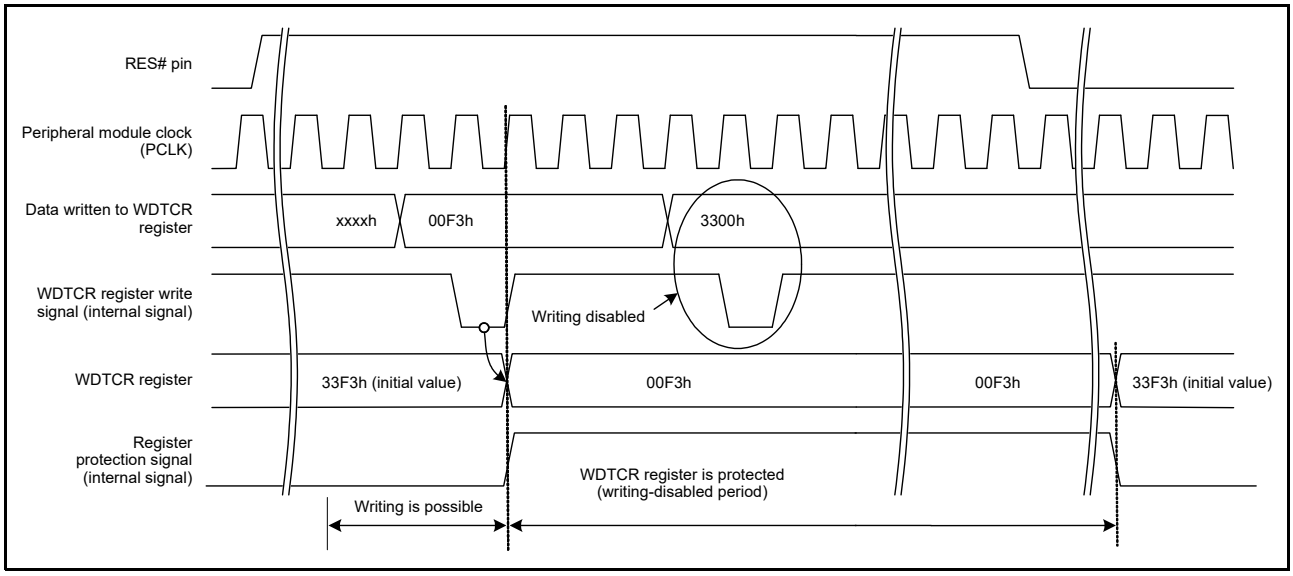


Figure 30.5 Control Waveforms Produced in Response to Writing to the WDTCR Register

30.3.3 Refresh Operation

The down-counter is refreshed by writing the values 00h and then FFh to the WDTRR register. If a value other than FFh is written after 00h, the down-counter is not refreshed. After such invalid writing, correct refreshing is performed by again writing to 00h and then FFh to the WDTRR register.
Even if a register other than the WDTRR register is accessed or the WDTRR register is read between writing 00h and writing FFh to the WDTRR register, correct refreshing will be done.
Writing to refresh the counter must be performed within the refresh-permitted period. Whether writing is done within the refresh-permitted period is determined when writing FFh. For this reason, correct refreshing will be done even if 00h is written outside the refresh-permitted period.

[Sample sequences of writing that are valid for refreshing the counter]

- 00h → FFh
- 00h (n–1-th time) → 00h (nth time) → FFh
- 00h → access to another register or read from the WDTRR register → FFh

[Sample sequences of writing that are not valid for refreshing the counter]

- 23h (a value other than 00h) → FFh
- 00h → 54h (a value other than FFh)
- 00h→ AAh (00h and a value other than FFh) → FFh

After FFh is written to the WDTRR register, refreshing the down-counter requires up to four cycles of the signal for counting. Therefore, writing FFh to the WDTRR register should be completed four-count cycles before the down-counter underflows.

Figure 30.6 shows the WDT refresh-operation waveforms when the clock division ratio = PCLK/64.

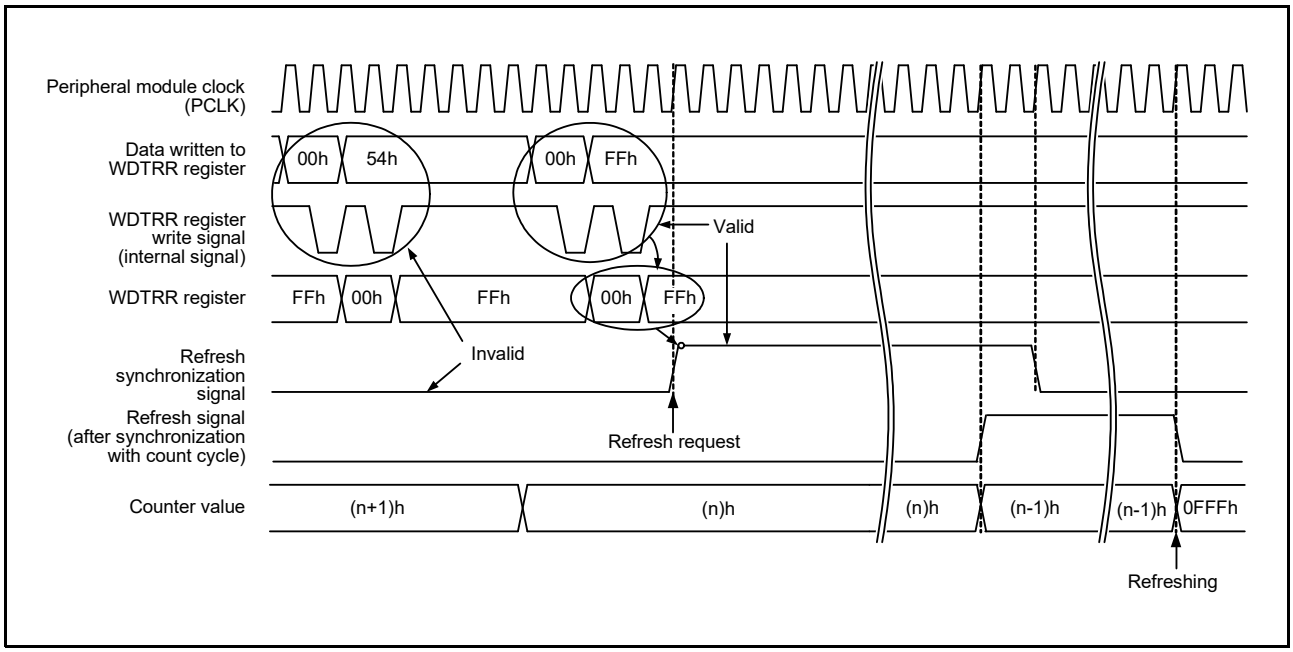


Figure 30.6 WDT Refresh Operation Waveforms (WDTCR.CKS[3:0] = 0100b, WDTCR.TOPS[1:0] = 01b)

30.3.4 Reset Output

When the WDTRCR.RSTIRQS bit is set to 1 in register start mode or when the WDTRSTIRQS bit in option function select register 0 (OFS0) is set to 1 in auto-start mode, a reset signal is output for one-count cycle when an underflow in the down-counter or a refresh error occurs.

In register start mode, the down-counter is initialized (all bits set to 0) and stopped in that state after output of the reset signal. After the reset is released and the program is restarted, the counter is set up again and counting down is started by refreshing.

In auto-start mode, counting down automatically starts after the reset is released.

30.3.5 Interrupt Source

When the WDTRCR.RSTIRQS bit is set to 0 in register start mode or when the OFS0.WDTRSTIRQS bit is set to 0 in auto-start mode, an interrupt (WUNI) signal is generated when an underflow in the counter or a refresh error occurs. This interrupt can be used as a non-maskable interrupt and an interrupt. For details, refer to section 14, Interrupt Controller (ICUG).

Table 30.4 WDT Interrupt Source

Name	Interrupt Source	DTC Activation	DMAC Activation
WUNI	Down-counter underflow Refresh error	Not possible	Not possible

30.3.6 Reading the Down-Counter Value

The WDT stores the counter value in the WDTSR.CNTVAL[13:0] bits. Thus, the counter value can be checked through the WDTSR.CNTVAL[13:0] bits.

Figure 30.7 shows the processing for reading the WDT down-counter value when the clock division ratio = PCLK/64.

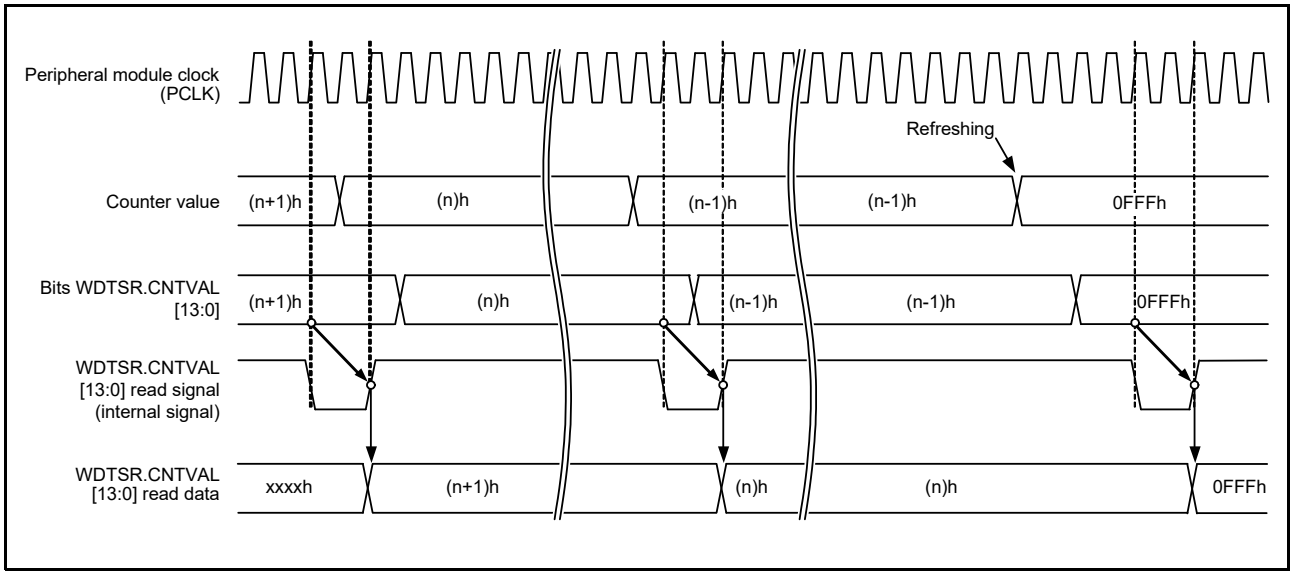


Figure 30.7 Processing for Reading WDT Down-Counter Value (WDTCR.CKS[3:0] = 0100b, WDTCR.TOPS[1:0] = 01b)

30.3.7 Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers

Table 30.5 lists the correspondence between option function select register 0 (OFS0) used in auto-start mode and the registers used in register start mode.

Do not change the OFS0 register setting during WDT operation.

For details on the OFS0 register, refer to section 7.2.3, Option Function Select Register 0 (OFS0).

Table 30.5 Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers

Target of Control	Function	OFS0 Register (Enabled in Auto-Start Mode) OFS0.WDTSTRT = 0	WDT Registers (Enabled in Register Start Mode) OFS0.WDTSTRT = 1
Down-counter	Timeout period selection	OFS0.WDTTOPS[1:0]	WDTCR.TOPS[1:0]
	Clock division ratio selection	OFS0.WDTCKS[3:0]	WDTCR.CKS[3:0]
	Window start position selection	OFS0.WDTRPSS[1:0]	WDTCR.RPSS[1:0]
	Window end position selection	OFS0.WDTRPES[1:0]	WDTCR.RPES[1:0]
Reset output or interrupt request output	Reset output or interrupt request output selection	OFS0.WDTRSTIRQS	WDTRCR.RSTIRQS

31. Independent Watchdog Timer (IWDTa)

In this section, “PCLK” is used to refer to PCLKB.

31.1 Overview

The independent watchdog timer (IWDT) can be used to detect programs being out of control.
The user can detect when a program runs out of control if an underflow occurs, by creating a program that refreshes the IWDT counter before it underflows.
The functions of the IWDT are different from those of the WDT in the following respects.

- The divided IWDT-dedicated clock (IWDTCLK) is used as the count source (not affected by the PCLK).
- When making a transition to sleep mode, software standby mode, or all-module clock stop mode, the IWDTCSTPR.SLCSTP bit or the OFS0.IWDTSLCSTP bit can be used to select whether to stop the counter or not.

Table 31.1 lists the specifications of the IWDT and Figure 31.1 shows a block diagram of the IWDT.

Table 31.1 IWDT Specifications

Item	Description
Count source*1	IWDT-dedicated clock (IWDTCLK)
Clock divide ratio	Divide by 1, 16, 32, 64, 128, or 256
Counter operation	Counting down using a 14-bit down-counter
Conditions for starting the counter	• Auto-start mode: Counting automatically starts after a reset is released • Register start mode: Counting is started by refresh operation (writing 00h and then FFh to the IWDTRR register).
Conditions for stopping the counter	• Reset • In low power consumption states (depends on the register setting*2) • A counter underflows or a refresh error occurs (only in register start mode)
Window function	Window start and end positions can be specified (refresh-permitted and refresh-prohibited periods)
Reset output sources	• Down-counter underflows • Refreshing outside the refresh-permitted period (refresh error)
Non-maskable interrupt/ interrupt sources	• Down-counter underflows • Refreshing outside the refresh-permitted period (refresh error)
Reading the counter value	The down-counter value can be read by the IWDTSR register.
Event link function (output)	• Down-counter underflow event output • Refresh error event output
Output signal (internal signal)	• Reset output • Interrupt request output • Sleep mode count stop control output
Auto-start mode (controlled by option function select register 0 (OFS0))	• Selecting the clock frequency divide ratio after a reset (OFS0.IWDTCKS[3:0] bits) • Selecting the timeout period of the independent watchdog timer (OFS0.IWDTTOPS[1:0] bits) • Selecting the window start position in the independent watchdog timer (OFS0.IWDTRPSS[1:0] bits) • Selecting the window end position in the independent watchdog timer (OFS0.IWDTRPES[1:0] bits) • Selecting the reset output or interrupt request output (OFS0.IWDTRSTIRQS bit) • Selecting the down-count stop function at transition to sleep mode, software standby mode, or all-module clock stop mode (OFS0.IWDTSLCSTP bit)
Register start mode (controlled by the IWDT registers)	• Selecting the clock frequency divide ratio after refreshing (IWDTCR.CKS[3:0] bits) • Selecting the timeout period of the independent watchdog timer (IWDTCR.TOPS[1:0] bits) • Selecting the window start position in the independent watchdog timer (IWDTCR.RPSS[1:0] bits) • Selecting the window end position in the independent watchdog timer (IWDTCR.RPES[1:0] bits) • Selecting the reset output or interrupt request output (IWDTRCR.RSTIRQS bit) • Selecting the down-count stop function at transition to sleep mode, software standby mode, or all-module clock stop mode (IWDTCSTPR.SLCSTP bit)

Note 1. Satisfy the frequency of the peripheral module clock (PCLK) ≥ 4 × (the frequency of the count source after divide).
Note 2. When the OFS0.IWDTSLCSTP bit is 1 in auto-start mode, and when the IWDTCSTPR.SLCSTP bit is 1 in register start mode.

To use the IWDT, the IWDT-dedicated clock (IWDTCLK) should be supplied so that the IWDT operates even if the peripheral module clock (PCLK) stops. The bus interface and registers operate with PCLK, and the 14-bit counter and control circuits operate with IWDTCLK.

Figure 31.1 is a block diagram of the IWDT.

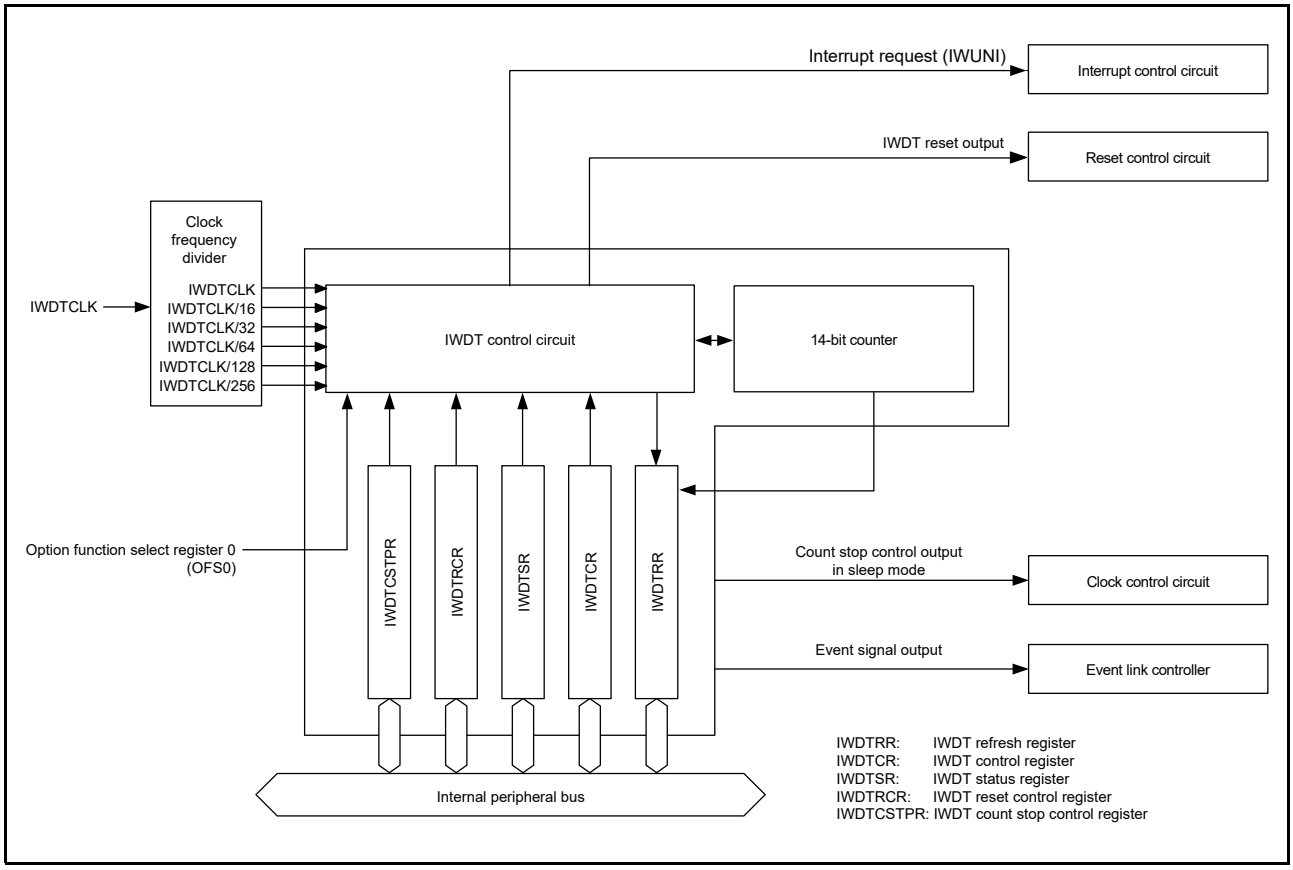
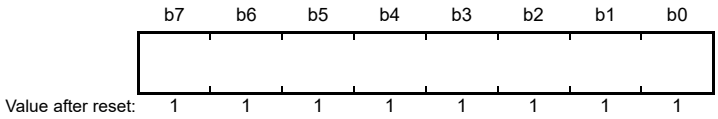


Figure 31.1 IWDT Block Diagram

31.2 Register Descriptions

31.2.1 IWDT Refresh Register (IWDTRR)

Address(es): IWDT.IWDTRR 0008 8030h



Bit	Description	R/W
b7 to b0	The counter is refreshed by writing 00h and then writing FFh to this register.	R/W

The IWDTRR register refreshes the counter of the IWDT.
The counter of the IWDT is refreshed by writing 00h and then writing FFh to the IWDTRR register (refresh operation) within the refresh-permitted period.
After the counter has been refreshed, it starts counting down from the value selected by the IWDTTOPS[1:0] bits in option function select register 0 (OFS0) in auto-start mode. In register start mode, counting down starts from the value selected by setting the IWDTCR.TOPS[1:0] bits in the first refresh operation after a reset is released.
When 00h is written, the read value is 00h. When a value other than 00h is written, the read value is FFh.
For details of the refresh operation, refer to section 31.3.3, Refresh Operation.

31.2.2 IWDT Control Register (IWDTCR)

Address(es): IWDT.IWDTCR 0008 8032h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	RPSS[1:0]		—	—	RPES[1:0]		CKS[3:0]				—	—	TOPS[1:0]	
Value after reset:	0	0	1	1	0	0	1	1	1	1	1	1	0	0	1	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TOPS[1:0]	Timeout Period Select	b1 b0 0 0: 1024 cycles (03FFh) 0 1: 4096 cycles (0FFFh) 1 0: 8192 cycles (1FFFh) 1 1: 16384 cycles (3FFFh)	R/W
b3, b2	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b7 to b4	CKS[3:0]	Clock Divide Ratio Select	b7 b4 0 0 0 0: No division 0 0 1 0: Divide-by-16 0 0 1 1: Divide-by-32 0 1 0 0: Divide-by-64 1 1 1 1: Divide-by-128 0 1 0 1: Divide-by-256 Other settings are prohibited.	R/W
b9, b8	RPES[1:0]	Window End Position Select	b9 b8 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (window end position is not specified.)	R/W
b11, b10	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b13, b12	RPSS[1:0]	Window Start Position Select	b13 b12 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (window start position is not specified.)	R/W
b15, b14	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R

There are some restrictions on writing to the IWDTCR register. For details, refer to section 31.3.2, Control over Writing to the IWDTCR, IWDTRCR, and IWDTCSTPR Registers.

In auto-start mode, the settings in the IWDTCR register are disabled, and the settings in option function select register 0 (OFS0) are enabled. The bit setting made to the IWDTCR register can also be made in the OFS0 register. For details, refer to section 31.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

TOPS[1:0] Bits (Timeout Period Select)

These bits select the timeout period (period until the counter underflows) from among 1024, 4096, 8196, or 16384 cycles, taking the divided clock specified by the CKS[3:0] bits as one cycle.
After the counter is refreshed, the combination of the CKS[3:0] and TOPS[1:0] bits determines the time (number of IWDTCLK cycles) until the counter underflows.
Relations between the CKS[3:0] and TOPS[1:0] bit setting, the timeout period, and the number of IWDTCLK cycles are listed in Table 31.2.

Table 31.2 Settings and Timeout Periods

CKS[3:0] Bits				TOPS[1:0] Bits				
b7	b6	b5	b4	b1	b0	Clock Divide Ratio	Timeout Period (Number of Cycles)	Cycles of IWDTCLK
0	0	0	0	0	0	No division	1024	1024
				0	1		4096	4096
				1	0		8192	8192
				1	1		16384	16384
0	0	1	0	0	0	Divide-by-16	1024	16384
				0	1		4096	65536
				1	0		8192	131072
				1	1		16384	262144
0	0	1	1	0	0	Divide-by-32	1024	32768
				0	1		4096	131072
				1	0		8192	262144
				1	1		16384	524288
0	1	0	0	0	0	Divide-by-64	1024	65536
				0	1		4096	262144
				1	0		8192	524288
				1	1		16384	1048576
1	1	1	1	0	0	Divide-by-128	1024	131072
				0	1		4096	524288
				1	0		8192	1048576
				1	1		16384	2097152
0	1	0	1	0	0	Divide-by-256	1024	262144
				0	1		4096	1048576
				1	0		8192	2097152
				1	1		16384	4194304

CKS[3:0] Bits (Clock Divide Ratio Select)

These bits select the IWDTCLK clock divide ratio from among divide-by 1, 16, 32, 64, 128, and 256. Combination with the TOPS[1:0] bit setting, a count period between 1024 and 4194304 cycles of the IWDTCLK clock can be selected for the IWDT.

RPES[1:0] Bits (Window End Position Select)

These bits select 75%, 50%, 25% or 0% of the count period for the window end position of the counter. The window end position should be a value smaller than the window start position (window start position > window end position). If the window end position is greater than the window start position, only the window start position setting is enabled.
The counter values for the window start and end positions selected by setting the RPSS[1:0] and RPES[1:0] bits change depending on the TOPS[1:0] bit setting.
Table 31.3 lists the counter values for the window start and end positions corresponding to TOPS[1:0] bit values.

Table 31.3 Relationship between Timeout Period and Window Start and End Counter Values

TOPS[1:0] Bits		Timeout Period		Window Start and End Counter Value			
b1	b0	Cycles	Counter Value	100%	75%	50%	25%
0	0	1024	03FFh	03FFh	02FFh	01FFh	00FFh
0	1	4096	0FFFh	0FFFh	0BFFh	07FFh	03FFh
1	0	8192	1FFFh	1FFFh	17FFh	0FFFh	07FFh
1	1	16384	3FFFh	3FFFh	2FFFh	1FFFh	0FFFh

RPSS[1:0] Bits (Window Start Position Select)

These bits select a counter window start position from 100%, 75%, 50%, or 25% of the count period (100% when the count starts and 0% when the counter underflows). The interval between the window start position and window end position is the refresh-permitted period and the other periods are refresh-prohibited periods.
Figure 31.2 shows the relationship between of the RPSS[1:0] and RPES[1:0] bit setting and the refresh-permitted and refresh-prohibited periods.

RPSS[1:0] Bits		RPES[1:0] Bits		Window	
b13	b12	b9	b8	Start (%)	End (%)
1	1	1	1	100	0
		1	0		25
		0	1		50
		0	0		75
1	0	1	1	75	0
		1	0		25
		0	1		50
		0	0		75
0	1	1	1	50	0
		1	0		25
		0	1		50
		0	0		75
0	0	1	1	25	0
		1	0		25
		0	1		50
		0	0		75

Note: If window end setting is greater than or equal to window start setting, the window end setting is set to 0%.

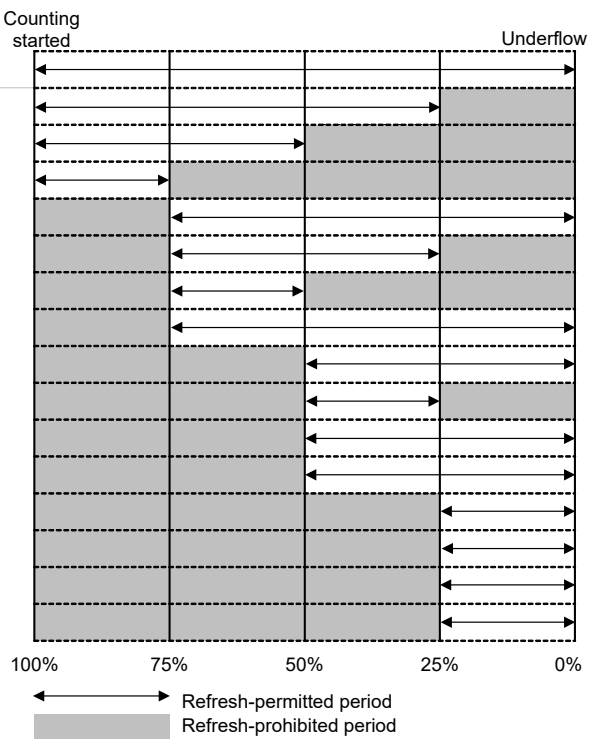


Figure 31.2 RPSS[1:0] and RPES[1:0] Bit Settings and the Refresh-Permitted Period

31.2.3 IWDT Status Register (IWDTSR)

Address(es): IWDT.IWDTSR 0008 8034h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	REFEF	UNDFF	CNTVAL[13:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b13 to b0	CNTVAL[13:0]	Counter Value	Value counted by the counter	R
b14	UNDFF	Underflow Flag	0: No underflow occurred 1: Underflow occurred	R/(W) *1
b15	REFEF	Refresh Error Flag	0: No refresh error occurred 1: Refresh error occurred	R/(W) *1

Note 1. Only 0 can be written to clear the flag.

CNTVAL[13:0] Bits (Counter Value)

These bits are used to confirm the counter value of the counter, but note that the read value may differ from the actual count by a value of one count.

UNDFF Flag (Underflow Flag)

This bit is used to confirm whether or not an underflow has occurred in the counter.
The value 1 indicates that the counter has underflowed. The value 0 indicates that the counter has not underflowed.
Write 0 to the UNDFF flag to set the value to 0. Writing 1 has no effect.

REFEF Flag (Refresh Error Flag)

This bit is used to confirm whether or not a refresh error (performing a refresh operation during a refresh-prohibited period).
The value 1 indicates that a refresh error has occurred. The value 0 indicates that no refresh error has occurred.
Write 0 to the REFEF flag to set the value to 0. Writing 1 has no effect.

31.2.4 IWDT Reset Control Register (IWDTRCR)

Address(es): IWDT.IWDTRCR 0008 8036h

	b7	b6	b5	b4	b3	b2	b1	b0
	RSTIRQS	—	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b7	RSTIRQS	Reset Interrupt Request Select	0: Non-maskable interrupt request or interrupt request output is enabled. 1: Reset output is enabled.	R/W

There are some restrictions on writing to the IWDTRCR register. For details, refer to section 31.3.2, Control over Writing to the IWDTCR, IWDTRCR, and IWDTCSTPR Registers.

In auto-start mode, the IWDTRCR register setting are disabled, and the settings in option function select register 0 (OFS0) are enabled. The bit setting mode to the IWDTRCR register can also be made in the OFS0 register. For details, refer to section 31.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

31.2.5 IWDT Count Stop Control Register (IWDTCSTPR)

Address(es): IWDT.IWDTCSTPR 0008 8038h

	b7	b6	b5	b4	b3	b2	b1	b0
	SLCSTP	—	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b7	SLCSTP	Sleep Mode Count Stop Control	0: Count stop is disabled. 1: Count is stopped at a transition to sleep mode, software standby mode, or all-module clock stop mode.	R/W

The IWDTCSTPR register controls whether to stop the IWDT counter in a low power consumption state. There are some restrictions on writing to the IWDTCSTPR register. For details, refer to section 31.3.2, Control over Writing to the IWDTCR, IWDTRCR, and IWDTCSTPR Registers.
In auto-start mode, the IWDTCSTPR register setting are disabled, and the settings in option function select register 0 (OFS0) are enabled. The bit setting mode to the IWDTCSTPR register can also be made in the OFS0 register. For details, refer to section 31.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

SLCSTP Bit (Sleep Mode Count Stop Control)

This bit selects whether to stop counting at a transition to sleep mode, software standby mode, or all-module clock stop mode.

31.2.6 Option Function Select Register 0 (OFS0)

For option function select register 0 (OFS0), refer to section 31.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

31.3 Operation

31.3.1 Count Operation in Each Start Mode

Select the IWDT start mode by setting the IWDTSTRT bit in option function select register 0 (OFS0).
When the OFS0.IWDTSTRT bit is 1 (register start mode), the IWDTCR, IWDTRCR, and IWDTCSTPR registers are enabled, and counting is started by refresh operation (writing) to the IWDTRR register. When the OFS0.IWDTSTRT bit is 0 (auto-start mode), the setting of the OFS0 register is enabled, and counting automatically starts after reset.

31.3.1.1 Register Start Mode

When the OFS0.IWDTSTRT bit in option function select register 0 is 1, register start mode is selected, and the IWDTCR, IWDTRCR, and IWDTCSTPR registers are enabled.
After a reset is released, set the clock divide ratio, window start and end positions, and timeout period in the IWDTCR register, the reset output or interrupt request output in the IWDTRCR register, and the counter stop control at transitions to low power consumption states in the IWDTCSTPR register. Then refresh the counter to start counting down from the value selected by setting the IWDTCR.TOPS[1:0] bits.
Thereafter, as long as the program continues normal operation and the counter is refreshed in the refresh-permitted period, the value in the counter is re-set each time the counter is refreshed and counting down continues. The IWDT does not output the reset signal as long as this continues. However, if the counter underflows because the counter cannot be refreshed due to a program runaway, or if a refresh error occurs because the counter was refreshed outside the refresh-permitted period, the IWDT outputs a reset signal or a non-maskable interrupt request/interrupt request (IWUNI). Set the IWDTRCR.RSTIRQS bit to select either reset output or interrupt request output.

Figure 31.3 shows an example of operation under the following conditions.

- Register start mode (OFS0.IWDTSTRT = 1)
- Reset output is enabled (IWDTRCR.RSTIRQS = 1)
- The window start position is 75% (IWDTCR.RPSS[1:0] = 10b)
- The window end position is 25% (IWDTCR.RPES[1:0] = 10b)

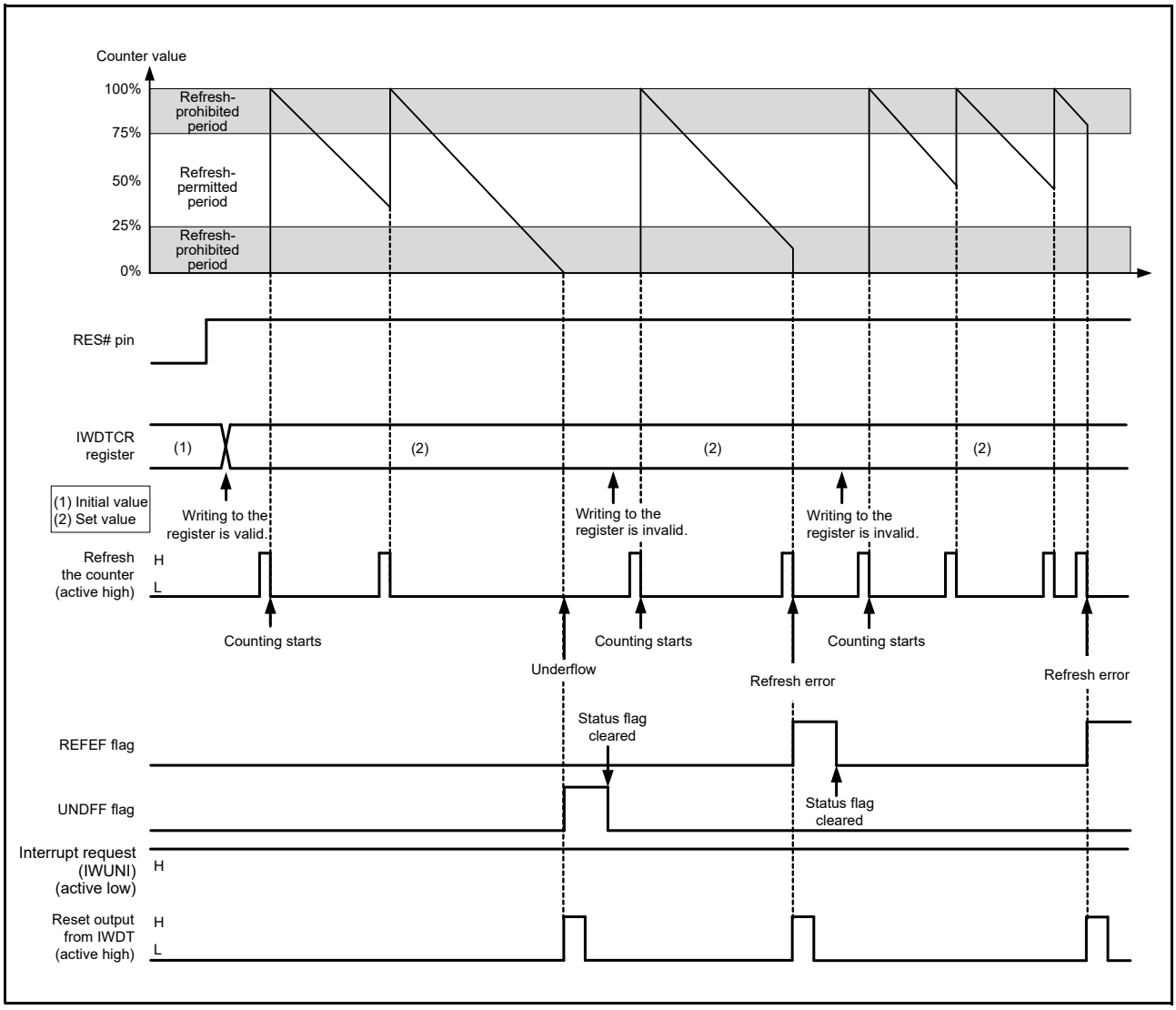


Figure 31.3 Operation Example in Register Start Mode

31.3.1.2 Auto-Start Mode

When the IWDTSTRT bit in option function select register 0 (OFS0) is 0, auto-start mode is selected, and the IWDTCR, IWDTRCR, and IWDTCSTPR registers are disabled.
Within the reset state, the clock divide ratio, window start and end positions, timeout period, reset output or interrupt request output, and counter stop control at transitions to low power consumption states are set using the values specified in the OFS0 register. When the reset is released, the counter automatically starts counting down from the value selected by the OFS0.IWDTTOPS[1:0] bits.
After that, as long as the program continues normal operation and the counter is refreshed in the refresh-permitted period, the value in the counter is re-set each time the counter is refreshed and counting down continues. The IWDT does not output the reset signal as long as this continues. However, if the counter underflows because refreshing of the counter is not possible due to the program having entered crashed execution or if a refresh error occurs due to refreshing outside the refresh-permitted period, the IWDT outputs the reset signal or non-maskable interrupt request/interrupt request (IWUNI). After the reset signal or non-maskable interrupt request/interrupt request (IWUNI) is generated, the counter reloads the timeout period after counting for one cycle, and restarts counting. Set the OFS0.IWDTRSTIRQS bit to select either reset output or interrupt request output.

Figure 31.4 shows an example of operation under the following conditions.

- Auto-start mode (OFS0.IWDTSTRT = 0)
- Non-maskable interrupt request output is enabled (OFS0.IWDTRSTIRQS = 0)
- The window start position is 75% (OFS0.IWDTRPSS[1:0] = 10b)
- The window end position is 25% (OFS0.IWDTRPES[1:0] = 10b)

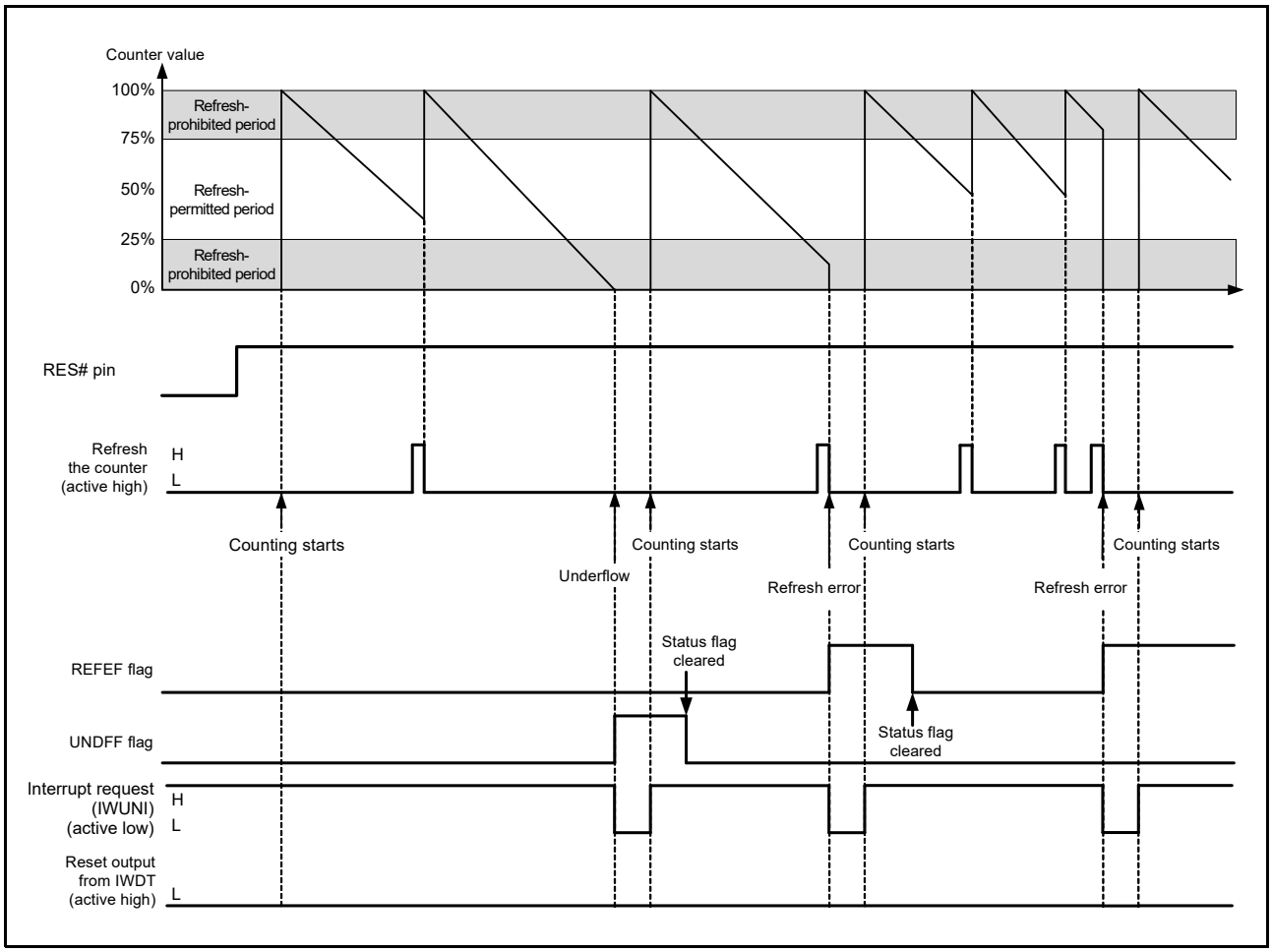


Figure 31.4 Operation Example in Auto-Start Mode

31.3.2 Control over Writing to the IWDTCR, IWDTRCR, and IWDTCSTPR Registers

Writing to the IWDTCR, IWDTRCR, or IWDTCSTPR register is only possible once between the release from the reset state and the first refresh operation.

After a refresh operation (counting starts) or the IWDTCR, IWDTRCR, or IWDTCSTPR register is written to, the protection signal in the IWDT becomes 1 to protect registers IWDTCR, IWDTRCR, and IWDTCSTPR against subsequent attempts at writing.

This protection is released by the reset source of the IWDT. With other reset sources, the protection is not released.

Figure 31.5 shows control waveforms produced in response to writing to the IWDTCR register.

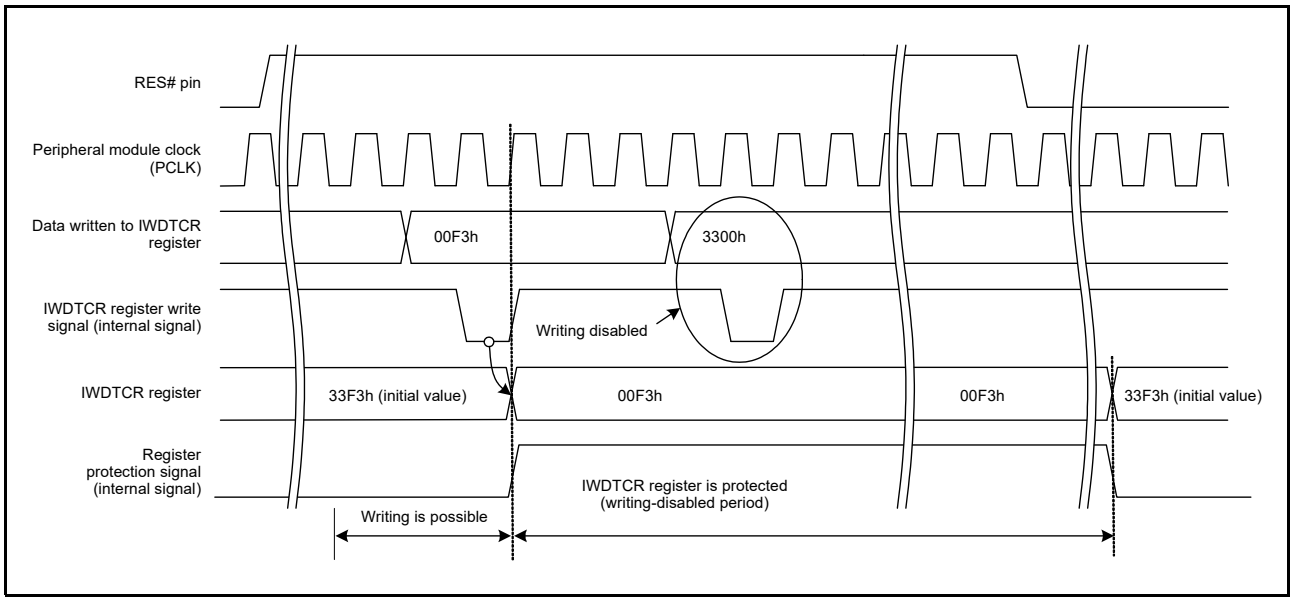


Figure 31.5 Control Waveforms Produced in Response to Writing to the IWDTCR Register

31.3.3 Refresh Operation

The counter is refreshed and starts operation (counting is started by refreshing) by writing the values 00h and then FFh to the IWDTRR register. If a value other than FFh is written after 00h, the counter is not refreshed. After such invalid writing, correct refreshing is performed by again writing 00h and then FFh to the IWDTRR register.
When writing is done in the order of 00h (first time) → 00h (second time), and if FFh is written after that, the writing order 00h → FFh is satisfied; writing 00h (n–1-th time) → 00h (nth time) → FFh is valid and correct refreshing will be done. Even when the first value written before 00h is not 00h, correct refreshing will be done if the operation contains the set of writing 00h → FFh. Moreover, even if a register other than the IWDTRR register is accessed or the IWDTRR register is read between writing 00h and writing FFh to the IWDTRR register, correct refreshing will be done.

[Sample sequences of writing that are valid for refreshing the counter]

- 00h → FFh
- 00h (n–1-th time) → 00h (nth time) → FFh
- 00h → access to another register or read from the IWDTRR register → FFh

[Sample sequences of writing that are not valid for refreshing the counter]

- 23h (a value other than 00h) → FFh
- 00h → 54h (a value other than FFh)
- 00h → AAh (00h and a value other than FFh) → FFh

Even when 00h is written to the IWDTRR register outside the refresh-permitted period, if FFh is written to the IWDTRR register in the refresh-permitted period, the writing sequence is valid and refreshing will be done.
After FFh is written to the IWDTRR register, refreshing the counter requires up to four cycles of the signal for counting (the IWDTCR.CKS[3:0] bits determine how many cycles of the IWDT-dedicated clock (IWDTCLK) make up one cycle for counting). Therefore, writing FFh to the IWDTRR register should be completed four-count cycles before the end position of the refresh-permitted period or a counter underflow. The value of the counter can be checked by the IWDTSR.CNTVAL[13:0] bits.

[Sample refreshing timings]

- When the window start position is set to 1FFFh, even if 00h is written to the IWDTRR register before 1FFFh is reached (2002h, for example), refreshing is done if FFh is written to the IWDTRR register after the value of the IWDTSR.CNTVAL[13:0] bits has reached 1FFFh.
- When the window end position is set to 1FFFh, refreshing is done if 2003h (four-count cycles before 1FFFh) or a greater value is read from the IWDTSR.CNTVAL[13:0] bits immediately after writing 00h → FFh to the IWDTRR register.
- When the refresh-permitted period continues until count 0000h, refreshing can be done immediately before an underflow. In this case, if 0003h (four-count cycles before an underflow) or a greater value is read from the IWDTSR.CNTVAL[13:0] bits immediately after writing 00h → FFh to the IWDTRR register, no underflow occurs and refreshing is done.

Figure 31.6 shows the IWDT refresh-operation waveforms when PCLK > IWDTCLK and clock divide ratio = IWDTCLK.

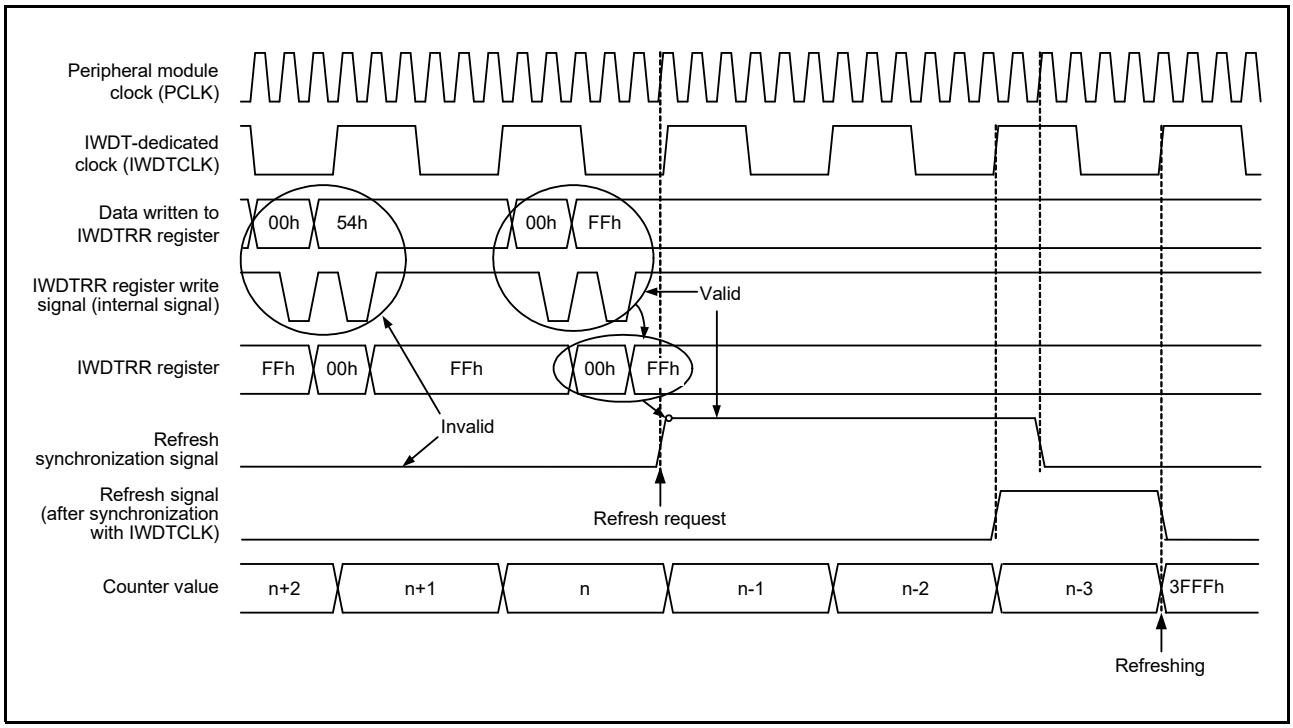


Figure 31.6 IWDT Refresh Operation Waveforms (IWDTCR.CKS[3:0] = 0000b, IWDTCR.TOPS[1:0] = 11b)

31.3.4 Status Flags

The IWDTSR.REFEF and IWDTSR.UNDFF flags retain the source of the reset signal output from the IWDT or the source of the interrupt request from the IWDT.
Thus, after release from the reset state or interrupt request generation, read the IWDTSR.REFEF and IWDTSR.UNDFF flags to check for the reset or interrupt source.
For each flag, writing 0 clears the bit and writing 1 has no effect.
Leaving the status flags unchanged does not affect operation. If the flags are not cleared, at the time of the next reset or interrupt request from the IWDT, the earlier reset or interrupt source is cleared and the new reset or interrupt source is written.
After 0 is written to each flag, up to three IWDTCLK cycles and two PCLK cycles are required before the value is reflected.

31.3.5 Reset Output

When the IWDTRCR.RSTIRQS bit is set to 1 in register start mode or when the IWDTRSTIRQS bit in option function select register 0 (OFS0) is set to 1 in auto-start mode, a reset signal is output when an underflow in the counter or a refresh error occurs.
In register start mode, the counter is initialized (0000h) and kept in that state after assertion of the reset signal. After the reset is released and the program is restarted, the counter is set up again and counting down is started by refreshing.
In auto-start mode, counting down automatically starts after the reset output.

31.3.6 Interrupt Sources

When the IWDTRCR.RSTIRQS bit is set to 0 in register start mode or when the OFS0.IWDTRSTIRQS bit is set to 0 in auto-start mode, an interrupt (IWUNI) signal is output when an underflow in the counter or a refresh error occurs. This interrupt can be used as a non-maskable interrupt or a maskable interrupt. For details, refer to section 14, Interrupt Controller (ICUG).

Table 31.4 IWDT Interrupt Source

Name	Interrupt Source	DTC Activation	DMAC Activation
IWUNI	Counter underflow Refresh error	Not possible	Not possible

31.3.7 Reading the Counter Value

As the counter in IWDT-dedicated clock (IWDTCLK), the counter value cannot be read directly. The IWDT synchronizes the counter value with the peripheral module clock (PCLK) and stores it in the IWDTSR.CNTVAL[13:0] bits. Thus, the counter value can be checked indirectly through the IWDTSR.CNTVAL[13:0] bits.
Reading the counter value requires multiple PCLK clock cycles (up to four clock cycles), and the read counter value may differ from the actual counter value by a value of one count.
Figure 31.7 shows the processing for reading the IWDT counter value when PCLK > IWDTCLK and clock divide ratio = IWDTCLK.

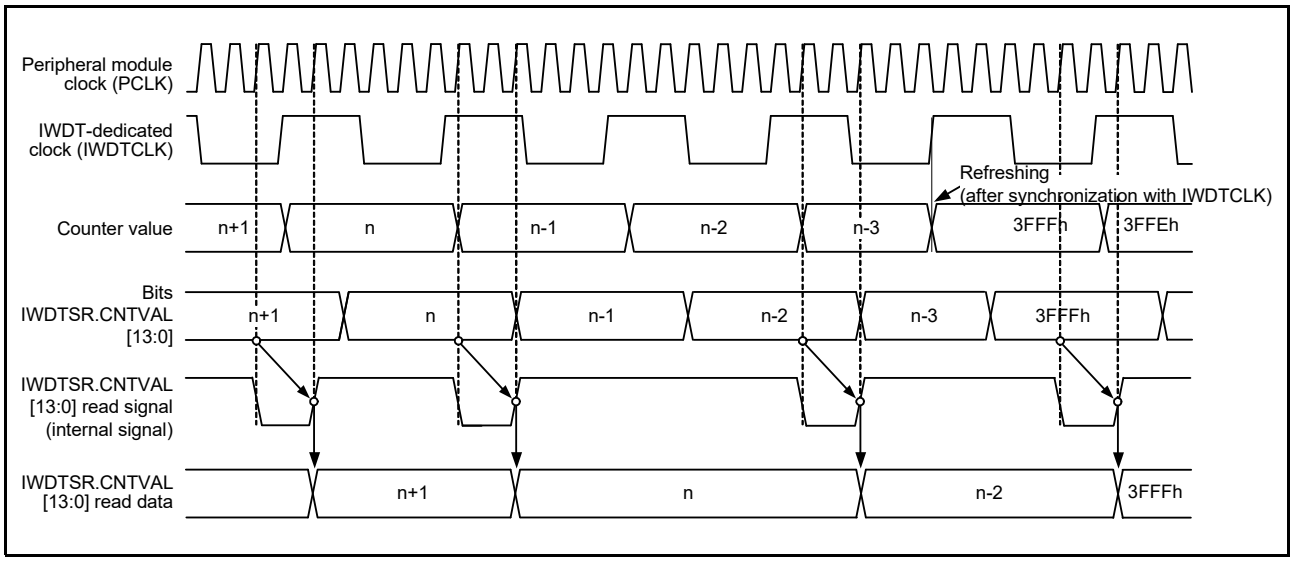


Figure 31.7 Processing for Reading IWDT Counter Value (IWDTCR.CKS[3:0] = 0000b, IWDTCR.TOPS[1:0] = 11b)

31.3.8 Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers

Table 31.5 lists the correspondence between option function select register 0 (OFS0) used in auto-start mode and the registers used in register start mode.
Do not change the OFS0 register setting during IWDT operation.
For details on the OFS0 register, refer to section 7.2.3, Option Function Select Register 0 (OFS0).

Table 31.5 Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers

Target of Control	Function	OFS0 Register (Enabled in Auto-Start Mode) OFS0.IWDTSTRT = 0	IWDT Registers (Enabled in Register Start Mode) OFS0.IWDTSTRT = 1
Counter	Timeout period selection	OFS0.IWDTTOPS[1:0]	IWDTCR.TOPS[1:0]
	Clock frequency divide ratio selection	OFS0.IWDTCKS[3:0]	IWDTCR.CKS[3:0]
	Window start position selection	OFS0.IWDTRPSS[1:0]	IWDTCR.RPSS[1:0]
	Window end position selection	OFS0.IWDTRPES[1:0]	IWDTCR.RPES[1:0]
Reset output or interrupt request output	Reset output or interrupt request output selection	OFS0.IWDTRSTIRQS	IWDTRCR.RSTIRQS
Count stop	Sleep mode count stop control	OFS0.IWDTSLCSTP	IWDTCSTPR.SLCSTP

31.4 Link Operation by ELC

The event link controller (ELC) can use the interrupt request signal generated by the IWDT as the event signal. Therefore, the ELC generates an event to the module specified previously when the IWDT outputs an interrupt request. The event signal is output by the counter underflow and refresh error.
An event signal is output regardless of the setting of the IWDTRCR.RSTIRQS bit in register start mode or the OFS0.IWDTRSTIRQS bit in auto-start mode. An event signal can also be output upon generation of the next interrupt source while the IWDTSR.REFEF or IWDTSR.UNDFF flag is 1.
For details, see section 19, Event Link Controller (ELC).

31.5 Usage Notes

31.5.1 Refresh Operations

When making the settings to control the timing of refreshing, consider variations in the range of errors due to the accuracy of the PCLK and IWDTCLK and set values which ensure that refreshing is possible.

31.5.2 Clock Divide Ratio Setting

Satisfy the frequency of the peripheral module clock (PCLK) ≥ 4 × (the frequency of the count source after divide).

32. Serial Communications Interface (SCIk, SCIh)

This MCU has four independent serial communications interface (SCI) channels. The SCI consists of the SCIk module (SCI1, SCI5, and SCI6) and the SCIh module (SCI12).
The SCIk module (SCI1, SCI5, and SCI6) can handle both asynchronous and clock synchronous serial communications. Asynchronous serial data communications can be carried out with standard asynchronous communications chips such as a Universal Asynchronous Receiver/Transmitter (UART) or Asynchronous Communications Interface Adapter (ACIA). As an extended function in asynchronous communications mode, the SCI also supports smart card (IC card) interfaces conforming to ISO/IEC 7816-3 (standard for Identification Cards). The SCI is also supports simple SPI interfaces, and simple I^2C-bus interfaces when configured for single-master systems.
The SCIh module includes the functions of the SCIg module, and supports an extended serial communication protocol formed of Start Frames and Information Frames.
In this section, “PCLK” is used to refer to PCLKB.

32.1 Overview

Table 32.1 lists the specifications of the SCIk module, Table 32.2 lists the specifications of the SCIh module, and Table 32.3 lists the specifications of the individual SCI channels.
Figure 32.1 shows the block diagram of SCI1, Figure 32.2 shows the block diagram of SCI5 and SCI6, and Figure 32.3 shows the block diagram of SCI12 (SCIh).

Table 32.1 SCIk Specifications (1/2)

Item	Description
Serial communication modes	• Asynchronous • Clock synchronous • Smart card interface • Simple I^2C-bus • Simple SPI bus
Transfer speed	Bit rate specifiable with the on-chip baud rate generator.
Full-duplex communications	Transmitter: Continuous transmission possible using double-buffer structure. Receiver: Continuous reception possible using double-buffer structure.
I/O pins	Refer to Table 32.4 to Table 32.6.
Data transfer	Selectable as LSB first or MSB first transfer*1
I/O signal level inverting function	Input signal and output signal can be inverted independently.
RXD input signal select function (supported by SCI5 only)	If the RXD signal is attenuated by transmission line effects, it can be improved by using a comparator instead of a receiver.
Interrupt sources	Transmit end, transmit data empty, receive data full, receive error, and data match Completion of generation of a start condition, restart condition, or stop condition (for simple I^2C mode)
Low power consumption function	Module stop state can be set for each channel.

Table 32.1 SCIk Specifications (2/2)

Item		Description
Asynchronous mode	Data length	7, 8, or 9 bits
	Transmission stop bit	1 or 2 bits
	Parity	Even parity, odd parity, or no parity
	Receive error detection	Parity, overrun, and framing errors
	Hardware flow control	CTSn# and RTSn# pins can be used in controlling transmission/reception.
	Data match detection	Compares receive data and comparison data, and generates interrupt when they are matched.
	Start-bit detection	Low level or falling edge is selectable.
	Receive data sampling timing adjustment	Sampling point of receive data can be changed to front or rear of center of bit.
	Transmit signal transition timing adjustment	Falling or rising edge of the transmit data can be delayed.
	Break detection	When a framing error occurs, a break can be detected by reading the RXDn pin level directly or reading the SPTR.RXDMON flag.
	Clock source	An internal or external clock can be selected. Transfer rate clock input from the TMR can be used. (SCI5 and SCI6)
	Double-speed mode	Baud rate generator double-speed mode is selectable.
	Multi-processor communications function	Serial communication among multiple processors
	Noise cancellation	The signal paths from input on the RXDn pins incorporate digital noise filters.
Clock synchronous mode	Data length	8 bits
	Receive error detection	Overrun error
	Hardware flow control	CTSn# and RTSn# pins can be used in controlling transmission/reception.
Smart card interface mode	Error processing	An error signal can be automatically transmitted when detecting a parity error during reception
		Data can be automatically retransmitted when receiving an error signal during transmission
	Data type	Both direct convention and inverse convention are supported.
Simple I^2C mode	Transfer format	I^2C-bus format
	Operating mode	Master (single-master operation only)
	Transfer rate	Fast mode is supported (refer to section 32.2.11, Bit Rate Register (BRR) to set the transfer rate).
	Noise cancellation	The signal paths from input on the SSCLn and SSDAn pins incorporate digital noise filters, and the interval for noise cancellation is adjustable.
Simple SPI bus	Data length	8 bits
	Detection of errors	Overrun error
	SS input pin function	Applying the high level to the SSn# pin can cause the output pins to enter the high-impedance state.
	Clock settings	Four kinds of settings for clock phase and clock polarity are selectable.
Bit rate modulation function		Correction of outputs from the on-chip baud rate generator can reduce errors.
Event link function (supported by SCI5 only)		Error (receive error or error signal detection) event output
		Receive data full event output
		Transmit data empty event output
		Transmit end event output

Note 1. In simple I^2C mode, only MSB first is available.

Table 32.2 SCIh Specifications (1/2)

Item		Description
Serial communication modes		• Asynchronous • Clock synchronous • Smart card interface • Simple I²C-bus • Simple SPI bus
Transfer speed		Bit rate specifiable with the on-chip baud rate generator.
Full-duplex communications		Transmitter: Continuous transmission possible using double-buffer structure. Receiver: Continuous reception possible using double-buffer structure.
I/O pins		Refer to Table 32.4 to Table 32.7.
Data transfer		Selectable as LSB first or MSB first transfer*1
Interrupt sources		Transmit end, transmit data empty, receive data full, and receive error Completion of generation of a start condition, restart condition, or stop condition (for simple I²C mode)
Low power consumption function		Module stop state can be set.
Asynchronous mode	Data length	7, 8, or 9 bits
	Transmission stop bit	1 or 2 bits
	Parity	Even parity, odd parity, or no parity
	Receive error detection	Parity, overrun, and framing errors
	Hardware flow control	CTSn# and RTSn# pins can be used in controlling transmission/reception.
	Start-bit detection	Low level or falling edge is selectable.
	Break detection	When a framing error occurs, a break can be detected by reading the RXDn pin level directly.
	Clock source	An internal or external clock can be selected. Transfer rate clock input from the TMR can be used.
	Double-speed mode	Baud rate generator double-speed mode is selectable.
	Multi-processor communications function	Serial communication among multiple processors
	Noise cancellation	The signal paths from input on the RXDn pins incorporate digital noise filters.
Clock synchronous mode	Data length	8 bits
	Receive error detection	Overrun error
	Hardware flow control	CTSn# and RTSn# pins can be used in controlling transmission/reception.
Smart card interface mode	Error processing	An error signal can be automatically transmitted when detecting a parity error during reception
		Data can be automatically retransmitted when receiving an error signal during transmission
	Data type	Both direct convention and inverse convention are supported.
Simple I²C mode	Transfer format	I²C-bus format
	Operating mode	Master (single-master operation only)
	Transfer rate	Fast mode is supported (refer to section 32.2.11, Bit Rate Register (BRR) to set the transfer rate).
	Noise cancellation	The signal paths from input on the SSCLn and SSDAn pins incorporate digital noise filters, and the interval for noise cancellation is adjustable.
Simple SPI bus	Data length	8 bits
	Detection of errors	Overrun error
	SS input pin function	Applying the high level to the SSn# pin can cause the output pins to enter the high-impedance state.
	Clock settings	Four kinds of settings for clock phase and clock polarity are selectable.

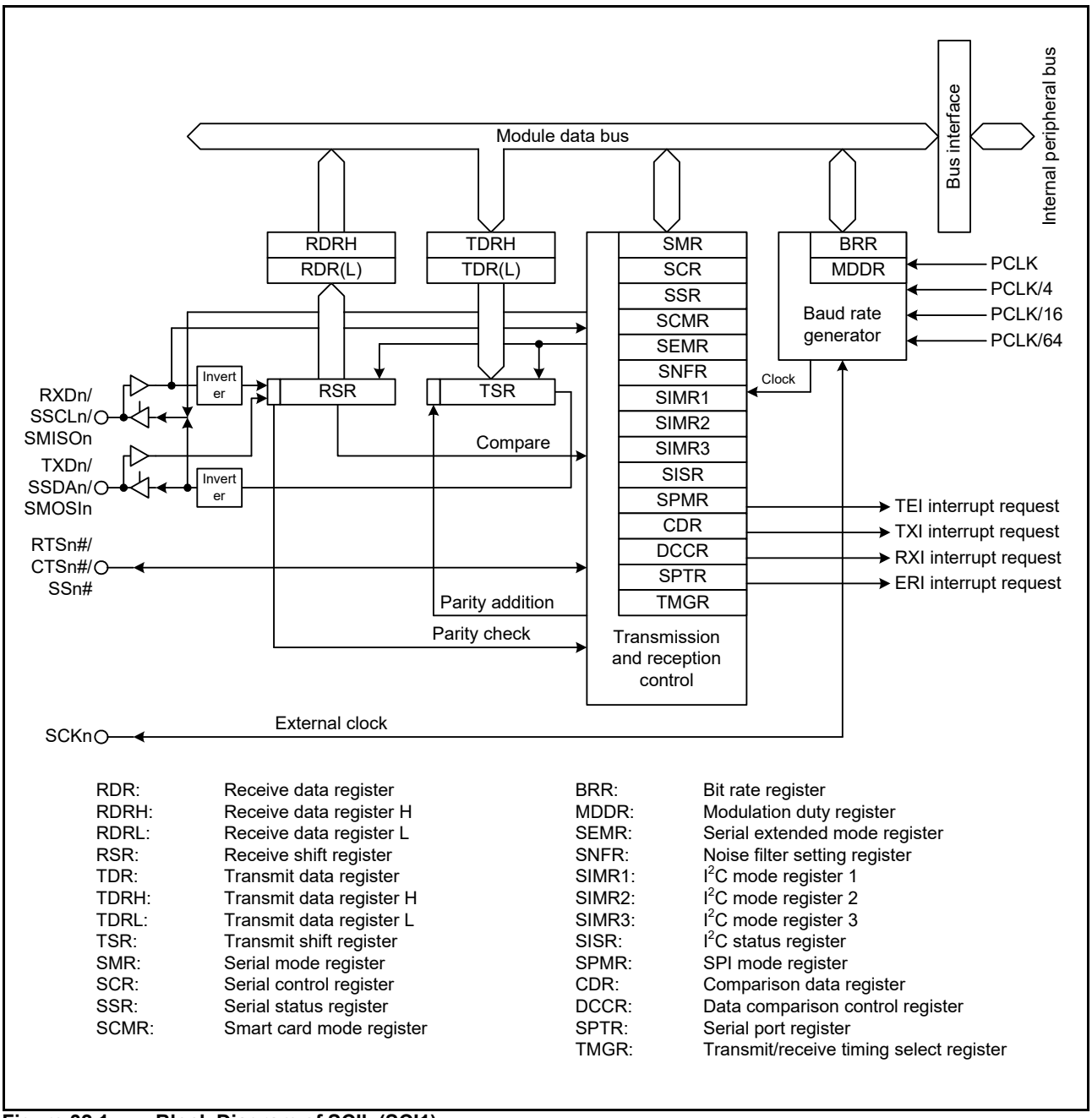
Table 32.2 SCIh Specifications (2/2)

Item		Description
Extended serial mode	Start Frame transmission	• Output of a low level as the Break Field over a specified width and generation of interrupts on completion • Detection of bus collisions and the generation of interrupts on detection
	Start Frame reception	• Detection of the Break Field low width and generation of an interrupt on detection • Comparison of Control Fields 0 and 1 and generation of an interrupt when the two match • Two kinds of data for comparison (primary and secondary) can be set in Control Field 1. • A priority interrupt bit can be set in Control Field 1. • Handling of Start Frames that do not include a Break Field • Handling of Start Frames that do not include a Control Field 0 • Function for measuring bit rates
	I/O control function	• Selectable polarity for TXDX12 and RXDX12 signals • Selection of a digital filter for the RXDX12 signal • Half-duplex operation employing RXDX12 and TXDX12 signals multiplexed on the same pin • Selectable timing for the sampling of data received through RXDX12
	Timer function	• Usable as a reload timer
Bit rate modulation function		Correction of outputs from the on-chip baud rate generator can reduce errors.

Note 1. In simple I^2C mode, only MSB first is available.

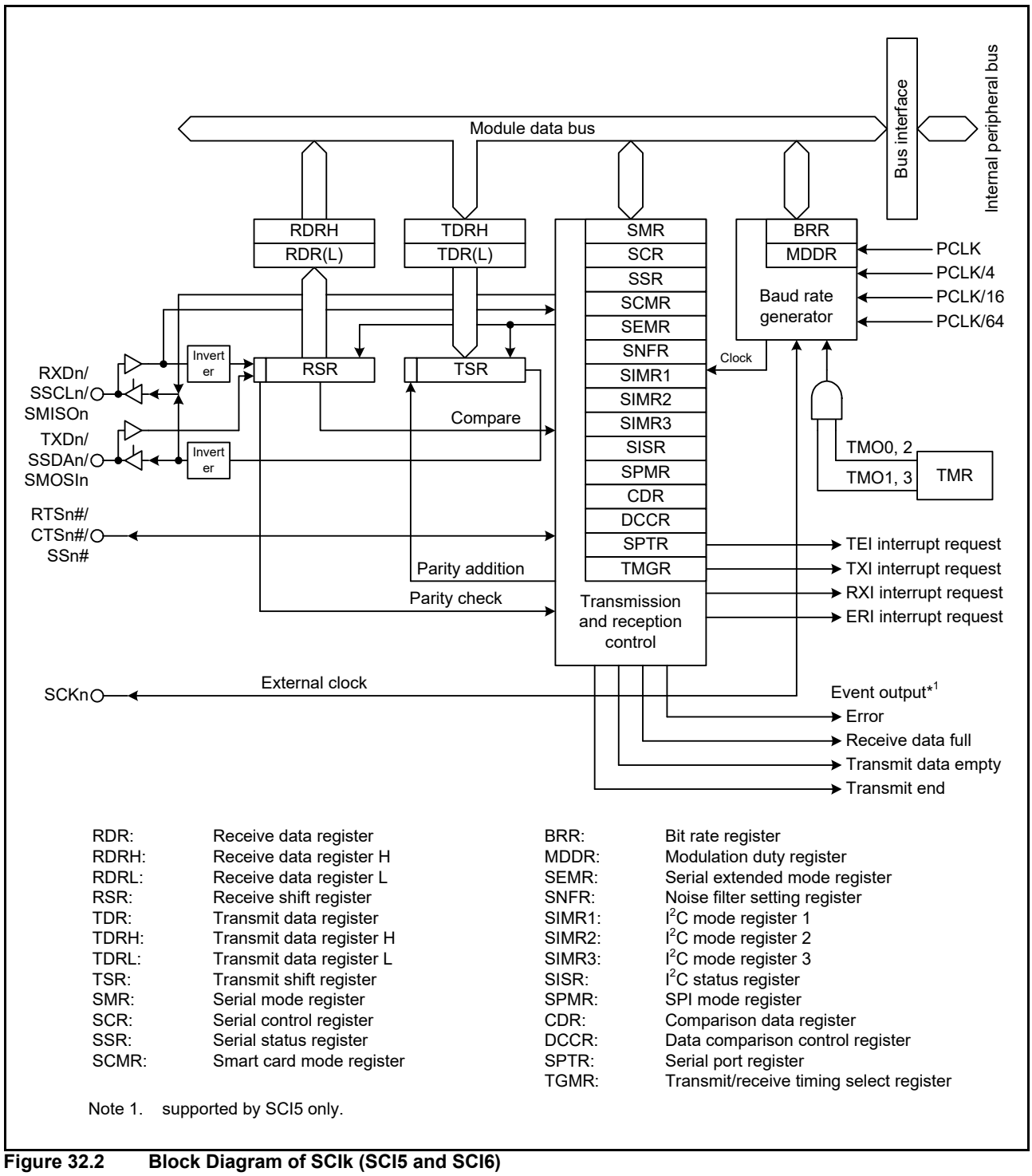
Table 32.3 Functions of SCI Channels

Item	SCI1	SCI5	SCI6	SCI12
Asynchronous mode	Available	Available	Available	Available
Clock synchronous mode	Available	Available	Available	Available
Smart card interface mode	Available	Available	Available	Available
Simple I^2C mode	Available	Available	Available	Available
Simple SPI mode	Available	Available	Available	Available
Data match detection	Available	Available	Available	Not available
Extended serial mode	Not available	Not available	Not available	Available
TMR clock input	Not available	Available	Available	Available
Event link function	Not available	Available	Not available	Not available



RDR:	Receive data register	BRR:	Bit rate register
RDRH:	Receive data register H	MDDR:	Modulation duty register
RDRL:	Receive data register L	SEMR:	Serial extended mode register
RSR:	Receive shift register	SNFR:	Noise filter setting register
TDR:	Transmit data register	SIMR1:	I^2C mode register 1
TDRH:	Transmit data register H	SIMR2:	I^2C mode register 2
TDRL:	Transmit data register L	SIMR3:	I^2C mode register 3
TSR:	Transmit shift register	SISR:	I^2C status register
SMR:	Serial mode register	SPMR:	SPI mode register
SCR:	Serial control register	CDR:	Comparison data register
SSR:	Serial status register	DCCR:	Data comparison control register
SCMR:	Smart card mode register	SPTR:	Serial port register
		TMGR:	Transmit/receive timing select register

Figure 32.1 Block Diagram of SCIk (SCI1)



RDR:	Receive data register	BRR:	Bit rate register
RDRH:	Receive data register H	MDDR:	Modulation duty register
RDRL:	Receive data register L	SEMR:	Serial extended mode register
RSR:	Receive shift register	SNFR:	Noise filter setting register
TDR:	Transmit data register	SIMR1:	I^2C mode register 1
TDRH:	Transmit data register H	SIMR2:	I^2C mode register 2
TDRL:	Transmit data register L	SIMR3:	I^2C mode register 3
TSR:	Transmit shift register	SISR:	I^2C status register
SMR:	Serial mode register	SPMR:	SPI mode register
SCR:	Serial control register	CDR:	Comparison data register
SSR:	Serial status register	DCCR:	Data comparison control register
SCMR:	Smart card mode register	SPTR:	Serial port register
		TGMR:	Transmit/receive timing select register

Note 1. supported by SCI5 only.

Figure 32.2 Block Diagram of SCIk (SCI5 and SCI6)

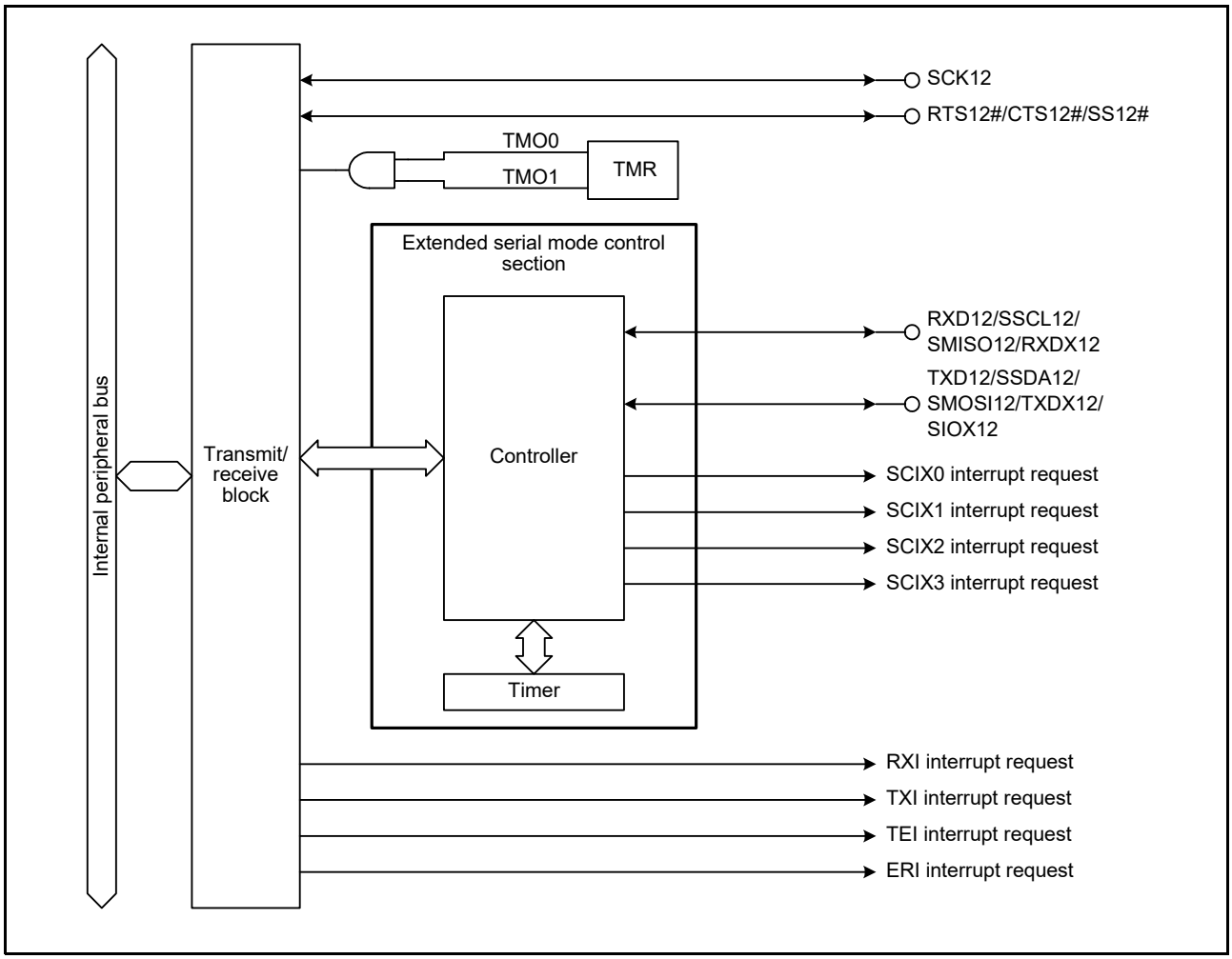


Figure 32.3 Block Diagram of SCIh (SCI12)

Table 32.4 to Table 32.7 list the pin configuration of the SCIs for the individual modes.

Table 32.4 SCI Pin Configuration in Asynchronous Mode and Clock Synchronous Mode

Channel	Pin Name	I/O	Function
SCI1	SCK1	I/O	SCI1 clock input/output
	RXD1	Input	SCI1 receive data input
	TXD1	Output	SCI1 transmit data output
	CTS1#/RTS1#	I/O	SCI1 transfer start control input/output
SCI5	SCK5	I/O	SCI5 clock input/output
	RXD5	Input	SCI5 receive data input
	TXD5	Output	SCI5 transmit data output
	CTS5#/RTS5#	I/O	SCI5 transfer start control input/output
SCI6	SCK6	I/O	SCI6 clock input/output
	RXD6	Input	SCI6 receive data input
	TXD6	Output	SCI6 transmit data output
	CTS6#/RTS6#	I/O	SCI6 transfer start control input/output
SCI12	SCK12	I/O	SCI12 clock input/output
	RXD12	Input	SCI12 receive data input
	TXD12	Output	SCI12 transmit data output
	CTS12#/RTS12#	I/O	SCI12 transfer start control input/output

Table 32.5 SCI Pin Configuration in Simple I^2C Mode

Channel	Pin Name	I/O	Function
SCI1	SSCL1	I/O	SCI1 I^2C clock input/output
	SSDA1	I/O	SCI1 I^2C data input/output
SCI5	SSCL5	I/O	SCI5 I^2C clock input/output
	SSDA5	I/O	SCI5 I^2C data input/output
SCI6	SSCL6	I/O	SCI6 I^2C clock input/output
	SSDA6	I/O	SCI6 I^2C data input/output
SCI12	SSCL12	I/O	SCI12 I^2C clock input/output
	SSDA12	I/O	SCI12 I^2C data input/output

Table 32.6 SCI Pin Configuration in Simple SPI Mode (1/2)

Channel	Pin Name	I/O	Function
SCI1	SCK1	I/O	SCI1 clock input/output
	SMISO1	I/O	SCI1 slave transmit data input/output
	SMOSI1	I/O	SCI1 master transmit data input/output
	SS1#	Input	SCI1 chip select input
SCI5	SCK5	I/O	SCI5 clock input/output
	SMISO5	I/O	SCI5 slave transmit data input/output
	SMOSI5	I/O	SCI5 master transmit data input/output
	SS5#	Input	SCI5 chip select input
SCI6	SCK6	I/O	SCI6 clock input/output
	SMISO6	I/O	SCI6 slave transmit data input/output
	SMOSI6	I/O	SCI6 master transmit data input/output
	SS6#	Input	SCI6 chip select input

Table 32.6 SCI Pin Configuration in Simple SPI Mode (2/2)

Channel	Pin Name	I/O	Function
SCI12	SCK12	I/O	SCI12 clock input/output
	SMISO12	I/O	SCI12 slave transmit data input/output
	SMOSI12	I/O	SCI12 master transmit data input/output
	SS12#	Input	SCI12 chip select input

Table 32.7 SCI Pin Configuration in Extended Serial Mode

Channel	Pin Name	I/O	Function
SCI12	RXDX12	Input	SCI12 receive data input
	TXDX12	Output	SCI12 transmit data output
	SIOX12	I/O	SCI12 transfer data input/output

32.2 Register Descriptions

32.2.1 Receive Shift Register (RSR)

The RSR register is a shift register which is used to receive serial data input from the RXDn pin and converts it into parallel data.
When one frame of data has been received, it is automatically transferred to the RDR register.
The RSR register cannot be directly accessed by the CPU.

32.2.2 Receive Data Register (RDR)

Address(es): SCI1.RDR 0008 A025h, SCI5.RDR 0008 A0A5h, SCI6.RDR 0008 A0C5h, SCI12.RDR 0008 B305h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

The RDR register is an 8-bit register that stores receive data.
When one frame of serial data has been received, the received serial data is transferred from the RSR register to the RDR register. Then the RSR register can receive the next data.
Since the RSR and RDR registers function as a double buffer in this way, continuous receive operations can be performed.
Read the RDR register only once after a receive data full interrupt (RXI) has occurred. Note that if next one frame of data is received before reading receive data from the RDR register, an overrun error occurs.
The RDR register cannot be written to by the CPU.

32.2.3 Receive Data Register H, L, HL (RDRH, RDRL, RDRHL)

- Receive Data Register H (RDRH)

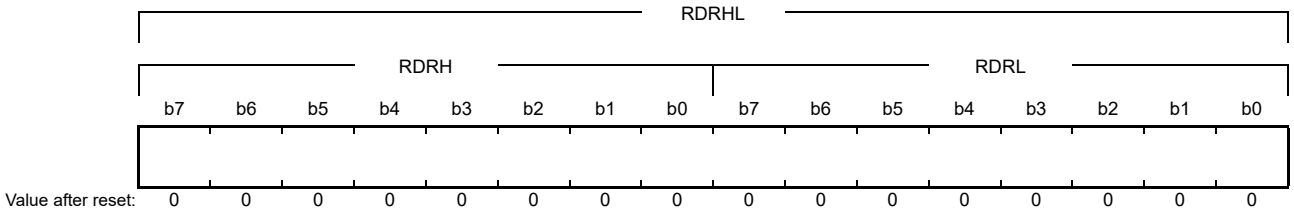
Address(es): SCI1.RDRH 0008 A030h, SCI5.RDRH 0008 A0B0h, SCI6.RDRH 0008 A0D0h, SCI12.RDRH 0008 B310h

- Receive Data Register L (RDRL)

Address(es): SCI1.RDRL 0008 A031h, SCI5.RDRL 0008 A0B1h, SCI6.RDRL 0008 A0D1h, SCI12.RDRL 0008 B311h

- Receive Data Register HL (RDRHL)

Address(es): SCI1.RDRHL 0008 A030h, SCI5.RDRHL 0008 A0B0h, SCI6.RDRHL 0008 A0D0h, SCI12.RDRHL 0008 B310h



The RDRH and RDRL registers are 8-bit registers that store receive data. Use these registers when asynchronous mode and 9-bit data length are selected.

The RDRL register is the shadow register of the RDR register; i.e. access to the RDRL register is equivalent to access to the RDR register.

After one frame of data is received, the received data is transferred from the RSR register to these registers, thus allowing the RSR register to receive the next data.

The RSR, RDRH and RDRL registers have a double-buffered construction to enable continuous reception.

Read the RDRH and RDRL registers should be performed only once in the order from the RDRH register to the RDRL register when a receive data full interrupt (RXI) request is issued. Note that an overrun error occurs when the next frame of data is received before the received data has been read from the RDRL register.

The CPU cannot write to the RDRH and RDRL registers. Bits 0 to 7 in the RDRH register are fixed to 0. These bits are read as 0.

The RDRHL register can be accessed in 16-bit units.

32.2.4 Transmit Data Register (TDR)

Address(es): SCI1.TDR 0008 A023h, SCI5.TDR 0008 A0A3h, SCI6.TDR 0008 A0C3h, SCI12.TDR 0008 B303h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1

The TDR register is an 8-bit register that stores transmit data.
When the SCI detects that the TSR register is empty, it transfers the transmit data written in the TDR register to the TSR register and starts transmission.
The double-buffered structures of the TDR register and the TSR register enable continuous serial transmission. If the next transmit data has already been written to the TDR register when one frame of data is transmitted, the SCI transfers the written data to the TSR register to continue transmission.
The CPU is able to read from or write to the TDR register at any time. Only write transmit data to the TDR register once after each instance of the transmit data empty interrupt (TXI).

32.2.5 Transmit Data Register H, L, HL (TDRH, TDRL, TDRHL)

- Transmit Data Register H (TDRH)

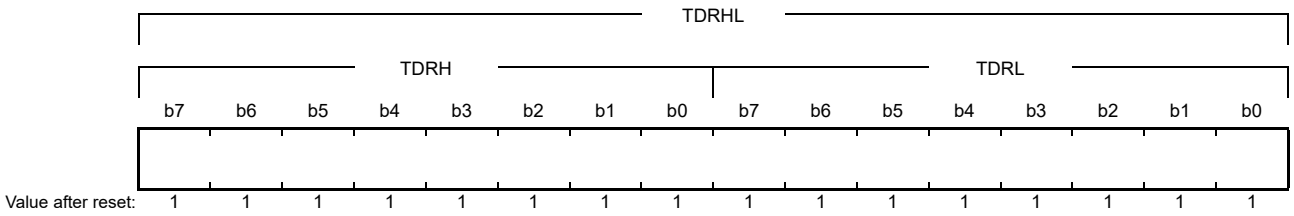
Address(es): SCI1.TDRH 0008 A02Eh, SCI5.TDRH 0008 A0AEh, SCI6.TDRH 0008 A0CEh, SCI12.TDRH 0008 B30Eh

- Transmit Data Register L (TDRL)

Address(es): SCI1.TDRL 0008 A02Fh, SCI5.TDRL 0008 A0AFh, SCI6.TDRL 0008 A0CFh, SCI12.TDRL 0008 B30Fh

- Transmit Data Register HL (TDRHL)

Address(es): SCI1.TDRHL 0008 A02Eh, SCI5.TDRHL 0008 A0AEh, SCI6.TDRHL 0008 A0CEh, SCI12.TDRHL 0008 B30Eh



The TDRH and TDRL registers are 8-bit registers that store transmit data. Use these registers when asynchronous mode and 9-bit data length are selected.
The TDRL register is the shadow register of the TDR register; i.e. access to the TDRL register is equivalent to access to the TDR register.
When empty space is detected in the TSR register, the transmit data stored in the TDRH and TDRL registers is transferred to the TSR register; i.e., transmitting is started.
The TSR, TDRH and TDRL registers have a double-buffered construction to realize continuous reception. When the next data to be transmitted is stored in the TDRL register after one frame of data has been transmitted, the transmitting operation is continued by transfer to the TSR register.
The CPU can read and write to the TDRH and TDRL registers. Bits 0 to 7 in the RDRH register are fixed to 1. These bits are read as 1. The write value should be 1.
Writing transmit data to the TDRH and TDRL registers should be performed only once in the order from the TDRH register to the TDRL register when a transmit data empty interrupt (TXI) request is issued.
The TDRHL register can be accessed in 16-bit units.

32.2.6 Transmit Shift Register (TSR)

The TSR register is a shift register that transmits serial data.
To perform serial data transmission, the SCI first automatically transfers transmit data from the TDR register to the TSR register, and then sends the data to the TXDn pin.
The TSR register cannot be directly accessed by the CPU.

32.2.7 Serial Mode Register (SMR)

Some bits in the SMR register have different functions in smart card interface mode and non-smart card interface mode.

(1) Non-Smart Card Interface Mode (SCMR.SMIF = 0)

Address(es): SCI1.SMR 0008 A020h, SCI5.SMR 0008 A0A0h, SCI6.SMR 0008 A0C0h, SCI12.SMR 0008 B300h

	b7	b6	b5	b4	b3	b2	b1	b0
	CM	CHR	PE	PM	STOP	MP	CKS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CKS[1:0]	Clock Select	b1 b0 0 0: PCLK (n = 0)*1 0 1: PCLK/4 (n = 1)*1 1 0: PCLK/16 (n = 2)*1 1 1: PCLK/64 (n = 3)*1	R/W*4
b2	MP	Multi-Processor Mode	(Valid only in asynchronous mode) 0: Multi-processor communications function is disabled 1: Multi-processor communications function is enabled	R/W*4
b3	STOP	Stop Bit Length	(Valid only in asynchronous mode) 0: 1 stop bit 1: 2 stop bits	R/W*4
b4	PM	Parity Mode	(Valid only when the PE bit is 1) 0: Selects even parity 1: Selects odd parity	R/W*4
b5	PE	Parity Enable	(Valid only in asynchronous mode) • When transmitting 0: Parity bit addition is not performed 1: The parity bit is added • When receiving 0: Parity bit checking is not performed 1: The parity bit is checked	R/W*4
b6	CHR	Character Length	(Valid only in asynchronous mode*2) Selects in combination with the SCMR.CHR1 bit. CHR1 CHR 0 0: Transmit/receive in 9-bit data length 0 1: Transmit/receive in 9-bit data length 1 0: Transmit/receive in 8-bit data length (initial value) 1 1: Transmit/receive in 7-bit data length*3	R/W*4
b7	CM	Communications Mode	0: Asynchronous mode or simple I2C mode 1: Clock synchronous mode or simple SPI mode	R/W*4

Note 1. n is the decimal notation of the value of n in the BRR register (refer to section 32.2.11, Bit Rate Register (BRR)).
Note 2. In other than asynchronous mode, this bit setting is invalid and a fixed data length of 8 bits is used.
Note 3. LSB first is fixed and the MSB (bit 7) in the TDR register is not transmitted in transmission.
Note 4. Writable only when the SCR.TE bit is 0 and the SCR.RE bit is 0 (both serial transmission and reception are disabled).

CKS[1:0] Bits (Clock Select)

These bits select the clock source for the on-chip baud rate generator.

For the relation between the settings of these bits and the baud rate, refer to section 32.2.11, Bit Rate Register (BRR).

MP Bit (Multi-Processor Mode)

Disables/enables the multi-processor communications function. The settings of the PE bit and PM bit are invalid in multi-processor mode.

STOP Bit (Stop Bit Length)

Selects the stop bit length in transmission.
In reception, only the first stop bit is checked regardless of this bit setting. If the second stop bit is 0, it is treated as the start bit of the next transmit frame.

PM Bit (Parity Mode)

Selects the parity mode (even or odd) for transmission and reception.
The setting of the PM bit is invalid in multi-processor mode.

PE Bit (Parity Enable)

When this bit is set to 1, the parity bit is added to transmit data, and the parity bit is checked in reception.
Irrespective of the setting of the PE bit, the parity bit is not added or checked in multi-processor format.

CHR Bit (Character Length)

Selects the data length for transmission and reception.
Selects in combination with the SCMR.CHR1 bit.
In other than asynchronous mode, a fixed data length of 8 bits is used.

(2) Smart Card Interface Mode (SCMR.SMIF = 1)

Address(es): SMCI1.SMR 0008 A020h, SMCI5.SMR 0008 A0A0h, SMCI6.SMR 0008 A0C0h, SMCI12.SMR 0008 B300h

	b7	b6	b5	b4	b3	b2	b1	b0
	GM	BLK	PE	PM	BCP[1:0]		CKS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CKS[1:0]	Clock Select	b1 b0 0 0: PCLK (n = 0)*1 0 1: PCLK/4 (n = 1)*1 1 0: PCLK/16 (n = 2)*1 1 1: PCLK/64 (n = 3)*1	R/W*2
b3, b2	BCP[1:0]	Base Clock Pulse	Selects the number of base clock cycles in combination with the SCMR.BCP2 bit. Table 32.8 lists the combinations of the SCMR.BCP2 bit and SMR.BCP[1:0] bits.	R/W*2
b4	PM	Parity Mode	(Valid only when the PE bit is 1) 0: Selects even parity 1: Selects odd parity	R/W*2
b5	PE	Parity Enable	When this bit is set to 1, a parity bit is added to transmit data, and the parity of received data is checked. Set this bit to 1 in smart card interface mode.	R/W*2
b6	BLK	Block Transfer Mode	0: Non-block transfer mode operation 1: Block transfer mode operation	R/W*2
b7	GM	GSM Mode	0: Non-GSM mode operation 1: GSM mode operation	R/W*2

Note 1. n is the decimal notation of the value of n in the BRR register (refer to section 32.2.11, Bit Rate Register (BRR)).
Note 2. Writable only when the SCR.TE bit is 0 and the SCR.RE bit is 0 (both serial transmission and reception are disabled).

CKS[1:0] Bits (Clock Select)

These bits select the clock source for the on-chip baud rate generator.

For the relationship between the settings of these bits and the baud rate, refer to section 32.2.11, Bit Rate Register (BRR).

BCP[1:0] Bits (Base Clock Pulse)

These bits select the number of base clock cycles in a 1-bit data transfer time in smart card interface mode.

Set these bits in combination with the SCMR.BCP2 bit.

For details, refer to section 32.6.4, Receive Data Sampling Timing and Reception Margin.

Table 32.8 Combinations of the SCMR.BCP2 Bit and SMR.BCP[1:0] Bits

SCMR.BCP2 Bit	SMR.BCP[1:0] Bits		Number of Base Clock Cycles for 1-Bit Transfer Period
0	0	0	93 clock cycles (S = 93)*1
0	0	1	128 clock cycles (S = 128)*1
0	1	0	186 clock cycles (S = 186)*1
0	1	1	512 clock cycles (S = 512)*1
1	0	0	32 clock cycles (S = 32)*1 (Initial Value)
1	0	1	64 clock cycles (S = 64)*1
1	1	0	372 clock cycles (S = 372)*1
1	1	1	256 clock cycles (S = 256)*1

Note 1. S is the value of S in the BRR register (refer to section 32.2.11, Bit Rate Register (BRR)).

PM Bit (Parity Mode)

Selects the parity mode for transmission and reception (even or odd).
For details on the usage of this bit in smart card interface mode, refer to section 32.6.2, Data Format (Except in Block Transfer Mode).

PE Bit (Parity Enable)

Set the PE bit to 1.
The parity bit is added to transmit data before transmission, and the parity bit is checked in reception.

BLK Bit (Block Transfer Mode)

Setting this bit to 1 allows block transfer mode operation.
For details, refer to section 32.6.3, Block Transfer Mode.

GM Bit (GSM Mode)

Setting this bit to 1 allows GSM mode operation.
In GSM mode, the SSR.TEND flag set timing is put forward to 11.0 etu (elementary time unit = 1-bit transfer time) from the start and the clock output control function is appended. For details, refer to section 32.6.6, Serial Data Transmission (Except in Block Transfer Mode) and section 32.6.8, Clock Output Control.

32.2.8 Serial Control Register (SCR)

Some bits in the SCR register have different functions in smart card interface mode and non-smart card interface mode.

(1) Non-Smart Card Interface Mode (SCMR.SMIF = 0)

Address(es): SCI1.SCR 0008 A022h, SCI5.SCR 0008 A0A2h, SCI6.SCR 0008 A0C2h, SCI12.SCR 0008 B302h

	b7	b6	b5	b4	b3	b2	b1	b0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CKE[1:0]	Clock Enable	(Asynchronous mode) b1 b0 0 0: On-chip baud rate generator The SCKn pin becomes high-impedance. 0 1: On-chip baud rate generator The clock with the same frequency as the bit rate is output from the SCKn pin. 1 x: External clock or TMR clock*2 • The clock with a frequency 16 times the bit rate should be input from the SCKn pin. Input a clock signal with a frequency eight times the bit rate when the SEMR.ABCS bit is 1. • The SCKn pin becomes high-impedance when the TMR clock*2 is used. (Clock synchronous mode) b1 b0 0 x: Internal clock The SCKn pin functions as the clock output pin. 1 x: External clock The SCKn pin functions as the clock input pin.	R/W*1
b2	TEIE	Transmit End Interrupt Enable	0: A TEI interrupt request is disabled 1: A TEI interrupt request is enabled	R/W
b3	MPIE	Multi-Processor Interrupt Enable	(Valid in asynchronous mode when the SMR.MP bit is 1) 0: Normal reception 1: When the data with the multi-processor bit set to 0 is received, the data is not read, and setting the status flags RDRF, ORER, and FER in the SSR register to 1 is disabled. When the data with the multi-processor bit set to 1 is received, the MPIE bit is automatically cleared to 0, and normal reception is resumed.	R/W
b4	RE	Receive Enable	0: Serial reception is disabled 1: Serial reception is enabled	R/W*3
b5	TE	Transmit Enable	0: Serial transmission is disabled 1: Serial transmission is enabled	R/W*3
b6	RIE	Receive Interrupt Enable	0: RXI and ERI interrupt requests are disabled 1: RXI and ERI interrupt requests are enabled	R/W
b7	TIE	Transmit Interrupt Enable	0: A TXI interrupt request is disabled 1: A TXI interrupt request is enabled	R/W

x: Don't care

Note 1. Writable only when TE = 0 and RE = 0.

Note 2. TMR clock is selectable for SCI5, SCI6, and SCI12.

Note 3. 1 can be written only when TE = 0 and RE = 0, while the SMR.CM bit is 1. After setting TE or RE to 1, only 0 can be written to TE and RE. While the SMR.CM bit is 0 and the SIMR1.IICM bit is 0, writing is enabled under any condition.

CKE[1:0] Bits (Clock Enable)

These bits select the clock source and SCKn pin function.
The combination of the settings of these bits and of the SEMR.ACS0 bit sets the internal TMR clock.

TEIE Bit (Transmit End Interrupt Enable)

Enables or disables a TEI interrupt request.
A TEI interrupt request is disabled by setting the TEIE bit to 0.
In simple I^2C mode, the TEI is allocated to the interrupt on completion of issuing a start, restart, or stop condition (STI). In this case, the TEIE bit can be used to enable or disable the STI.

MPIE Bit (Multi-Processor Interrupt Enable)

When this bit is set to 1 and the data with the multi-processor bit set to 0 is received, the data is not read and setting the status flags RDRF, ORER, and FER in the SSR register to 1 is disabled. When the data with the multi-processor bit set to 1 is received, the MPIE bit is automatically cleared to 0, and normal reception is resumed. For details, refer to section 32.4, Multi-Processor Communications Function.
When the data with the multi-processor bit set to 0 is received, the receive data is not transferred from the RSR register to the RDR register, a receive error is not detected, and setting the flags RDRF, ORER, and FER to 1 is disabled.
When the data with the multi-processor bit set to 1 is received, the MPB bit is set to 1, the MPIE bit is automatically cleared to 0, the RXI and ERI interrupt requests are enabled (if the SCR.RIE bit is set to 1), and setting the flags RDRF, ORER, and FER to 1 is enabled.
Set the MPIE bit to 0 if multi-processor communications function is not to be used.

RE Bit (Receive Enable)

Enables or disables serial reception.
When this bit is set to 1, serial reception is started by detecting the start bit in asynchronous mode or the synchronous clock input in clock synchronous mode. Note that the SMR register should be set prior to setting the RE bit to 1 in order to designate the reception format.
Even if reception is halted by setting the RE bit to 0, the ORER, FER, PER, and RDRF flags in the SSR register are not affected and the previous value is retained.

TE Bit (Transmit Enable)

Enables or disables serial transmission.
When this bit is set to 1, serial transmission is started by writing transmit data to the TDR register. Note that the SMR register should be set prior to setting the TE bit to 1 in order to designate the transmission format.

RIE Bit (Receive Interrupt Enable)

Enables or disables RXI and ERI interrupt requests.
An RXI interrupt request is disabled by setting the RIE bit to 0.
An ERI interrupt request can be canceled by reading 1 from the ORER, FER, or PER flag in the SSR register and then setting the flag to 0, or setting the RIE bit to 0.

TIE Bit (Transmit Interrupt Enable)

Enables or disables TXI interrupt request.
A TXI interrupt request is disabled by setting the TIE bit to 0.

(2) Smart Card Interface Mode (SCMR.SMIF = 1)

Address(es): SMCI1.SCR 0008 A022h, SMCI5.SCR 0008 A0A2h, SMCI6.SCR 0008 A0C2h, SMCI12.SCR 0008 B302h

	b7	b6	b5	b4	b3	b2	b1	b0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CKE[1:0]	Clock Enable	• When SMR.GM = 0 b1 b0 0 0: Output disabled The SCKn pin becomes high-impedance. 0 1: Clock output 1 x: Setting prohibited • When SMR.GM = 1 b1 b0 0 0: Output fixed low x 1: Clock output 1 0: Output fixed high	R/W*1
b2	TEIE	Transmit End Interrupt Enable	This bit should be 0 in smart card interface mode.	R/W
b3	MPIE	Multi-Processor Interrupt Enable	This bit should be 0 in smart card interface mode.	R/W
b4	RE	Receive Enable	0: Serial reception is disabled 1: Serial reception is enabled	R/W*2
b5	TE	Transmit Enable	0: Serial transmission is disabled 1: Serial transmission is enabled	R/W*2
b6	RIE	Receive Interrupt Enable	0: RXI and ERI interrupt requests are disabled 1: RXI and ERI interrupt requests are enabled	R/W
b7	TIE	Transmit Interrupt Enable	0: A TXI interrupt request is disabled 1: A TXI interrupt request is enabled	R/W

x: Don't care

Note 1. Writable only when TE = 0 and RE = 0.

Note 2. 1 can be written only when TE = 0 and RE = 0. After setting TE or RE to 1, only 0 can be written in TE and RE.

For details on interrupt requests, refer to section 32.12, Interrupt Sources.

CKE[1:0] Bits (Clock Enable)

These bits control the clock output from the SCKn pin.

In GSM mode, clock output can be dynamically switched. For details, refer to section 32.6.8, Clock Output Control.

TEIE Bit (Transmit End Interrupt Enable)

This bit should be 0 in smart card interface mode.

MPIE Bit (Multi-Processor Interrupt Enable)

This bit should be 0 in smart card interface mode.

RE Bit (Receive Enable)

Enables or disables serial reception.

When this bit is set to 1, serial reception is started by detecting the start bit. Note that the SMR register should be set prior to setting the RE bit to 1 in order to designate the reception format.

Even if reception is halted by setting the RE bit to 0, the ORER, FER, and PER flags in the SSR register are not affected and the previous value is retained.

TE Bit (Transmit Enable)

Enables or disables serial transmission.

When this bit is set to 1, serial transmission is started by writing transmit data to the TDR register. Note that the SMR register should be set prior to setting the TE bit to 1 in order to designate the transmission format.

RIE Bit (Receive Interrupt Enable)

Enables or disables RXI and ERI interrupt requests.

An RXI interrupt request is disabled by setting the RIE bit to 0.

An ERI interrupt request can be canceled by reading 1 from the ORER, FER, or PER flag in the SSR register and then setting the flag to 0, or setting the RIE bit to 0.

TIE Bit (Transmit Interrupt Enable)

Enables or disables TXI interrupt request.

A TXI interrupt request is disabled by setting the TIE bit to 0.

32.2.9 Serial Status Register (SSR)

Some bits in the SSR register have different functions in smart card interface mode and non-smart card interface mode.

(1) Non-Smart Card Interface Mode (SCMR.SMIF = 0)

Address(es): SCI1.SSR 0008 A024h, SCI5.SSR 0008 A0A4h, SCI6.SSR 0008 A0C4h, SCI12.SSR 0008 B304h

	b7	b6	b5	b4	b3	b2	b1	b0
	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
Value after reset:	1	0	0	0	0	1	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MPBT	Multi-Processor Bit Transfer	Sets the multi-processor bit for adding to the transmission frame 0: Data transmission cycles 1: ID transmission cycles	R/W
b1	MPB	Multi-Processor	Value of the multi-processor bit in the reception frame 0: Data transmission cycles 1: ID transmission cycles	R
b2	TEND	Transmit End Flag	0: A character is being transmitted. 1: Character transfer has been completed.	R
b3	PER	Parity Error Flag	0: No parity error occurred 1: A parity error has occurred	R/(W) *1
b4	FER	Framing Error Flag	0: No framing error occurred 1: A framing error has occurred	R/(W) *1
b5	ORER	Overrun Error Flag	0: No overrun error occurred 1: An overrun error has occurred	R/(W) *1
b6	RDRF	Receive Data Full Flag	0: No valid data is held in the RDR register 1: Received data is held in the RDR register	R/(W) *2
b7	TDRE	Transmit Data Empty Flag	0: Data to be transmitted is held in the TDR register 1: No data is held in the TDR register	R/(W) *2

Note 1. Only 0 can be written to this bit, to clear the flag. To clear this flag, confirm that the flag is 1 and then set it to 0.
Note 2. Write 1 when writing is necessary.

MPB Bit (Multi-Processor)

Holds the value of the multi-processor bit in the reception frame. This bit does not change when the SCR.RE bit is 0.

TEND Flag (Transmit End Flag)

Indicates completion of transmission.

[Setting conditions]

- When the SCR.TE bit is set to 0 (serial transmission is disabled)
  When the SCR.TE bit is changed from 0 to 1, the TEND flag is not affected and retains the value 1.
- When the TDR register is not updated at the time of transmission of the tail-end bit of a character being transmitted

[Clearing condition]

- When transmit data are written to the TDR register while the SCR.TE bit is 1
  When setting the TEND flag to 0 to complete the interrupt handling, refer to section 14.5.2, Level Detection.

PER Flag (Parity Error Flag)

Indicates that a parity error has occurred during reception in asynchronous mode and the reception ends abnormally.

[Setting condition]

- When a parity error is detected during reception while data match detection is disabled (for SCI1, SCI5, and SCI6)
- When a parity error is detected during reception (for SCI12)
  Although receive data when the parity error occurs is transferred to the RDR register, no RXI interrupt request occurs. Note that when the PER flag is being set to 1, the subsequent receive data is not transferred to the RDR register.

[Clearing condition]

- When 0 is written to the PER flag after reading PER = 1
  When setting the PER flag to 0 to complete the interrupt handling, refer to section 14.5.2, Level Detection.
  Even when the SCR.RE bit is set to 0 (serial reception is disabled), the PER flag is not affected and retains its previous value.

FER Flag (Framing Error Flag)

Indicates that a framing error has occurred during reception in asynchronous mode and the reception ends abnormally.

[Setting condition]

- When the stop bit is 0 while data match detection is disabled (for SCI1, SCI5, and SCI6)
- When the stop bit is 0 (for SCI12)
  In 2-stop-bit mode, only the first stop bit is checked whether it is 1 but the second stop bit is not checked. Note that although receive data when the framing error occurs is transferred to the RDR register, no RXI interrupt request occurs. In addition, when the FER flag is being set to 1, the subsequent receive data is not transferred to the RDR register.

[Clearing condition]

- When 0 is written to the FER flag after reading FER = 1
  When setting the FER flag to 0 to complete the interrupt handling, refer to section 14.5.2, Level Detection.
  Even when the SCR.RE bit is set to 0, the FER flag is not affected and retains its previous value.

ORER Flag (Overrun Error Flag)

Indicates that an overrun error has occurred during reception and the reception ends abnormally.

[Setting condition]

- When the next data is received before receive data is read from the RDR register
  In the RDR register, receive data prior to an overrun error occurrence is retained, but data received after the overrun error occurrence is lost. When the ORER flag is set to 1, subsequent serial reception cannot be performed. Note that, in clock synchronous mode, serial transmission also cannot continue.

[Clearing condition]

- When 0 is written to the ORER flag after reading ORER = 1
  When setting the ORER flag to 0 to complete the interrupt handling, refer to section 14.5.2, Level Detection.
  Even when the SCR.RE bit is set to 0, the ORER flag is not affected and retains its previous value.

RDRF Flag (Receive Data Full Flag)

Indicates whether the RDR register has received data.

[Setting condition]

- When data has been received normally, and transferred from the RSR register to the RDR register

[Clearing condition]

- When data is read from the RDR register

TDRE Flag (Transmit Data Empty Flag)

Indicates whether the TDR register has data to be transmitted.

[Setting condition]

- When data is transferred from the TDR register to the TSR register

[Clearing condition]

- When data is written to the TDR register

(2) Smart Card Interface Mode (SCMR.SMIF = 1)

Address(es): SMCI1.SSR 0008 A024h, SMCI5.SSR 0008 A0A4h, SMCI6.SSR 0008 A0C4h, SMCI12.SSR 0008 B304h

	b7	b6	b5	b4	b3	b2	b1	b0
	TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT
Value after reset:	1	0	0	0	0	1	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MPBT	Multi-Processor Bit Transfer	This bit should be set to 0 in smart card interface mode.	R/W
b1	MPB	Multi-Processor	This bit is not used in smart card interface mode. It should be set to 0.	R
b2	TEND	Transmit End Flag	0: A character is being transmitted. 1: Character transfer has been completed.	R
b3	PER	Parity Error Flag	0: No parity error occurred 1: A parity error has occurred	R/(W) *1
b4	ERS	Error Signal Status Flag	0: Low error signal not responded 1: Low error signal responded	R/(W) *1
b5	ORER	Overrun Error Flag	0: No overrun error occurred 1: An overrun error has occurred	R/(W) *1
b6	RDRF	Receive Data Full Flag	0: No valid data is held in the RDR register 1: Received data is held in the RDR register	R/(W) *2
b7	TDRE	Transmit Data Empty Flag	0: Data to be transmitted is held in the TDR register 1: No data is held in the TDR register	R/(W) *2

Note 1. Only 0 can be written to this bit, to clear the flag. To clear this flag, confirm that the flag is 1 and then set it to 0.
Note 2. Write 1 when writing is necessary.

TEND Flag (Transmit End Flag)

With no error signal from the receiving side, this bit is set to 1 when further data for transfer is ready to be transferred to the TDR register.

[Setting conditions]

- When the SCR.TE bit is 0 (serial transmission is disabled)
  When the SCR.TE bit is changed from 0 to 1, the TEND flag is not affected and retains the value 1.
- When a specified period has elapsed after the latest transmission of 1 byte, the ERS flag is 0, and the TDR register is not updated
  The set timing is determined by register settings as listed below.
  When SMR.GM = 0 and SMR.BLK = 0, 12.5 etu after the start of transmission
  When SMR.GM = 0 and SMR.BLK = 1, 11.5 etu after the start of transmission
  When SMR.GM = 1 and SMR.BLK = 0, 11.0 etu after the start of transmission
  When SMR.GM = 1 and SMR.BLK = 1, 11.0 etu after the start of transmission

[Clearing condition]

- When transmit data are written to the TDR register while the SCR.TE bit is 1
  When setting the TEND flag to 0 to complete the interrupt handling, refer to section 14.5.2, Level Detection.

PER Flag (Parity Error Flag)

Indicates that a parity error has occurred during reception in asynchronous mode and the reception ends abnormally.

[Setting condition]

- When a parity error is detected during reception
  Although receive data when the parity error occurs is transferred to the RDR register, no RXI interrupt request occurs. Note that when the PER flag is being set to 1, the subsequent receive data is not transferred to the RDR register.

[Clearing condition]

- When 0 is written to the PER flag after reading PER = 1
  When setting the PER flag to 0 to complete the interrupt handling, refer to section 14.5.2, Level Detection.
  Even when the SCR.RE bit is set to 0 (serial reception is disabled), the PER flag is not affected and retains its previous value.

ERS Flag (Error Signal Status Flag)

[Setting condition]

- When a low error signal is sampled

[Clearing condition]

- When 0 is written to the ERS flag after reading ERS = 1
  When setting the ERS flag to 0 to complete the interrupt handling, refer to section 14.5.2, Level Detection.
  Even when the SCR.RE bit is set to 0, the ERS flag is not affected and retains its previous value.

ORER Flag (Overrun Error Flag)

Indicates that an overrun error has occurred during reception and the reception ends abnormally.

[Setting condition]

- When the next data is received before receive data is read from the RDR register
  In the RDR register, the receive data prior to an overrun error occurrence is retained, but data received following the overrun error occurrence is lost. When the ORER flag is set to 1, subsequent serial reception cannot be performed.

[Clearing condition]

- When 0 is written to the ORER flag after reading ORER = 1
  When setting the ORER flag to 0 to complete the interrupt handling, refer to section 14.5.2, Level Detection.
  Even when the SCR.RE bit is set to 0, the ORER flag is not affected and retains its previous value.

RDRF Flag (Receive Data Full Flag)

Indicates whether the RDR register has received data.

[Setting condition]

- When data has been received normally, and transferred from the RSR register to the RDR register

[Clearing condition]

- When data is read from the RDR register

TDRE Flag (Transmit Data Empty Flag)

Indicates whether the TDR register has data to be transmitted.

[Setting condition]

- When data is transferred from the TDR register to the TSR register

[Clearing condition]

- When data is written to the TDR register

32.2.10 Smart Card Mode Register (SCMR)

Address(es): SCI1.SCMR 0008 A026h, SCI5.SCMR 0008 A0A6h, SCI6.SCMR 0008 A0C6h, SCI12.SCMR 0008 B306h, SMCI1.SCMR 0008 A026h, SMCI5.SCMR 0008 A0A6h, SMCI6.SCMR 0008 A0C6h, SMCI12.SCMR 0008 B306h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCP2	—	—	CHR1	SDIR	SINV	—	SMIF
Value after reset:	1	1	1	1	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	SMIF	Smart Card Interface Mode Select	0: Non-smart card interface mode (Asynchronous mode, clock synchronous mode, simple SPI mode, or simple I2C mode) 1: Smart card interface mode	R/W*1
b1	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b2	SINV	Transmitted/Received Data Invert*2, *3	0: Data bits in the TDR register are transferred to the TSR register as they are. Data bits in the RSR register are transferred to the RDR register as they are. 1: Data bits in the TDR register are transferred to the TSR register with inverting. Data bits in the RSR register are transferred to the RDR register with inverting.	R/W*1
b3	SDIR	Transmitted/Received Data Transfer Direction*2, *4	0: Transfer with LSB first 1: Transfer with MSB first	R/W*1
b4	CHR1	Character Length 1*5	Selects in combination with the SMR.CHR bit. CHR1 CHR 0 0: Transmit/receive in 9-bit data length 0 1: Transmit/receive in 9-bit data length 1 0: Transmit/receive in 8-bit data length (initial value) 1 1: Transmit/receive in 7-bit data length*6	R/W*1
b6, b5	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b7	BCP2	Base Clock Pulse 2	Selects the number of base clock cycles in combination with the SMR.BCP[1:0] bits. Table 32.9 lists the combinations of the SCMR.BCP2 bit and SMR.BCP[1:0] bits.	R/W*1

Note 1. Writable only when the SCR.TE bit is 0 and the SCR.RE bit is 0 (both serial transmission and reception are disabled).
Note 2. This bit can be used in the smart card interface mode, asynchronous mode (multi-processor mode), clock synchronous mode, and simple SPI mode.
Note 3. Set this bit to 0 if operation is to be in simple I2C mode.
Note 4. Set this bit to 1 if operation is to be in simple I2C mode.
Note 5. This bit is only valid in asynchronous mode. The setting is invalid and a fixed data length of 8 bits is used in modes other than asynchronous mode.
Note 6. LSB first should be selected and the value of MSB (b7) in the TDR register cannot be transmitted.

SMIF Bit (Smart Card Interface Mode Select)

When this bit is set to 1, smart card interface mode is selected.

When this bit is set to 0, non-smart card interface mode, i.e., asynchronous mode (including multi-processor mode), clock synchronous mode, simple SPI mode, or simple I2C mode is selected.

SINV Bit (Transmitted/Received Data Invert)

This bit is used to invert the logic level of the data bits when the data is transferred between data register and shift register. This bit does not affect the logic level of the parity bit. To invert the parity bit, invert the SMR.PM bit.

CHR1 Bit (Character Length 1)

Selects the data length of transmit/receive data.

Selects in combination with the SMR.CHR bit.

A fixed data length of 8 bits is used in modes other than asynchronous mode.

BCP2 Bit (Base Clock Pulse 2)

Selects the number of base clock cycles in a 1-bit data transfer time in smart card interface mode. Set this bit in combination with the SMR.BCP[1:0] bits.

Table 32.9 Combinations of the SCMR.BCP2 Bit and SMR.BCP[1:0] Bits

SCMR.BCP2 Bit	SMR.BCP[1:0] Bits		Number of Base Clock Cycles for 1-Bit Transfer Period
0	0	0	93 clock cycles (S = 93)*1
0	0	1	128 clock cycles (S = 128)*1
0	1	0	186 clock cycles (S = 186)*1
0	1	1	512 clock cycles (S = 512)*1
1	0	0	32 clock cycles (S = 32)*1 (Initial Value)
1	0	1	64 clock cycles (S = 64)*1
1	1	0	372 clock cycles (S = 372)*1
1	1	1	256 clock cycles (S = 256)*1

Note 1. S is the value of S in the BRR register (refer to section 32.2.11, Bit Rate Register (BRR)).

32.2.11 Bit Rate Register (BRR)

Address(es): SCI1.BRR 0008 A021h, SCI5.BRR 0008 A0A1h, SCI6.BRR 0008 A0C1h, SCI12.BRR 0008 B301h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1

The BRR register is an 8-bit register that adjusts the bit rate.
As each SCI channel has independent baud rate generator control, different bit rates can be set for each. Table 32.10 and Table 32.11 shows the relationship between the setting (N) in the BRR register and the bit rate (B) for normal asynchronous mode, multi-processor communication, clock synchronous mode, smart card interface mode, simple SPI mode, and simple I²C mode.
The BRR register is writable only when the TE and RE bits in the SCR register are 0.

Table 32.10 Relationship between N Setting in the BRR Register and Bit Rate B (for SCI1, SCI5, and SCI6)

Mode	SEMR Settings			BRR Setting	Error (%)
	BGDM bit	ABCS bit	ABCSE bit		
Asynchronous, multi-processor communication	0	0	0	$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 64 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
	0	1	0	$N = \frac{PCLK \times 10^6}{32 \times 2^{2n-1} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 32 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
	1	0	0		
	1	1	0	$N = \frac{PCLK \times 10^6}{16 \times 2^{2n-1} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 16 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
	0 or 1	0 or 1	1	$N = \frac{PCLK \times 10^6}{12 \times 2^{2n-1} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 12 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
Clock synchronous, simple SPI				$N = \frac{PCLK \times 10^6}{8 \times 2^{2n-1} \times B} - 1$	
Smart card interface				$N = \frac{PCLK \times 10^6}{S \times 2^{2n+1} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times S \times 2^{2n+1} \times (N+1)} - 1 \right\} \times 100$
Simple I²C*1				$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	

B: Bit rate (bps)
N: BRR setting for on-chip baud rate generator ($0 \le N \le 255$)
PCLK: Operating frequency (MHz)
n and S: Determined by the settings of the SMR and SCMR registers as listed in Table 32.13 and Table 32.14.
Note 1. Adjust the bit rate so that the widths at high and low level of the SCL output in simple I²C mode satisfy the I²C-bus standard.

Table 32.11 Relationship between N Setting in the BRR Register and Bit Rate B (for SCI12)

Mode	SEMR Settings		BRR Setting	Error (%)
	BGDM bit	ABCS bit		
Asynchronous, multi-processor communication	0	0	$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 64 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
	0	1	$N = \frac{PCLK \times 10^6}{32 \times 2^{2n-1} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 32 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
	1	0		
	1	1	$N = \frac{PCLK \times 10^6}{16 \times 2^{2n-1} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 16 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
Clock synchronous, simple SPI			$N = \frac{PCLK \times 10^6}{8 \times 2^{2n-1} \times B} - 1$	
Smart card interface			$N = \frac{PCLK \times 10^6}{S \times 2^{2n+1} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times S \times 2^{2n+1} \times (N+1)} - 1 \right\} \times 100$
Simple I^2C*1			$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	

B: Bit rate (bps)
N: BRR setting for on-chip baud rate generator ($0 \le N \le 255$)
PCLK: Operating frequency (MHz)
n and S: Determined by the settings of the SMR and SCMR registers as listed in Table 32.13 and Table 32.14.
Note 1. Adjust the bit rate so that the widths at high and low level of the SCL output in simple I^2C mode satisfy the I^2C-bus standard.

Table 32.12 Calculating Widths at High and Low Level for SCL

Mode	SCL	Formula (Result in Seconds)
I^2C	High period (minimum value)	$(N+1) \times 4 \times 2^{2n-1} \times 7 \times \frac{1}{PCLK \times 10^6}$
	Low period (minimum value)	$(N+1) \times 4 \times 2^{2n-1} \times 8 \times \frac{1}{PCLK \times 10^6}$

Table 32.13 Clock Source Settings

SMR.CKS[1:0] Bit Setting	Clock Source	n
0 0	PCLK	0
0 1	PCLK/4	1
1 0	PCLK/16	2
1 1	PCLK/64	3

Table 32.14 Base Clock Settings in Smart Card Interface Mode

SCMR.BCP2 Bit Setting	SMR.BCP[1:0] Bit Setting	Base Clock Cycles for 1-bit Period	S
0	0 0	93 clock cycles	93
0	0 1	128 clock cycles	128
0	1 0	186 clock cycles	186
0	1 1	512 clock cycles	512
1	0 0	32 clock cycles	32
1	0 1	64 clock cycles	64
1	1 0	372 clock cycles	372
1	1 1	256 clock cycles	256

Table 32.15 lists examples of N settings in the BRR register in normal asynchronous mode. Table 32.17 lists the maximum bit rate settable for each operating frequency. Examples of BRR (N) settings in clock synchronous mode and simple SPI mode are listed in Table 32.20. Examples of BRR (N) settings in smart card interface mode are listed in Table 32.22. Examples of BRR (N) settings in simple I^2C mode are listed in Table 32.24. In smart card interface mode, the number of base clock cycles S in a 1-bit data transfer time can be selected. For details, refer to section 32.6.4, Receive Data Sampling Timing and Reception Margin. Table 32.18 and Table 32.21 list the maximum bit rates with external clock input.

When either the SEMR.ABCS or BGDM bit is set to 1 in asynchronous mode, the bit rate becomes twice that listed in Table 32.15. When both of those bits are set to 1, the bit rate becomes four times the listed value.

Table 32.15 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode)

Bit Rate (bps)	Operating Frequency PCLK (MHz)														
	8			9.8304			10			12			12.288		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	141	0.03	2	174	–0.26	2	177	–0.25	2	212	0.03	2	217	0.08
150	2	103	0.16	2	127	0.00	2	129	0.16	2	155	0.16	2	159	0.00
300	1	207	0.16	1	255	0.00	2	64	0.16	2	77	0.16	2	79	0.00
600	1	103	0.16	1	127	0.00	1	129	0.16	1	155	0.16	1	159	0.00
1200	0	207	0.16	0	255	0.00	1	64	0.16	1	77	0.16	1	79	0.00
2400	0	103	0.16	0	127	0.00	0	129	0.16	0	155	0.16	0	159	0.00
4800	0	51	0.16	0	63	0.00	0	64	0.16	0	77	0.16	0	79	0.00
9600	0	25	0.16	0	31	0.00	0	32	–1.36	0	38	0.16	0	39	0.00
19200	0	12	0.16	0	15	0.00	0	15	1.73	0	19	–2.34	0	19	0.00
31250	0	7	0.00	0	9	–1.70	0	9	0.00	0	11	0.00	0	11	2.40
38400	—	—	—	0	7	0.00	0	7	1.73	0	9	–2.34	0	9	0.00

Bit Rate (bps)	Operating Frequency PCLK (MHz)														
	14			16			17.2032			18			19.6608		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	248	–0.17	3	70	0.03	3	75	0.48	3	79	–0.12	3	86	0.31
150	2	181	0.16	2	207	0.16	2	223	0.00	2	233	0.16	2	255	0.00
300	2	90	0.16	2	103	0.16	2	111	0.00	2	116	0.16	2	127	0.00
600	1	181	0.16	1	207	0.16	1	223	0.00	1	233	0.16	1	255	0.00
1200	1	90	0.16	1	103	0.16	1	111	0.00	1	116	0.16	1	127	0.00
2400	0	181	0.16	0	207	0.16	0	223	0.00	0	233	0.16	0	255	0.00
4800	0	90	0.16	0	103	0.16	0	111	0.00	0	116	0.16	0	127	0.00
9600	0	45	–0.93	0	51	0.16	0	55	0.00	0	58	–0.69	0	63	0.00
19200	0	22	–0.93	0	25	0.16	0	27	0.00	0	28	1.02	0	31	0.00
31250	0	13	0.00	0	15	0.00	0	16	1.20	0	17	0.00	0	19	–1.70
38400	—	—	—	0	12	0.16	0	13	0.00	0	14	–2.34	0	15	0.00

Bit Rate (bps)	Operating Frequency PCLK (MHz)														
	20			25			30			33			40		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	88	–0.25	3	110	–0.02	3	132	0.13	3	145	0.33	3	177	–0.25
150	3	64	0.16	3	80	0.47	3	97	–0.35	3	106	0.39	3	129	0.16
300	2	129	0.16	2	162	–0.15	2	194	0.16	2	214	–0.07	3	64	0.16
600	2	64	0.16	2	80	0.47	2	97	–0.35	2	106	0.39	2	129	0.16
1200	1	129	0.16	1	162	–0.15	1	194	0.16	1	214	–0.07	2	64	0.16
2400	1	64	0.16	1	80	0.47	1	97	–0.35	1	106	0.39	1	129	0.16
4800	0	129	0.16	0	162	–0.15	0	194	0.16	0	214	–0.07	1	64	0.16
9600	0	64	0.16	0	80	0.47	0	97	–0.35	0	106	0.39	0	129	0.16
19200	0	32	–1.36	0	40	–0.76	0	48	–0.35	0	53	–0.54	0	64	0.16
31250	0	19	0.00	0	24	0.00	0	29	0.00	0	32	0.00	0	39	0.00
38400	0	15	1.73	0	19	1.73	0	23	1.73	0	26	–0.54	0	32	–1.36

Bit Rate (bps)	Operating Frequency PCLK (MHz)					
	50			60		
	n	N	Error (%)	n	N	Error (%)
110	3	221	–0.02			
150	3	162	–0.15	3	194	0.16
300	3	80	0.47	3	97	–0.35
600	2	162	–0.15	3	48	–0.35
1200	2	80	0.47	2	97	–0.35
2400	1	162	–0.15	2	48	–0.35
4800	1	80	0.47	1	97	–0.35
9600	0	162	–0.15	1	48	–0.35
19200	0	80	0.47	0	97	–0.35
31250	0	49	0.00	0	59	0.00
38400	0	40	–0.76	0	48	–0.35

Note: This is an example when the ABCS, ABCSE, and BGDM bits in the SEMR register are 0.
When either the ABCS bit or BGDM bit is set to 1, the bit rate doubles.
When both ABCS and BGDM bits in the SEMR register are set to 1, the bit rate increases four times.
When the ABCSE bit is set to 1, the bit rate increases 16/3 times.

Table 32.16 Maximum Bit Rate for Each Operating Frequency (Asynchronous Mode) (for SCI1, SCI5, and SCI6)

PCLK (MHz)	SEMR Settings					Maximum Bit Rate (bps)	PCLK (MHz)	SEMR Settings					Maximum Bit Rate (bps)
	BGDM Bit	ABCS Bit	ABCSE Bit	n	N			BGDM Bit	ABCS Bit	ABCSE Bit	n	N	
8	0	0	0	0	0	250000	19.6608	0	0	0	0	0	614400
		1	0	0	0	500000			1	0	0	0	1228800
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1000000			1	0	0	0	2457600
	0 or 1	0 or 1	1	0	0	1333333		0 or 1	0 or 1	1	0	0	3276800
9.8304	0	0	0	0	0	307200	20	0	0	0	0	0	625000
		1	0	0	0	614400			1	0	0	0	1250000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1228800			1	0	0	0	2500000
	0 or 1	0 or 1	1	0	0	1638400		0 or 1	0 or 1	1	0	0	3333333
10	0	0	0	0	0	312500	25	0	0	0	0	0	781250
		1	0	0	0	625000			1	0	0	0	1562500
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1250000			1	0	0	0	3125000
	0 or 1	0 or 1	1	0	0	1666667		0 or 1	0 or 1	1	0	0	4166667
12	0	0	0	0	0	375000	30	0	0	0	0	0	937500
		1	0	0	0	750000			1	0	0	0	1875000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1500000			1	0	0	0	3750000
	0 or 1	0 or 1	1	0	0	2000000		0 or 1	0 or 1	1	0	0	5000000
12.288	0	0	0	0	0	384000	33	0	0	0	0	0	1031250
		1	0	0	0	768000			1	0	0	0	2062500
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1536000			1	0	0	0	4125000
	0 or 1	0 or 1	1	0	0	2048000		0 or 1	0 or 1	1	0	0	5500000
14	0	0	0	0	0	437500	40	0	0	0	0	0	1250000
		1	0	0	0	875000			1	0	0	0	2500000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1750000			1	0	0	0	5000000
	0 or 1	0 or 1	1	0	0	2333333		0 or 1	0 or 1	1	0	0	6666667
16	0	0	0	0	0	500000	50	0	0	0	0	0	1562500
		1	0	0	0	1000000			1	0	0	0	3125000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	2000000			1	0	0	0	6250000
	0 or 1	0 or 1	1	0	0	2666667		0 or 1	0 or 1	1	0	0	8333333
17.2032	0	0	0	0	0	537600	60	0	0	0	0	0	1875000
		1	0	0	0	1075200			1	0	0	0	3750000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	2150400			1	0	0	0	7500000
	0 or 1	0 or 1	1	0	0	2867200		0 or 1	0 or 1	1	0	0	10000000
18	0	0	0	0	0	562500							
		1	0	0	0	1125000							
	1	0	0	0	0								
		1	0	0	0	2250000							

Table 32.17 Maximum Bit Rate for Each Operating Frequency (Asynchronous Mode) (for SCI12)

	SEMR Settings						SEMR Settings				
PCLK (MHz)	BGDM Bit	ABCS Bit	n	N	Maximum Bit Rate (bps)	PCLK (MHz)	BGDM Bit	ABCS Bit	n	N	Maximum Bit Rate (bps)
8	0	0	0	0	250000	19.6608	0	0	0	0	614400
		1	0	0	500000			1	0	0	1228800
	1	0	0	0			1	0	0	0	
		1	0	0	1000000			1	0	0	2457600
9.8304	0	0	0	0	307200	20	0	0	0	0	625000
		1	0	0	614400			1	0	0	1250000
	1	0	0	0			1	0	0	0	
		1	0	0	1228800			1	0	0	2500000
10	0	0	0	0	312500	25	0	0	0	0	781250
		1	0	0	625000			1	0	0	1562500
	1	0	0	0			1	0	0	0	
		1	0	0	1250000			1	0	0	3125000
12	0	0	0	0	375000	30	0	0	0	0	937500
		1	0	0	750000			1	0	0	1875000
	1	0	0	0			1	0	0	0	
		1	0	0	1500000			1	0	0	3750000
12.288	0	0	0	0	384000	33	0	0	0	0	1031250
		1	0	0	768000			1	0	0	2062500
	1	0	0	0			1	0	0	0	
		1	0	0	1536000			1	0	0	4125000
14	0	0	0	0	437500	40	0	0	0	0	1250000
		1	0	0	875000			1	0	0	2500000
	1	0	0	0			1	0	0	0	
		1	0	0	1750000			1	0	0	5000000
16	0	0	0	0	500000	50	0	0	0	0	1562500
		1	0	0	1000000			1	0	0	3125000
	1	0	0	0			1	0	0	0	
		1	0	0	2000000			1	0	0	6250000
17.2032	0	0	0	0	537600	60	0	0	0	0	1875000
		1	0	0	1075200			1	0	0	3750000
	1	0	0	0			1	0	0	0	
		1	0	0	2150400			1	0	0	7500000
18	0	0	0	0	562500						
		1	0	0	1125000						
	1	0	0	0							
		1	0	0	2250000						

Table 32.18 Maximum Bit Rate with External Clock Input (Asynchronous Mode)

		Maximum Bit Rate (bps)	
PCLK (MHz)	External Input Clock (MHz)	SEMR.ABCS Bit = 0	SEMR.ABCS Bit = 1
8	2.0000	125000	250000
9.8304	2.4576	153600	307200
10	2.5000	156250	312500
12	3.0000	187500	375000
12.288	3.0720	192000	384000
14	3.5000	218750	437500
16	4.0000	250000	500000
17.2032	4.3008	268800	537600
18	4.5000	281250	562500
19.6608	4.9152	307200	614400
20	5.0000	312500	625000
25	6.2500	390625	781250
30	7.5000	468750	937500
33	8.2500	515625	1031250
40	10.0000	625000	1250000
50	12.5000	781250	1562500
60	15.0000	937500	1875000

Table 32.19 Maximum Bit Rate with TMR Clock Input (Asynchronous Mode)

		Maximum Bit Rate (bps)	
PCLK (MHz)	TMR Clock (MHz)	SEMR.ABCS Bit = 0	SEMR.ABCS Bit = 1
8	4	250000	500000
9.8304	4.9152	307200	614400
10	5	312500	625000
12	6	375000	750000
12.288	6.144	384000	768000
14	7	437500	875000
16	8	500000	1000000
17.2032	8.6016	537600	1075200
18	9	562500	1125000
19.6608	9.8304	614400	1228800
20	10	625000	1250000
25	12.5	781250	1562500
30	15	937500	1875000
33	16.5	1031250	2062500
40	20	1250000	2500000
50	25	1562500	3125000
60	30	1875000	3750000

Table 32.20 BRR Settings for Various Bit Rates (Clock Synchronous Mode, Simple SPI Mode)

Bit Rate (bps)	Operating Frequency PCLK (MHz)																			
	8		10		16		20		25		30		33		40		50		60	
	n	N	n	N	n	N	n	N	n	N	n	N	n	N	n	N	n	N	n	N
110																				
250	3	124	3	155	3	249														
500	2	249	3	77	3	124	3	155	3	194	3	233	3	255						
1 k	2	124	2	155	2	249	3	77	3	97	3	116	3	128	3	155	3	194	3	233
2.5 k	1	199	1	249	2	99	2	124	2	155	2	187	2	205	2	249	3	77	3	93
5 k	1	99	1	124	1	199	1	249	2	77	2	93	2	102	2	124	2	155	3	46
10 k	0	199	0	249	1	99	1	124	1	155	1	187	1	205	1	249	2	77	2	93
25 k	0	79	0	99	0	159	0	199	0	249	1	74	1	82	1	99	1	124	1	149
50 k	0	39	0	49	0	79	0	99	0	124	0	149	0	164	1	49	1	61	1	74
100 k	0	19	0	24	0	39	0	49	0	62	0	74	0	82	0	99	0	124	0	149
250 k	0	7	0	9	0	15	0	19	0	24	0	29	0	32	0	39	0	49	0	59
500 k	0	3	0	4	0	7	0	9	—	—	0	14	—	—	0	19	0	24	0	29
1 M	0	1			0	3	0	4	—	—			—	—	0	9	—	—	0	14
2.5 M			0	0*1			0	1			0	2			0	3	0	4	0	5
5 M							0	0*1							0	1			0	2
7.5 M											0	0*1							0	1

Blank cell: Cannot be set since the bit rate error exceeds 5%.

—: Can be set, but a bit rate error of 1 to 5% will occur.

Note 1. Continuous transmission or reception is not possible. After transmitting/receiving one frame of data, there is an interval of a 1-bit period before starting transmitting/receiving the next frame of data. The output of the synchronization clock is stopped for a 1-bit period. For this reason, it takes 9 bits worth of time to transfer one frame (8 bits) of data, and the average transfer rate is 8/9 times the bit rate.

Table 32.21 Maximum Bit Rate with External Clock Input (Clock Synchronous Mode, Simple SPI Mode)

PCLK (MHz)	External Input Clock (MHz)	Maximum Bit Rate (Mbps)
8	1.3333	1.3333
10	1.6667	1.6667
12	2.0000	2.0000
14	2.3333	2.3333
16	2.6667	2.6667
18	3.0000	3.0000
20	3.3333	3.3333
25	4.1667	4.1667
30	5.0000	5.0000
33	5.5000	5.5000
40	6.6667	6.6667
50	8.3333	8.3333
60	10.0000	10.0000

Table 32.22 BRR Settings for Various Bit Rates (Smart Card Interface Mode, n = 0, S = 372)

Bit Rate (bps)	PCLK (MHz)	n	N	Error (%)
9600	7.1424	0	0	0.00
	10.00	0	1	–30.00
	10.7136	0	1	–25.00
	13.00	0	1	–8.99
	14.2848	0	1	0.00
	16.00	0	1	12.01
	18.00	0	2	–15.99
	20.00	0	2	–6.66
	25.00	0	3	–12.49
	30.00	0	3	5.01
	33.00	0	4	–7.59
	40.00	0	5	–6.66
	50.00	0	6	0.01
	60.00	0	7	5.01

Table 32.23 Maximum Bit Rate for Each Operating Frequency (Smart Card Interface Mode, S = 32)

PCLK (MHz)	Maximum Bit Rate (bps)	n	N
10.00	156250	0	0
10.7136	167400	0	0
13.00	203125	0	0
16.00	250000	0	0
18.00	281250	0	0
20.00	312500	0	0
25.00	390625	0	0
30.00	468750	0	0
33.00	515625	0	0
40.00	625000	0	0
50.00	781250	0	0
60.00	937500	0	0

Table 32.24 BRR Settings for Various Bit Rates (Simple I²C Mode)

Bit Rate (bps)	Operating Frequency PCLK (MHz)														
	8			10			16			20			25		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
10 k	0	24	0.0	0	31	–2.3	1	12	–3.8	1	15	–2.3	1	19	–2.3
25 k	0	9	0.0	0	12	–3.8	1	4	0.0	1	6	–10.7	1	7	–2.3
50 k	0	4	0.0	0	6	–10.7	1	2	–16.7	1	3	–21.9	1	3	–2.3
100 k	0	2	–16.7	0	3	–21.9	0	4	0.0	0	6	–10.7	1	1	–2.3
250 k	0	0	0.0	0	1	–37.5	0	1	0.0	0	2	–16.7	0	3	–21.9
350 k										0	1	–10.7	0	2	–25.6

Bit Rate (bps)	Operating Frequency PCLK (MHz)														
	30			33			40			50			60		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
10 k	1	23	–2.3	1	25	–0.8	0	124	0.00	2	9	–2.3	1	46	–0.27
25 k	1	9	–6.3	1	10	–6.3	0	40	0.00	2	3	–2.3	0	74	0.00
50 k	1	4	–6.3	1	5	–14.1	0	24	0.00	2	1	–2.3	0	37	–1.32
100 k	1	2	–21.9	1	2	–14.1	0	12	–3.85	1	3	–2.3	0	18	–1.32
250 k	0	3	–6.3	0	4	–17.5	0	4	0.00	0	6	–10.7	0	7	–6.25
350 k	0	2	–10.7	0	2	–1.8	0	3	–10.71	0	4	–10.7	0	4	7.14

Table 32.25 Minimum Widths at High and Low Level for SCL at Various Bit Rates (Simple I²C Mode)

Bit Rate (bps)	Operating Frequency PCLK (MHz)											
	8			10			16			20		
	n	N	Min. Widths at High/Low Level for SCL (µs)	n	N	Min. Widths at High/Low Level for SCL (µs)	n	N	Min. Widths at High/Low Level for SCL (µs)	n	N	Min. Widths at High/Low Level for SCL (µs)
10 k	0	24	43.75/50.00	0	31	44.80/51.20	1	12	45.50/52.00	1	15	44.80/51.20
25 k	0	9	17.50/20.00	0	12	18.20/20.80	1	4	17.50/20.00	1	6	19.60/22.40
50 k	0	4	8.75/10.00	0	6	9.80/11.20	1	2	10.50/12.00	1	3	11.20/12.80
100 k	0	2	5.25/6.00	0	3	5.60/6.40	0	4	4.37/5.00	0	6	4.90/5.60
250 k	0	0	1.75/2.00	0	1	2.80/3.20	0	1	1.75/2.00	0	2	2.10/2.40
350 k										0	1	1.40/1.60

Bit Rate (bps)	Operating Frequency PCLK (MHz)											
	25			30			33			40		
	n	N	Min. Widths at High/Low Level for SCL (µs)	n	N	Min. Widths at High/Low Level for SCL (µs)	n	N	Min. Widths at High/Low Level for SCL (µs)	n	N	Min. Widths at High/Low Level for SCL (µs)
10 k	1	19	44.80/51.20	1	23	44.80/51.20	1	25	44.12/50.42	1	32	46.20/52.80
25 k	1	7	17.92/20.48	1	9	18.66/21.33	1	10	18.66/21.33	1	12	18.20/20.80
50 k	1	3	8.96/10.24	1	4	9.33/10.66	1	5	10.18/11.63	1	6	9.80/11.20
100 k	1	1	4.48/5.12	1	2	5.60/6.40	1	2	5.09/5.81	0	13	4.90/5.60
250 k	0	3	2.24/2.56	0	3	1.86/2.13	0	4	2.12/2.42	0	4	1.75/2.00
350 k	0	2	1.68/1.92	0	2	1.40/1.60	0	2	1.27/1.45	0	3	1.40/1.60

Bit Rate (bps)	Operating Frequency PCLK (MHz)					
	50			60		
	n	N	Min. Widths at High/Low Level for SCL (µs)	n	N	Min. Widths at High/Low Level for SCL (µs)
10 k	2	9	44.80/51.20	3	2	44.80/51.20
25 k	2	3	17.92/20.48	2	4	18.67/21.33
50 k	2	1	8.96/10.24	1	9	9.33/10.67
100 k	1	3	4.48/5.12	1	4	4.67/5.33
250 k	0	6	1.96/2.24	0	7	1.87/2.13
350 k	0	4	1.40/1.60	0	5	1.40/1.60

32.2.12 Modulation Duty Register (MDDR)

Address(es): SCI1.MDDR 0008 A032h, SCI5.MDDR 0008 A0B2h, SCI6.MDDR 0008 A0D2h, SCI12.MDDR 0008 B312h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1

The MDDR register corrects the bit rate adjusted by the BRR register.
When the SEMR.BRME bit is set to 1, the bit rate generated by the on-chip baud rate generator is evenly corrected according to the settings of the MDDR register (M/256). The relationship between the MDDR register setting (M) and the bit rate (B) is given in Table 32.26 and Table 32.27.
The range of the value that can be set in the MDDR register is from 80h to FFh. A value other than these cannot be set.
The MDDR register is writable only when the TE and RE bits in the SCR register are 0.

Table 32.26 Relationship between MDDR Setting (M) and Bit Rate (B) When Bit Rate Modulation Function is Used (for SCI1, SCI5, and SCI6)

Mode	SEMR Settings			BRR Setting	Error (%)
	BGDM Bit	ABCS Bit	ABCSE Bit		
Asynchronous, multi-processor communication	0	0	0	$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 64 \times 2^{2n-1} \times \frac{256}{M} \times (N+1)} - 1 \right\} \times 100$
	1	0	0	$N = \frac{PCLK \times 10^6}{32 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 32 \times 2^{2n-1} \times \frac{256}{M} \times (N+1)} - 1 \right\} \times 100$
	0	1	0		
	1	1	0	$N = \frac{PCLK \times 10^6}{16 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 16 \times 2^{2n-1} \times \frac{256}{M} \times (N+1)} - 1 \right\} \times 100$
	0 or 1	0 or 1	1	$N = \frac{PCLK \times 10^6}{12 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 12 \times 2^{2n-1} \times \frac{256}{M} \times (N+1)} - 1 \right\} \times 100$
Clock synchronous, simple SPI*1				$N = \frac{PCLK \times 10^6}{8 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	
Smart card interface				$N = \frac{PCLK \times 10^6}{S \times 2^{2n+1} \times \frac{256}{M} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times S \times 2^{2n+1} \times \frac{256}{M} \times (N+1)} - 1 \right\} \times 100$
Simple I²C*2				$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	

B: Bit rate (bps)
M: MDDR setting (128 ≤ MDDR ≤ 256)
N: BRR setting for baud rate generator (0 ≤ N ≤ 255)
PCLK: Operating frequency (MHz)
n and S: Determined by the settings of the SMR and SCMR registers as listed in Table 32.13 and Table 32.14, section 32.2.11, Bit Rate Register (BRR).

Note 1. Do not use this function in clock synchronous mode and in the highest speed settings in simple SPI mode (SMR.CKS[1:0] =

00b, SCR.CKE[1] = 0, and BRR = 0).
Note 2. Adjust the bit rate so that the widths at high and low level of the SCL output in simple I²C mode satisfy the I²C-bus standard.

Table 32.27 Relationship between MDDR Setting (M) and Bit Rate (B) When Bit Rate Modulation Function is Used (for SCI12)

Mode	SEMR Settings		BRR Setting	Error (%)
	BGDM Bit	ABCS Bit		
Asynchronous, multi-processor communication	0	0	$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 64 \times 2^{2n-1} \times \frac{256}{M} \times (N+1)} - 1 \right\} \times 100$
	1	0	$N = \frac{PCLK \times 10^6}{32 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 32 \times 2^{2n-1} \times \frac{256}{M} \times (N+1)} - 1 \right\} \times 100$
	0	1		
	1	1	$N = \frac{PCLK \times 10^6}{16 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 16 \times 2^{2n-1} \times \frac{256}{M} \times (N+1)} - 1 \right\} \times 100$
Clock synchronous, simple SPI*1			$N = \frac{PCLK \times 10^6}{8 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	
Smart card interface			$N = \frac{PCLK \times 10^6}{S \times 2^{2n+1} \times \frac{256}{M} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times S \times 2^{2n+1} \times \frac{256}{M} \times (N+1)} - 1 \right\} \times 100$
Simple I²C*2			$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	

B: Bit rate (bps)
M: MDDR setting (128 ≤ MDDR ≤ 256)
N: BRR setting for baud rate generator (0 ≤ N ≤ 255)
PCLK: Operating frequency (MHz)
n and S: Determined by the settings of the SMR and SCMR registers as listed in Table 32.13 and Table 32.14, section 32.2.11, Bit Rate Register (BRR).
Note 1. Do not use this function in clock synchronous mode and in the highest speed settings in simple SPI mode (SMR.CKS[1:0] = 00b, SCR.CKE[1] = 0, and BRR = 0).
Note 2. Adjust the bit rate so that the widths at high and low level of the SCL output in simple I²C mode satisfy the I²C-bus standard.

Smaller settings of the SMR.CKS[1:0] bits and larger settings of the BRR register reduce difference in the length of the 1-bit period.

32.2.13 Serial Extended Mode Register (SEMR)

Address(es): SCI1.SEMR 0008 A027h, SCI5.SEMR 0008 A0A7h, SCI6.SEMR 0008 A0C7h, SCI12.SEMR 0008 B307h

	b7	b6	b5	b4	b3	b2	b1	b0
	RXDESEL	BGDM	NFEN	ABCS	ABCSE	BRME	ITE	ACS0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ACS0	Asynchronous Mode Clock Source Select	(Valid only in asynchronous mode) 0: External clock input 1: Logical AND of two compare matches output from TMR (valid for SCI5, SCI6, and SCI12 only) Available compare match output varies per SCI channel.	R/W*1
b1	ITE	Instant Transmission Enable*2	(Valid only in asynchronous mode) 0: Internal wait time is inserted after the transmission is enabled 1: Data transmission is started immediately after the transmission is enabled.	R/W*1
b2	BRME	Bit Rate Modulation Enable	0: Bit rate modulation function is disabled. 1: Bit rate modulation function is enabled.	R/W*1
b3	ABCSE	Asynchronous Mode Base Clock Select Extended*2	(Valid only when using a on-chip baud rate generator in asynchronous mode) 0: The number of clock cycles for 1-bit period depends on the BGDM and ABCS bits setting. 1: Selects 6 base clock cycles for 1-bit period.	R/W*1
b4	ABCS	Asynchronous Mode Base Clock Select	(Valid only in asynchronous mode) 0: Selects 16 base clock cycles for 1-bit period. 1: Selects 8 base clock cycles for 1-bit period.	R/W*1
b5	NFEN	Digital Noise Filter Function Enable	(In asynchronous mode) 0: Noise cancellation function for the RXDn input signal is disabled. 1: Noise cancellation function for the RXDn input signal is enabled. (in simple I^2C mode) 0: Noise cancellation function for the SSCLn and SSDAn input signals is disabled. 1: Noise cancellation function for the SSCLn and SSDAn input signals is enabled. The NFEN bit should be 0 in any mode other than above.	R/W*1
b6	BGDM	Baud Rate Generator Double-Speed Mode Select	(Only valid the SCR.CKE[1] bit is 0 in asynchronous mode) 0: Baud rate generator outputs the clock with normal frequency. 1: Baud rate generator outputs the clock with doubled frequency.	R/W*1
b7	RXDESEL	Asynchronous Start Bit Edge Detection Select	(Valid only in asynchronous mode) 0: The low level on the RXDn pin is detected as the start bit. 1: A falling edge on the RXDn pin is detected as the start bit.	R/W*1

Note 1. Writable only when the SCR.TE bit is 0 and the SCR.RE bit is 0 (both serial transmission and reception are disabled).
Note 2. This bit is reserved for SCI12. It is read as 0. The write value should be 0.

The SEMR register is used to select a clock source for 1-bit period in asynchronous mode or a detection method of the start bit.

ACS0 Bit (Asynchronous Mode Clock Source Select)

Selects the clock source in the asynchronous mode.

The ACS0 bit is valid in asynchronous mode (SMR.CM bit = 0) and when an external clock input is selected (SCR.CKE[1:0] bits = 10b or 11b). This bit is used to select an external clock input or the logical AND of compare matches output from the internal TMR.

Set the ACS0 bit to 0 in other than asynchronous mode.

For SCI5, SCI6, and SCI12, the TMOn output (n = 0 to 3) of TMR units 0 and 1 can be set as the base clock source. Refer to Table 32.28 for details.

The ACS0 bit for SCI1 is reserved. The write value to this bit for SCI1 should be 0.

Table 32.28 Correspondence between SCI Channels and Compare Match Outputs

SCI	TMR	Compare Match Output
SCI5	Unit 0	TMO0, TMO1
SCI6	Unit 1	TMO2, TMO3
SCI12	Unit 0	TMO0, TMO1

Figure 32.4 shows a setting example of when TMO0 and TMO1 in the TMR unit 0 are selected for output.

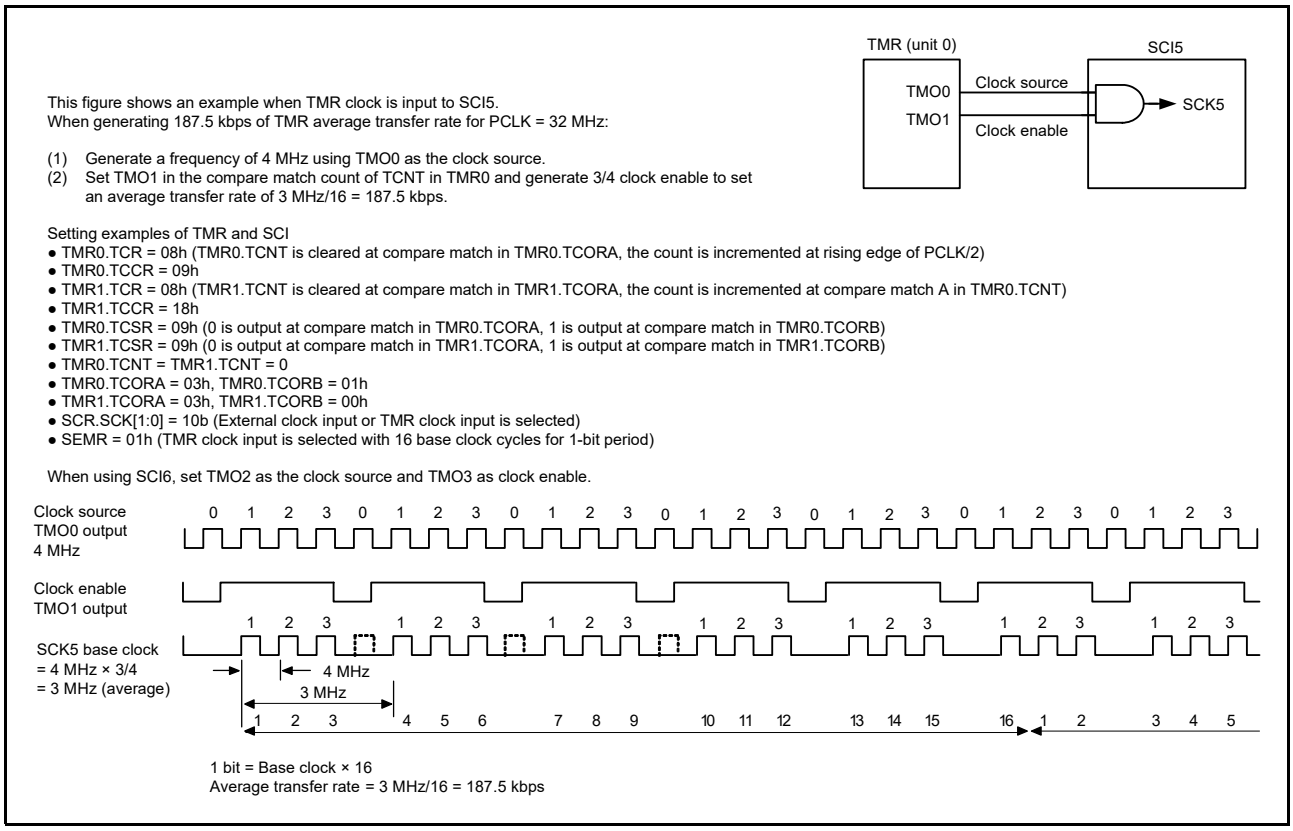


Figure 32.4 Example of Average Transfer Rate Setting When TMR Clock is Input

ITE Bit (Instant Transmission Enable)

This bit is used to start data transmission without internal wait time in asynchronous mode. When the TE bit is set to 1 while this bit is 0, one frame of internal wait time is inserted before the data transmission is started. When this bit is 1, the data transmission is started immediately after the TE bit is set to 1.

BRME Bit (Bit Rate Modulation Enable)

This bit enables and disables the bit rate modulation function. The bit rate generated by on-chip baud rate generator is evenly corrected when this function is enabled.

ABCSE Bit (Asynchronous Mode Base Clock Select Extended*2)

When setting this bit to 1, the 1-bit period becomes 6 cycles of the base clock and the clock of doubled frequency is output from the baud rate generator.
This bit is valid when the on-chip baud rate generator is selected as the clock source (SCR.CKE[1] = 0) in asynchronous mode (SMR.CM = 0).
When the bit rate is made to be 1/6 of PCLK frequency, set this bit to 1, the SMR.CKS[1:0] bits to 00b, and the BRR register to 00h.

NFEN Bit (Digital Noise Filter Function Enable)

This bit enables or disables the digital noise filter function.
When the function is enabled, noise cancellation is applied to the RXDn input signal in asynchronous mode, and noise cancellation is applied to the SSDAn and SSCLn input signals in simple I^2C mode.
In any mode other than above, set the NFEN bit to 0 to disable the digital noise filter function.
When the function is disabled, input signals are transferred as is, as internal signals.

BGDM Bit (Baud Rate Generator Double-Speed Mode Select)

Selects the cycle of output clock for the baud rate generator.
This bit is valid when the on-chip baud rate generator is selected as the clock source (SCR.CKE[1] = 0) in asynchronous mode (SMR.CM = 0). For the clock output from the baud rate generator, either normal or doubled frequency can be selected. The base clock is generated by the clock output from the baud rate generator. When the BGDM bit is set to 1, the base clock cycle is halved and the bit rate is doubled.
Set this bit to 0 in modes other than asynchronous mode.

RXDESEL Bit (Asynchronous Start Bit Edge Detection Select)

Selects the detection method of the start bit for reception in asynchronous mode. When a break occurs, data receiving operation depends on the settings of this bit. Set this bit to 1 when reception should be stopped while a break occurs or when reception should be started without retaining the RXDn pin input at high level for the period of one data frame or longer after completion of the break.
Set this bit to 0 in modes other than asynchronous mode.

32.2.14 Noise Filter Setting Register (SNFR)

Address(es): SCI1.SNFR 0008 A028h, SCI5.SNFR 0008 A0A8h, SCI6.SNFR 0008 A0C8h, SCI12.SNFR 0008 B308h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	NFCS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	NFCS[2:0]	Noise Filter Clock Select	In asynchronous mode, the standard setting for the base clock is as follows. b2 b0 0 0 0: The clock signal divided by 1 is used with the noise filter. In simple I2C mode, the standard settings for the clock source of the on-chip baud rate generator selected by the SMR.CKS[1:0] bits are given below. b2 b0 0 0 1: The clock signal divided by 1 is used with the noise filter. 0 1 0: The clock signal divided by 2 is used with the noise filter. 0 1 1: The clock signal divided by 4 is used with the noise filter. 1 0 0: The clock signal divided by 8 is used with the noise filter. Settings other than above are prohibited.	R/W*1
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing to these bits is only possible when the RE and TE bits in the SCR register are 0 (serial reception and transmission disabled).

NFCS[2:0] Bits (Noise Filter Clock Select)

These bits select the sampling clock for the digital noise filter. To use the noise filter in asynchronous mode, set these bits to 000b. In simple I2C mode, set the bits to a value in the range from 001b to 100b.

32.2.15 I²C Mode Register 1 (SIMR1)

Address(es): SCI1.SIMR1 0008 A029h, SCI5.SIMR1 0008 A0A9h, SCI6.SIMR1 0008 A0C9h, SCI12.SIMR1 0008 B309h

	b7	b6	b5	b4	b3	b2	b1	b0
	IICDL[4:0]					—	—	IICM
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IICM	Simple I²C Mode Select	SMIF IICM 0 0: Asynchronous mode, Multi-processor mode, Clock synchronous mode (in asynchronous mode, synchronous, or simple SPI mode) 0 1: Simple I²C mode 1 0: Smart card interface mode 1 1: Setting prohibited.	R/W*1
b2, b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7 to b3	IICDL[4:0]	SSDA Output Delay Select	(Cycles below are of the clock signal from the on-chip baud rate generator.) b7 b3 0 0 0 0 0: No output delay 0 0 0 0 1: 0 to 1 cycle 0 0 0 1 0: 1 to 2 cycles 0 0 0 1 1: 2 to 3 cycles 0 0 1 0 0: 3 to 4 cycles 0 0 1 0 1: 4 to 5 cycles : : 1 1 1 1 0: 29 to 30 cycles 1 1 1 1 1: 30 to 31 cycles	R/W*1

Note 1. Writing to these bits is only possible when the RE and TE bits in the SCR register are 0 (both serial transmission and reception are disabled).

SIMR1 is used to select simple I²C mode and the number of delay stages for the SSDA output.

IICM Bit (Simple I²C Mode Select)

In conjunction with the SCMR.SMIF bit, this bit selects the operating mode.

IICDL[4:0] Bits (SSDA Output Delay Select)

These bits are used to set a delay for output on the SSDAn pin relative to the falling edge of the output on the SSCLn pin. The available delay settings range from no delay to 31 cycles, with the clock signal from the on-chip baud rate generator as the base. The signal obtained by frequency-dividing PCLK by the divisor set in the SMR.CKS[1:0] bits is supplied as the clock signal from the on-chip baud rate generator. Set these bits to 00000b unless operation is in simple I²C mode. In simple I²C mode, set the bits to a value in the range from 00001b to 11111b.

32.2.16 I2C Mode Register 2 (SIMR2)

Address(es): SCI1.SIMR2 0008 A02Ah, SCI5.SIMR2 0008 A0AAh, SCI6.SIMR2 0008 A0CAh, SCI12.SIMR2 0008 B30Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	IICACKT	—	—	—	IICCSC	IICINTM
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IICINTM	I2C Interrupt Mode Select	0: Use ACK/NACK interrupts. 1: Use reception and transmission interrupts.	R/W*1
b1	IICCSC	Clock Synchronization	0: No synchronization with the clock signal 1: Synchronization with the clock signal	R/W*1
b4 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	IICACKT	ACK Transmission Data	0: ACK transmission 1: NACK transmission and reception of ACK/NACK	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing to these bits is only possible when the RE and TE bits in the SCR register are 0 (serial reception and transmission disabled).

SIMR2 is used to select how reception and transmission are controlled in simple I2C mode.

IICINTM Bit (I2C Interrupt Mode Select)

This bit selects the sources of interrupt requests in simple I2C mode.

IICCSC Bit (Clock Synchronization)

Set the IICCSC bit to 1 if the internally generated SSCLn clock signal is to be synchronized when the SSCLn pin has been placed at the low level in the case of a wait inserted by the other device, etc.

The SSCLn clock signal is not synchronized if the IICCSC bit is 0. The SSCLn clock signal is generated in accord with the rate selected in the BRR regardless of the level being input on the SSCLn pin.

Set the IICCSC bit to 1 except during debugging.

IICACKT Bit (ACK Transmission Data)

Transmitted data contains ACK bits. Set this bit to 1 when ACK and NACK bits are received.

32.2.17 I2C Mode Register 3 (SIMR3)

Address(es): SCI1.SIMR3 0008 A02Bh, SCI5.SIMR3 0008 A0ABh, SCI6.SIMR3 0008 A0CBh, SCI12.SIMR3 0008 B30Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	IICSCLS[1:0]		IICSDAS[1:0]		IICSTIF	IICSTP REQ	IICRST AREQ	IICSTA REQ
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IICSTAREQ	Start Condition Generation	0: A start condition is not generated. 1: A start condition is generated.*1, *3, *4, *5	R/W
b1	IICRSTAREQ	Restart Condition Generation	0: A restart condition is not generated. 1: A restart condition is generated.*2, *3, *4, *5	R/W
b2	IICSTPREQ	Stop Condition Generation	0: A stop condition is not generated. 1: A stop condition is generated.*2, *3, *4, *5	R/W
b3	IICSTIF	Issuing of Start, Restart, or Stop Condition Completed Flag	0: There are no requests for generating conditions or a condition is being generated. 1: A start, restart, or stop condition is completely generated.	R/W
b5, b4	IICSDAS[1:0]	SSDA Output Select	b5 b4 0 0: Serial data output 0 1: Generate a start, restart, or stop condition. 1 0: Output the low level on the SSDAn pin. 1 1: Place the SSDAn pin in the high-impedance state.	R/W
b7, b6	IICSCLS[1:0]	SSCL Output Select	b7 b6 0 0: Serial clock output 0 1: Generate a start, restart, or stop condition. 1 0: Output the low level on the SSCLn pin. 1 1: Place the SSCLn pin in the high-impedance state.	R/W

Note 1. Generate a start condition only when the SSCLn and SSDAn pins are both high (the corresponding bits in the corresponding PIDR registers are 1).
Note 2. Generate a restart or stop condition only when the SSCLn pin is low (the corresponding bit in the PIDR register is 0).
Note 3. Do not set more than one from among the IICSTAREQ, IICRSTAREQ, and IICSTPREQ bits to 1 at a given time.
Note 4. Execute the generation of a condition after the value of the IICSTIF flag is 0.
Note 5. Do not write 0 to this bit while it is 1. Generation of a condition is suspended by writing 0 to this bit while it is 1.

SIMR3 is used to control the simple I2C mode start and stop conditions, and to hold the SSDAn and SSCLn pins at fixed levels.

IICSTAREQ Bit (Start Condition Generation)

When a start condition is to be generated, set both the IICSDAS[1:0] and IICSCLS[1:0] bits to 01b as well as setting the IICSTAREQ bit to 1.

[Setting condition]

- Writing 1 to the bit

[Clearing condition]

- Completion of generation of the start condition

IICRSTAREQ Bit (Restart Condition Generation)

When a restart condition is to be generated, set both the IICSDAS[1:0] and IICSCLS[1:0] bits to 01b as well as setting the IICRSTAREQ bit to 1.

[Setting condition]

- Writing 1 to the bit

[Clearing condition]

- Completion of generation of the restart condition

IICSTPREQ Bit (Stop Condition Generation)

When a stop condition is to be generated, set both the IICSDAS[1:0] and IICSCLS[1:0] bits to 01b as well as setting the IICSTPREQ bit to 1.

[Setting condition]

- Writing 1 to the bit

[Clearing condition]

- Completion of generation of the stop condition

IICSTIF Flag (Issuing of Start, Restart, or Stop Condition Completed Flag)

After generating a condition, this bit indicates that the generation is completed. When using the IICSTAREQ, IICRSTAREQ, or IICSTPREQ bit to cause generation of a condition, do so after setting the IICSTIF flag to 0.

When the IICSTIF flag is 1 while an interrupt request is enabled by setting the SCR.TEIE bit, an STI request is output.

[Setting condition]

- Completion of the generation of a start, restart, or stop condition (however, in cases where this conflicts with any of the conditions for the flag becoming 0 listed below, the other condition takes precedence)

[Clearing conditions]

- Writing 0 to the bit (confirm that the IICSTIF flag is 0 before doing so)
- Writing 0 to the SIMR1.IICM bit (when operation is not in simple I^2C mode)
- Writing 0 to the SCR.TE bit

IICSDAS[1:0] Bits (SSDA Output Select)

These bits control output from the SSDAn pin.

Set the IICSDAS[1:0] and IICSCLS[1:0] bits to the same value during normal operations.

IICSCLS[1:0] Bits (SSCL Output Select)

These bits control output from the SSCLn pin.

Set the IICSCLS[1:0] and IICSDAS[1:0] bits to the same value during normal operations.

32.2.18 I2C Status Register (SISR)

Address(es): SCI1.SISR 0008 A02Ch, SCI5.SISR 0008 A0ACh, SCI6.SISR 0008 A0CCh, SCI12.SISR 0008 B30Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	IICACKR
Value after reset:	0	0	x	x	0	x	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	IICACKR	ACK Reception Data Flag	0: ACK received 1: NACK received	R/W*1
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	—	Reserved	The read value is undefined.	R
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5, b4	—	Reserved	The read value is undefined.	R
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only 0 can be written to this bit, to clear the flag.

SISR is used to monitor state in relation to simple I2C mode.

IICACKR Flag (ACK Reception Data Flag)

Received ACK and NACK bits can be read from this bit.
The IICACKR flag is updated at the rising of SSCLn clock for the ACK/NACK receiving bit.

32.2.19 SPI Mode Register (SPMR)

Address(es): SCI1.SPMR 0008 A02Dh, SCI5.SPMR 0008 A0ADh, SCI6.SPMR 0008 A0CDh, SCI12.SPMR 0008 B30Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	CKPH	CKPOL	—	MFF	—	MSS	CTSE	SSE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SSE	SSn# Pin Function Enable	0: SSn# pin function is disabled. 1: SSn# pin function is enabled.	R/W*1
b1	CTSE	CTS Enable	0: CTS function is disabled (RTS output function is enabled). 1: CTS function is enabled.	R/W*1
b2	MSS	Master Slave Select	0: Transmission is through the SMOSIn pin and reception is through the SMISOn pin (master mode). 1: Reception is through the SMOSIn pin and transmission is through the SMISOn pin (slave mode).	R/W*1
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	MFF	Mode Fault Flag	0: No mode fault error 1: Mode fault error	R/W*2
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	CKPOL	Clock Polarity Select	0: Clock polarity is not inverted. 1: Clock polarity is inverted.	R/W*1
b7	CKPH	Clock Phase Select	0: Clock is not delayed. 1: Clock is delayed.	R/W*1

Note 1. Writing to these bits is only possible when the RE and TE bits in the SCR register are 0 (both serial transmission and reception are disabled).
Note 2. Only 0 can be written to these bits, which clears the flag.

SPMR is used to select the extension settings in asynchronous and clock synchronous modes.

SSE Bit (SSn# Pin Function Enable)

Set this bit to 1 if the SSn# pin is to be used in control of transmission and reception (in simple SPI mode). Set this bit to 0 in any other mode. Furthermore, even for usage in simple SPI mode, the SSn# pin on the master side is not required to control reception and transmission when master mode (SCR.CKE[1:0] = 00b and MSS = 0) is selected and there is a single master, so the setting for the SSE bit is 0. Do not set both the SSE and CTSE bits to enabled (even if this setting is made, operation is the same as that when these bits are set to 0).

CTSE Bit (CTS Enable)

Set this bit to 1 if the SSn# pin is to be used for inputting of the CTS control signal to control of transmission and reception. The RTS signal is output when this bit is set to 0. Set this bit to 0 in smart card interface mode, simple SPI mode, and simple I^2C mode. Do not set both the CTSE and SSE bits to enabled (even if this setting is made, operation is the same as that when these bits are set to 0).

MSS Bit (Master Slave Select)

This bit selects between master and slave operation in simple SPI mode. When the MSS bit is set to 1, data is received through the SMOSIn pin and transmitted through the SMISOn pin.
Set this bit to 0 in modes other than simple SPI mode.

MFF Flag (Mode Fault Flag)

This bit indicates mode fault errors.
In a multi-master configuration, determine the mode fault error occurrence by reading the MFF flag.
[Setting condition]

- Input on the SSn# pin being at the low level during master operation in simple SPI mode (SSE bit = 1 and MSS bit = 0)

[Clearing condition]

- Writing 0 to the bit after it was read as 1

CKPOL Bit (Clock Polarity Select)

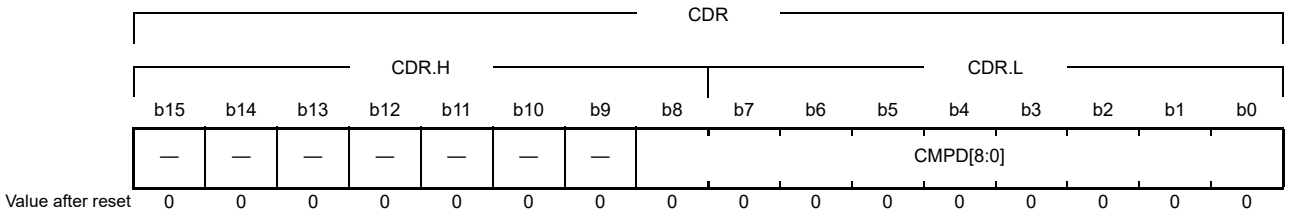
This bit selects the polarity of the clock signal output through the SCKn pin. Refer to Figure 32.62 for details.
Set the bit to 0 in other than simple SPI mode and clock synchronous mode.

CKPH Bit (Clock Phase Select)

This bit selects the phase of the clock signal output through the SCKn pin. Refer to Figure 32.62 for details.
Set the bit to 0 in other than simple SPI mode and clock synchronous mode.

32.2.20 Comparison Data Register (CDR)

Address(es): SCI1.CDR 0008 A03Ah, SCI5.CDR 0008 A0BAh, SCI6.CDR 0008 A0DAh,
SCI1.CDR.H 0008 A03Ah, SCI5.CDR.H 0008 A0BAh, SCI6.CDR.H 0008 A0DAh,
SCI1.CDR.L 0008 A03Bh, SCI5.CDR.L 0008 A0BBh, SCI6.CDR.L 0008 A0DBh



Bit	Symbol	Bit Name	Description	R/W
b8 to b0	CMPD[8:0]	Comparison Data	These bits specify the data of comparison source when using the data match detection function.	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R

CMPD[8:0] Bits (Comparison Data)

These bits are used for detecting a data match. The length of the valid bit is the same as the character length set in the SMR.CHR and SCMR.CHR1 bits.
The DCCR.DCMF flag becomes 1 when the received data matches with the value of these bits.

32.2.21 Data Comparison Control Register (DCCR)

Address(es): SCI1.DCCR 0008 A033h, SCI5.DCCR 0008 A0B3h, SCI6.DCCR 0008 A0D3h

	b7	b6	b5	b4	b3	b2	b1	b0
	DCME	IDSEL	—	DFER	DPER	—	—	DCMF
Value after reset	0	1	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DCMF	Data Match Flag	0: Data is not matched 1: Data is matched	R/(W) *1
b2, b1	—	Reserved	These bits are read as 0. The write value should be 0	R/W
b3	DPER	Match Data Parity Error Flag	0: A parity error is not found in the matched data. 1: A parity error is found in the matched data.	R/(W) *1
b4	DFER	Match Data Framing Error Flag	0: A framing error is not found in the matched data. 1: A framing error is found in the matched data.	R/(W) *1
b5	—	Reserved	This bit is read as 0. The write value should be 0	R/W
b6	IDSEL	ID Frame Select*2	0: All data is to be compared 1: The data with the multi-processor bit set to 1 is to be compared	R/W
b7	DCME	Data Match Detection Enable*2	0: Data match detection is disabled 1: Data match detection is enabled	R/W

Note 1. Only 0 can be written to this bit, which clears the flag. To clear this flag, confirm that the flag is 1, and then write 0 to the flag.
Note 2. This bit is only valid in asynchronous mode.

DCMF Flag (Data Match Flag)

This flag indicates the comparison result of the received data and the value of the CDR register.
[Setting condition]

- When the received data matches with the value in the CDR register while the DCME bit is 1

[Clearing condition]

- When 0 is written to the flag after reading this flag as 1

Even when the SCR.RE bit is set to 0, the DCMF flag has no effect and retains the previous state.

DPER Flag (Match Data Parity Error Flag)

This flag indicates if a parity error has occurred in the matched data.
[Setting condition]

- When a parity error is found in the received data in which a data match is detected.

[Clearing condition]

- When 0 is written to the flag after reading the flag as 1.

Refer to section 14.5.2, Level Detection to exit from the interrupt handling routine after setting the DPER flag to 0.
Even when the SCR.RE bit is set to 0, the DPER flag has no effect and retains the previous state.

DFER Flag (Match Data Framing Error Flag)

This flag indicates if a framing error has occurred in the matched data.
[Setting condition]

- When the stop bit of the received frame in which a data match is detected is 0

[Clearing condition]

- When 0 is written to the flag after reading the flag as 1.

Refer to section 14.5.2, Level Detection to exit from the interrupt handling routine after setting the DFER flag to 0.
Even when the SCR.RE bit is set to 0, the DFER flag has no effect and retains the previous state.

IDSEL Bit (ID Frame Select)

This bit specifies the condition of the received data to be compared. The bit is valid only when the DCME bit is 1.
When setting this bit to 1, only data in received frames (ID frames) in which the multi-processor bit has the value 1 are compared.
When this bit is set to 0, all received data is compared.

DCME Bit (Data Match Detection Enable)

This bit enables or disables the data match detection function. The function is valid only in asynchronous mode. In other modes, set this bit to 0.
This bit automatically becomes 0 when a data match is detected.

32.2.22 Serial Port Register (SPTR)

Address(es): SCI1.SPTR 0008 A03Ch, SCI5.SPTR 0008 A0BCh, SCI6.SPTR 0008 A0DCh

	b7	b6	b5	b4	b3	b2	b1	b0
	TTADJ	RTADJ	TINV	RINV	—	SPB2IO	SPB2DT	RXDMON
Value after reset	0	0	0	0	0	0	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	RXDMON	RXD Line Monitoring Flag	When the RINV bit is 0 0: The RXDn pin level is low 1: The RXDn pin level is high When the RINV bit is 1 0: The RXDn pin level is high 1: The RXDn pin level is low	R
b1	SPB2DT	Serial Port Break Data*1	Combine bits SPB2DT, SPB2IO, TINV, and SCR.TE to control the TXDn pin. Refer to Table 32.29 for details.	R/W
b2	SPB2IO	Serial Port Break I/O*1		R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R
b4	RINV	Receiver Input Invert*2	0: Does not invert the input signal from the RXD pin 1: Inverts the input signal from the RXD pin	R/W *3
b5	TINV	Transmitter Output Invert*2	0: Does not invert the output signal to the TXD pin 1: Invert the output signal to the TXD pin	R/W *3
b6	RTADJ	Receive Data Sampling Timing Adjustment*4	0: Does not adjust the sampling point of receive data 1: Adjust the sampling point of receive data	R/W *3
b7	TTADJ	Transmit Signal Transition Timing Adjustment*4	0: Does not adjust the transition timing of transmit data 1: Adjust the transition timing of transmit data	R/W *3

Note 1. This bit is only valid in asynchronous mode.
Note 2. Set this bit to 0 if operation is to be in smart card interface mode or simple I^2C mode.
Note 3. This bit is rewritable only when the TE and RE bits in the SCR register are both 0.
Note 4. This bit is only valid when the on-chip baud rate generator is selected as the clock source in asynchronous mode.

RXDMON Flag (RXD Line Monitoring Flag)

This flag is used for monitoring the level of the RXDn pin.

SPB2DT Bit (Serial Port Break Data)

This bit specifies the output level of the TXDn pin when the SCR.TE bit is 0. Refer to Table 32.29 for details.

SPB2IO Bit (Serial Port Break I/O)

This bit specifies input or output of the TXDn pin when the SCR.TE bit is 0. Set this bit to 1 (output) when controlling the TXDn pin by software.

Table 32.29 Controlling the TXDn pin

SCR.TE Bit Setting	SPB2IO Bit Setting	SPB2DT Bit Setting	TINV Bit Setting	TXDn Pin Status
0 (Transmission disabled)	0 (Input)	0 or 1	0 or 1	Hi-Z
	1 (Output)	0	0	Low is output
			1	High is output
		1	0	High is output
			1	Low is output
1 (Transmission enabled)	0 or 1	0 or 1	0 or 1	Transmit data output pin

RINV Bit (Receiver Input Invert)

This bit is used to logically invert the input signal from the RXDn pin in front of the receive shift register. Besides data bits, start bit, parity bit, and stop bit are inverted.

TINV Bit (Transmitter Output Invert)

This bit is used to logically invert the output signal from the transmit shift register in front of the TXDn pin. Besides data bits, start bit, parity bit, and stop bit are inverted.

RTADJ Bit (Receive Data Sampling Timing Adjustment)

This bit is used to change the sampling point of the receive data from the default sampling point. This bit is used to improve the receive margin when the high and low widths of the receive signal are changed due to the characteristics of the transmission line or interface devices. Normally, set this bit to 0.

TTADJ Bit (Transmit Signal Transition Timing Adjustment)

This bit is used to change the transition point of the transmit data from the default transition point. This bit is used to improve the receive margin of the receiver when the high and low widths of the transmitted signal are intended to be changed due to the characteristics of the transmission line or interface devices. Normally, set this bit to 0.

32.2.23 Transmit/Receive Timing Select Register (TMGR)

Address(es): SCI1.TMGR 0008 A03Dh, SCI5.TMGR 0008 A0BDh, SCI6.TMGR 0008 A0DDh

	b7	b6	b5	b4	b3	b2	b1	b0
	TTMG[3:0]				RTMG[3:0]			
Value after reset	0	0	0	0	0	0	1	1

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	RTMG[3:0]	Receive Data Sampling Timing Select*1	b3 b0 1 1 1 1: Data are sampled 7 clocks earlier than default point. 1 1 1 0: Data are sampled 6 clocks earlier than default point. 1 1 0 1: Data are sampled 5 clocks earlier than default point. 1 1 0 0: Data are sampled 4 clocks earlier than default point. 1 0 1 1: Data are sampled 3 clocks earlier than default point. 1 0 1 0: Data are sampled 2 clocks earlier than default point. 1 0 0 1: Data are sampled 1 clock earlier than default point. x 0 0 0: Data are sampled at default point. 0 0 0 1: Data are sampled 1 clock later than default point. 0 0 1 0: Data are sampled 2 clocks later than default point. 0 0 1 1: Data are sampled 3 clocks later than default point. 0 1 0 0: Data are sampled 4 clocks later than default point. 0 1 0 1: Data are sampled 5 clocks later than default point. 0 1 1 0: Data are sampled 6 clocks later than default point. 0 1 1 1: Data are sampled 7 clocks later than default point.	R/W *2
b7 to b4	TTMG[3:0]	Transmit Signal Transition Timing Select*3	b7 b4 1 1 1 1: Delays the 1 to 0 transitions for 7 clocks. 1 1 1 0: Delays the 1 to 0 transitions for 6 clocks. 1 1 0 1: Delays the 1 to 0 transitions for 5 clocks. 1 1 0 0: Delays the 1 to 0 transitions for 4 clocks. 1 0 1 1: Delays the 1 to 0 transitions for 3 clocks. 1 0 1 0: Delays the 1 to 0 transitions for 2 clocks. 1 0 0 1: Delays the 1 to 0 transitions for 1 clock. x 0 0 0: Does not change the waveform. 0 0 0 1: Delays the 0 to 1 transitions for 1 clock. 0 0 1 0: Delays the 0 to 1 transitions for 2 clocks. 0 0 1 1: Delays the 0 to 1 transitions for 3 clocks. 0 1 0 0: Delays the 0 to 1 transitions for 4 clocks. 0 1 0 1: Delays the 0 to 1 transitions for 5 clocks. 0 1 1 0: Delays the 0 to 1 transitions for 6 clocks. 0 1 1 1: Delays the 0 to 1 transitions for 7 clocks.	R/W *4

Note 1. These bits are only valid when the SPTR.RTADJ bit is 1.
Note 2. These bits are rewritable only when the SPTR.RTADJ bit is 0.
Note 3. These bits are only valid when the SPTR.TTADJ bit is 1.
Note 4. These bits are rewritable only when the SPTR.TTADJ bit is 0.

The TMGR register is used to adjust the sampling point of the receive data and the transition timing of the transmit data. This register is only valid when the on-chip baud rate generator is selected as the clock source in asynchronous mode. This register is not present in SCI12.

RTMG[3:0] Bits (Receive Data Sampling Timing Select)

These bit are used to select the sampling points of the receive data. These bit are only valid when the SPTR.RTADJ bit is 1. When the RTMG[3] bit is 0, each bit is sampled later than the default sampling point. When 1, each bit is sampled earlier than the default sampling point.

Set the amount of movement of the sampling points to the RTMG[2:0] bits by the number of the base clocks. Refer to Table 32.30 for the range of the value that can be set in the RTMG[2:0] bits.

Table 32.30 Range of Setting Values for the RTMG[2:0] Bits

SEMR.ABCSE Bit Setting	SEMR.ABCS Bit Setting	Number of the Base Clock Cycles for 1 Data Bit	Setting Range
0	0	16 cycles	0 to 7 (000b to 111b)
0	1	8 cycles	0 to 3 (000b to 011b)
1	0 or 1	6 cycles	0 to 2 (000b to 010b)

TTMG[3:0] Bits (Transmit Signal Transition Timing Select)

These bits are used to select the transition timing of the transmit signal in the transmit shift register. These bits are only valid when the SPTR.TTADJ bit is 1.

When the TTMG[3] bit is 0, the 0 to 1 transitions are delayed. When the TTMG[3] bit is 1, the 1 to 0 transitions are delayed. Output signal from the TXDn pin depends on the SPTR.TINV bit as follows.

(1) When the SPTR.TINV bit is 0

When the TTMG[3] bit is 0, high pulse width is narrower than low pulse width because low to high transitions (rising edges) are delayed.

When the TTMG[3] bit is 1, high pulse width is wider than low pulse width because high to low transitions (falling edges) are delayed.

(2) When the TINV bit is 1

When the TTMG[3] bit is 0, high pulse width is wider than low pulse width because high to low transitions (falling edges) are delayed.

When the TTMG[3] bit is 1, high pulse width is narrower than low pulse width because low to high transitions (rising edges) are delayed.

Set the delay amount to the TTMG[2:0] bits in number of the base clock. Refer to Table 32.31 for the range of the value that can be set in the TTMG[2:0] bits.

Table 32.31 Range of Setting Values for the TTMG[2:0] Bits

SEMR.ABCSE Bit Setting	SEMR.ABCS Bit Setting	Number of the Base Clock Cycles for 1 Data Bit	Setting Range
0	0	16 cycles	0 to 7 (000b to 111b)
0	1	8 cycles	0 to 7 (000b to 111b)
1	0 or 1	6 cycles	0 to 5 (000b to 101b)

32.2.24 Extended Serial Module Enable Register (ESMER)

Address(es): SCI12.ESMER 0008 B320h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	ESME
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ESME	Extended Serial Mode Enable	0: The extended serial mode is disabled. 1: The extended serial mode is enabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ESME Bit (Extended Serial Mode Enable)

When the ESME bit is 1, the facilities of the extended serial mode control section are enabled.

When the ESME bit is 0, the extended serial mode control section is initialized.

Table 32.32 Settings of the ESME Bit and Timer Operation Mode

ESME Bit	Timer Mode	Break Field Low Width Determination Mode	Break Field Low Width Output Mode
0	Available*1	Not available	Not available
1	Available	Available	Available

Note 1. Operation is only possible with PCLK selected.

32.2.25 Control Register 0 (CR0)

Address(es): SCI12.CR0 0008 B321h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	BRME	RXDSF	SFSF	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b1	SFSF	Start Frame Status Flag	0: Start Frame detection function is disabled. 1: Start Frame detection function is enabled.	R
b2	RXDSF	RXDX12 Input Status Flag	0: RXDX12 input is enabled. 1: RXDX12 input is disabled.	R
b3	BRME	Bit Rate Measurement Enable	0: Measurement of bit rate is disabled. 1: Measurement of bit rate is enabled.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

32.2.26 Control Register 1 (CR1)

Address(es): SCI12.CR1 0008 B322h

	b7	b6	b5	b4	b3	b2	b1	b0
	PIBS[2:0]			PIBE	CF1DS[1:0]		CF0RE	BFE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BFE	Break Field Enable	0: Break Field detection is disabled. 1: Break Field detection is enabled.	R/W
b1	CF0RE	Control Field 0 Reception Enable	0: Reception of Control Field 0 is disabled. 1: Reception of Control Field 0 is enabled.	R/W
b3, b2	CF1DS[1:0]	Control Field 1 Data Register Select	b3 b2 0 0: Selects comparison with the value in the PCF1DR register. 0 1: Selects comparison with the value in the SCF1DR register. 1 0: Selects comparison with the values in the PCF1DR and SCF1DR registers. 1 1: Setting prohibited.	R/W
b4	PIBE	Priority Interrupt Bit Enable	0: The priority interrupt bit is disabled. 1: The priority interrupt bit is enabled.	R/W
b7 to b5	PIBS[2:0]	Priority Interrupt Bit Select	b7 b5 0 0 0: 0th bit of Control Field 1 0 0 1: 1st bit of Control Field 1 0 1 0: 2nd bit of Control Field 1 0 1 1: 3rd bit of Control Field 1 1 0 0: 4th bit of Control Field 1 1 0 1: 5th bit of Control Field 1 1 1 0: 6th bit of Control Field 1 1 1 1: 7th bit of Control Field 1	R/W

32.2.27 Control Register 2 (CR2)

Address(es): SCI12.CR2 0008 B323h

	b7	b6	b5	b4	b3	b2	b1	b0
	RTS[1:0]		BCCS[1:0]		—	DFCS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	DFCS[2:0]	RXDX12 Signal Digital Filter Clock Select	b2 b0 0 0 0: Filter is disabled. 0 0 1: Filter clock is base clock*1, *2 0 1 0: Filter clock is PCLK/8 0 1 1: Filter clock is PCLK/16 1 0 0: Filter clock is PCLK/32 1 0 1: Filter clock is PCLK/64 1 1 0: Filter clock is PCLK/128 1 1 1: Setting prohibited	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5, b4	BCCS[1:0]	Bus Collision Detection Clock Select	• When SEMR.BGDM = 0 or SEMR.BGDM = 1 and SMR.CKS[1:0] = a value other than 00b b5 b4 0 0: Base clock 0 1: Base clock frequency divided by 2 1 0: Base clock frequency divided by 4 1 1: Setting prohibited • When SEMR.BGDM = 1 and SMR.CKS[1:0] = 00b b5 b4 0 0: Base clock frequency divided by 2 0 1: Base clock frequency divided by 4 1 0: Setting prohibited 1 1: Setting prohibited	R/W
b7, b6	RTS[1:0]	RXDX12 Reception Sampling Timing Select	• When SCI12.SEMR.ABCS = 0 b7 b6 0 0: Rising edge of the 8th cycle of base clock 0 1: Rising edge of the 10th cycle of base clock 1 0: Rising edge of the 12th cycle of base clock 1 1: Rising edge of the 14th cycle of base clock • When SCI12.SEMR.ABCS = 1 b7 b6 0 0: Rising edge of the 4th cycle of base clock 0 1: Rising edge of the 5th cycle of base clock 1 0: Rising edge of the 6th cycle of base clock 1 1: Rising edge of the 7th cycle of base clock	R/W

Note: The period of the base clock is 1/16 of a single bit period when the SCI12.SEMR.ABCS is 0, and 1/8 of a single bit period when the SCI12.SEMR.ABCS is 1.
Note 1. To use the base clock, set the SCI12.SCR.TE bit to 1.
Note 2. The base clock divided by 2 is the filter clock when the SEMR.BGDM bit is 1 and the SMR.CKS[1:0] bits are 00b.

32.2.28 Control Register 3 (CR3)

Address(es): SCI12.CR3 0008 B324h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SDST
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SDST	Start Frame Detection Start	0: Detection of Start Frame is not performed. 1: Detection of Start Frame is performed.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SDST Bit (Start Frame Detection Start)

Detection of a Start Frame begins when this bit is set to 1. The bit is read as 0.

32.2.29 Port Control Register (PCR)

Address(es): SCI12.PCR 0008 B325h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	SHARPS	—	—	RXDXPS	TXDXPS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TXDXPS	TXDX12 Signal Polarity Select	0: The polarity of TXDX12 signal is not inverted for output. 1: The polarity of TXDX12 signal is inverted for output.	R/W
b1	RXDXPS	RXDX12 Signal Polarity Select	0: The polarity of RXDX12 signal is not inverted for input. 1: The polarity of RXDX12 signal is inverted for input.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	SHARPS	TXDX12/RXDX12 Pin Multiplexing Select	0: The TXDX12 and RXDX12 pins are independent. 1: The TXDX12 and RXDX12 signals are multiplexed on the same pin.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SHARPS Bit (TXDX12/RXDX12 Pin Multiplexing Select)

When this bit is set to 1, the TXDX12 and RXDX12 signals are multiplexed on the same pin so that half-duplex communications become possible.

32.2.30 Interrupt Control Register (ICR)

Address(es): SCI12.ICR 0008 B326h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	AEDIE	BCDIE	PIBDIE	CF1MIE	CF0MIE	BFDIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BFDIE	Break Field Low Width Detected Interrupt Enable	0: Interrupts on detection of the low width for a Break Field are disabled. 1: Interrupts on detection of the low width for a Break Field are enabled.	R/W
b1	CF0MIE	Control Field 0 Match Detected Interrupt Enable	0: Interrupts on detection of a match with Control Field 0 are disabled. 1: Interrupts on detection of a match with Control Field 0 are enabled.	R/W
b2	CF1MIE	Control Field 1 Match Detected Interrupt Enable	0: Interrupts on detection of a match with Control Field 1 are disabled. 1: Interrupts on detection of a match with Control Field 1 are enabled.	R/W
b3	PIBDIE	Priority Interrupt Bit Detected Interrupt Enable	0: Interrupts on detection of the priority interrupt bit are disabled. 1: Interrupts on detection of the priority interrupt bit are enabled.	R/W
b4	BCDIE	Bus Collision Detected Interrupt Enable	0: Interrupts on detection of a bus collision are disabled. 1: Interrupts on detection of a bus collision are enabled.	R/W
b5	AEDIE	Valid Edge Detected Interrupt Enable	0: Interrupts on detection of a valid edge are disabled. 1: Interrupts on detection of a valid edge are enabled.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

32.2.31 Status Register (STR)

Address(es): SCI12.STR 0008 B327h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	AEDF	BCDF	PIBDF	CF1MF	CF0MF	BFDF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BFDF	Break Field Low Width Detection Flag	[Setting conditions] • Detection of the low width for a Break Field • Completion of the output of the low width for a Break Field • Underflow of the timer [Clearing condition] • Writing 1 to the STCR.BFDCL bit	R
b1	CF0MF	Control Field 0 Match Flag	[Setting condition] • A match between the value received in Control Field 0 and the set value. [Clearing condition] • Writing 1 to the STCR.CF0MCL bit	R
b2	CF1MF	Control Field 1 Match Flag	[Setting condition] • A match between the data received in Control Field 1 and the set values. [Clearing condition] • Writing 1 to the STCR.CF1MCL bit	R
b3	PIBDF	Priority Interrupt Bit Detection Flag	[Setting condition] • Detection of the priority interrupt bit [Clearing condition] • Writing 1 to the STCR.PIBDCL bit	R
b4	BCDF	Bus Collision Detected Flag	[Setting condition] • Detection of the bus collision [Clearing condition] • Writing 1 to the STCR.BCDCL bit	R
b5	AEDF	Valid Edge Detection Flag	[Setting condition] • Detection of a valid edge [Clearing condition] • Writing 1 to the STCR.AEDCL bit	R
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R

32.2.32 Status Clear Register (STCR)

Address(es): SCI12.STCR 0008 B328h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	AEDCL	BCDCL	PIBDCL	CF1MCL	CF0MCL	BFDCL
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BFDCL	BFDF Clear	Setting this bit to 1 clears the STR.BFDF flag. This bit is read as 0.	R/W
b1	CF0MCL	CF0MF Clear	Setting this bit to 1 clears the STR.CF0MF flag. This bit is read as 0.	R/W
b2	CF1MCL	CF1MF Clear	Setting this bit to 1 clears the STR.CF1MF flag. This bit is read as 0.	R/W
b3	PIBDCL	PIBDF Clear	Setting this bit to 1 clears the STR.PIBDF flag. This bit is read as 0.	R/W
b4	BCDCL	BCDF Clear	Setting this bit to 1 clears the STR.BCDF flag. This bit is read as 0.	R/W
b5	AEDCL	AEDF Clear	Setting this bit to 1 clears the STR.AEDF flag. This bit is read as 0.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

32.2.33 Control Field 0 Data Register (CF0DR)

Address(es): SCI12.CF0DR 0008 B329h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

The CF0DR register is an 8-bit readable and writable register that holds a value for comparison with Control Field 0.

32.2.34 Control Field 0 Compare Enable Register (CF0CR)

Address(es): SCI12.CF0CR 0008 B32Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	CF0CE7	CF0CE6	CF0CE5	CF0CE4	CF0CE3	CF0CE2	CF0CE1	CF0CE0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CF0CE0	Control Field 0 Bit 0 Compare Enable	0: Comparison with bit 0 of Control Field 0 is disabled. 1: Comparison with bit 0 of Control Field 0 is enabled.	R/W
b1	CF0CE1	Control Field 0 Bit 1 Compare Enable	0: Comparison with bit 1 of Control Field 0 is disabled. 1: Comparison with bit 1 of Control Field 0 is enabled.	R/W
b2	CF0CE2	Control Field 0 Bit 2 Compare Enable	0: Comparison with bit 2 of Control Field 0 is disabled. 1: Comparison with bit 2 of Control Field 0 is enabled.	R/W
b3	CF0CE3	Control Field 0 Bit 3 Compare Enable	0: Comparison with bit 3 of Control Field 0 is disabled. 1: Comparison with bit 3 of Control Field 0 is enabled.	R/W
b4	CF0CE4	Control Field 0 Bit 4 Compare Enable	0: Comparison with bit 4 of Control Field 0 is disabled. 1: Comparison with bit 4 of Control Field 0 is enabled.	R/W
b5	CF0CE5	Control Field 0 Bit 5 Compare Enable	0: Comparison with bit 5 of Control Field 0 is disabled. 1: Comparison with bit 5 of Control Field 0 is enabled.	R/W
b6	CF0CE6	Control Field 0 Bit 6 Compare Enable	0: Comparison with bit 6 of Control Field 0 is disabled. 1: Comparison with bit 6 of Control Field 0 is enabled.	R/W
b7	CF0CE7	Control Field 0 Bit 7 Compare Enable	0: Comparison with bit 7 of Control Field 0 is disabled. 1: Comparison with bit 7 of Control Field 0 is enabled.	R/W

32.2.35 Control Field 0 Receive Data Register (CF0RR)

Address(es): SCI12.CF0RR 0008 B32Bh

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

CF0RR is a readable register that holds the value received in Control Field 0.

32.2.36 Primary Control Field 1 Data Register (PCF1DR)

Address(es): SCI12.PCF1DR 0008 B32Ch

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

PCF1DR is an 8-bit readable and writable register that holds the 8-bit primary value for comparison with Control Field 1.

32.2.37 Secondary Control Field 1 Data Register (SCF1DR)

Address(es): SCI12.SCF1DR 0008 B32Dh

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

PCF1DR is an 8-bit readable and writable register that holds the 8-bit secondary value for comparison with Control Field 1.

32.2.38 Control Field 1 Compare Enable Register (CF1CR)

Address(es): SCI12.CF1CR 0008 B32Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	CF1CE7	CF1CE6	CF1CE5	CF1CE4	CF1CE3	CF1CE2	CF1CE1	CF1CE0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CF1CE0	Control Field 1 Bit 0 Compare Enable	0: Comparison with bit 0 of Control Field 1 is disabled. 1: Comparison with bit 0 of Control Field 1 is enabled.	R/W
b1	CF1CE1	Control Field 1 Bit 1 Compare Enable	0: Comparison with bit 1 of Control Field 1 is disabled. 1: Comparison with bit 1 of Control Field 1 is enabled.	R/W
b2	CF1CE2	Control Field 1 Bit 2 Compare Enable	0: Comparison with bit 2 of Control Field 1 is disabled. 1: Comparison with bit 2 of Control Field 1 is enabled.	R/W
b3	CF1CE3	Control Field 1 Bit 3 Compare Enable	0: Comparison with bit 3 of Control Field 1 is disabled. 1: Comparison with bit 3 of Control Field 1 is enabled.	R/W
b4	CF1CE4	Control Field 1 Bit 4 Compare Enable	0: Comparison with bit 4 of Control Field 1 is disabled. 1: Comparison with bit 4 of Control Field 1 is enabled.	R/W
b5	CF1CE5	Control Field 1 Bit 5 Compare Enable	0: Comparison with bit 5 of Control Field 1 is disabled. 1: Comparison with bit 5 of Control Field 1 is enabled.	R/W
b6	CF1CE6	Control Field 1 Bit 6 Compare Enable	0: Comparison with bit 6 of Control Field 1 is disabled. 1: Comparison with bit 6 of Control Field 1 is enabled.	R/W
b7	CF1CE7	Control Field 1 Bit 7 Compare Enable	0: Comparison with bit 7 of Control Field 1 is disabled. 1: Comparison with bit 7 of Control Field 1 is enabled.	R/W

32.2.39 Control Field 1 Receive Data Register (CF1RR)

Address(es): SCI12.CF1RR 0008 B32Fh

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

CF1RR is a readable register that holds the value received in Control Field 1.

32.2.40 Timer Control Register (TCR)

Address(es): SCI12.TCR 0008 B330h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	TCST
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TCST	Timer Count Start	0: Stops the timer counting 1: Starts the timer counting	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

32.2.41 Timer Mode Register (TMR)

Address(es): SCI12.TMR 0008 B331h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	TCSS[2:0]			TWRC	—	TOMS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TOMS[1:0]	Timer Operating Mode Select*1	b1 b0 0 0: Timer mode 0 1: Break Field low width determination mode 1 0: Break Field low width output mode 1 1: Setting prohibited	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	TWRC	Counter Write Control	0: Data is written to the reload register and counter 1: Data is written to the reload register only	R/W
b6 to b4	TCSS[2:0]	Timer Count Clock Source Select*1	b6 b4 0 0 0: PCLK 0 0 1: PCLK/2 0 1 0: PCLK/4 0 1 1: PCLK/8 1 0 0: PCLK/16 1 0 1: PCLK/32 1 1 0: PCLK/64 1 1 1: PCLK/128	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. Rewrite the TOMS[1:0] and TCSS[2:0] bits only when the timer is stopped (TCST = 0).

TWRC Bit (Counter Write Control)

This bit determines whether a value written to the TPRE or TCNT register is written to the reload register only or is written to both the reload register and the counter.

32.2.42 Timer Prescaler Register (TPRE)

Address(es): SCI12.TPRE 0008 B332h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1

TPRE consists of an 8-bit reload register, a read buffer, and a counter, each of which has FFh as its initial value. The counter counts down in synchronization with the counter clock selected by the TMR.TCSS[2:0] bits, and is reloaded with the value in the reload register when it underflows. Underflows of this register provide the clock source to drive counting by the TCNT register. The reload register and read buffer are allocated to the same address. Data is written to the reload register in writing, and the counter value that has been transferred to the read buffer is returned in reading.
It takes one PCLK cycle to load a value from the reload register to the counter.

32.2.43 Timer Count Register (TCNT)

Address(es): SCI12.TCNT 0008 B333h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1

TCNT consists of an 8-bit reload register, a read buffer, and a counter, each of which has FFh as its initial value. This down-counter counts underflows of the TPRE register until the TCNT register underflows, and is then reloaded with the value from the reload register. The reload register and read buffer are allocated to the same address. Data is written to the reload register in writing, and the counter value that has been transferred to the read buffer is returned in reading.
It takes one PCLK cycle to load a value from the reload register to the counter.

32.2.44 Device Function Select Register 0 (PRDFR0)

Address(es): SYSTEM.PRDFR0 0008 00D0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	SCI11RXD[1:0]		—	—	SCI9RXD[1:0]		SCI8RXD[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	SCI5RXD[1:0]		—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11, b10	SCI5RXD[1:0]	SCI5 RXD Input Signal Select	These bits select the signal to be input to the RXD pin of SCI5. b11 b10 0 0: Input signal from the RXD5 pin 0 1: COMP3 level detection signal 1 0: COMP4 level detection signal Settings other than above are prohibited.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b17, b16	SCI8RXD[1:0]	RSCI8 RXD Input Signal Select	These bits select the signal to be input to the RXD pin of RSCI8. b17 b16 0 0: Input signal from the RXD008 pin 0 1: COMP3 level detection signal 1 0: COMP4 level detection signal Settings other than above are prohibited.	R/W
b19, b18	SCI9RXD[1:0]	RSCI9 RXD Input Signal Select	These bits select the signal to be input to the RXD pin of RSCI9. b19 b18 0 0: Input signal from the RXD009 pin 0 1: COMP3 level detection signal 1 0: COMP4 level detection signal Settings other than above are prohibited.	R/W
b21, b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23, b22	SCI11RXD[1:0]	RSCI11 RXD Input Signal Select	These bits select the signal to be input to the RXD pin of RSCI11. b23 b22 0 0: Input signal from the RXD011 pin 0 1: COMP3 level detection signal 1 0: COMP4 level detection signal Settings other than above are prohibited.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

If the RXD signal is attenuated by the transmission line effects and does not meet the VIH/VIL specifications, it can be improved by intervening a comparator.
When using a comparator, connect the RXD signal to one of the CMPC30 to CMPC33 or CMPC40 to CMPC43 pins. Also, disable the noise filter in the comparator. In this case, the RXD5 pin is not used.

32.3 Operation in Asynchronous Mode

Figure 32.5 shows the general format for asynchronous serial communications.
One frame consists of a start bit (low level), transmit/receive data, a parity bit, and stop bits (high level).
In asynchronous serial communications, the communications line is usually held in the mark state (high level).
The SCI monitors the communications line. When the SCI detects a low, it regards that as a start bit and starts serial communication.
Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communications. Both the transmitter and the receiver also have a double-buffered structure, so that data can be read or written during transmission or reception, enabling continuous data transmission and reception.

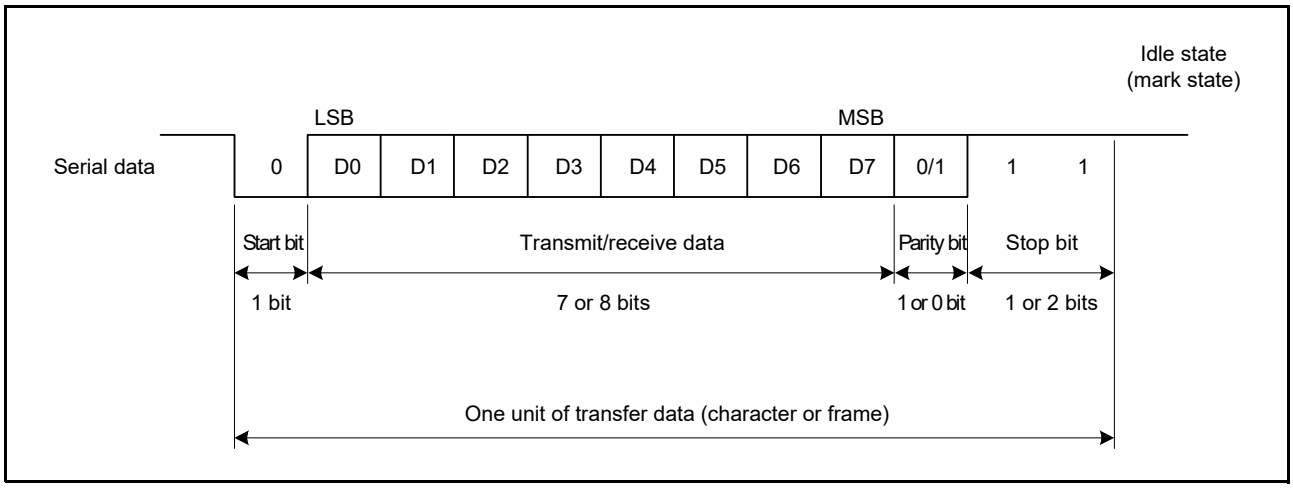


Figure 32.5 Data Format in Asynchronous Serial Communications (Example with 8-Bit Data, Parity, 2 Stop Bits)

32.3.1 Serial Data Transfer Format

Table 32.33 lists the serial data transfer formats that can be used in asynchronous mode.
Any of 18 transfer formats can be selected according to the SMR and SCMR setting. For details of multi-processor function, refer to section 32.4, Multi-Processor Communications Function.

Table 32.33 Serial Transfer Formats (Asynchronous Mode)

SCMR Setting	SMR Setting				Serial Transfer Format and Frame Length
CHR1	CHR	PE	MP	STOP	1 2 3 4 5 6 7 8 9 10 11 12 13
0	0	0	0	0	S 9-bit data STOP
0	0	0	0	1	S 9-bit data STOP STOP
0	0	1	0	0	S 9-bit data P STOP
0	0	1	0	1	S 9-bit data P STOP STOP
1	0	0	0	0	S 8-bit data STOP
1	0	0	0	1	S 8-bit data STOP STOP
1	0	1	0	0	S 8-bit data P STOP
1	0	1	0	1	S 8-bit data P STOP STOP
1	1	0	0	0	S 7-bit data STOP
1	1	0	0	1	S 7-bit data STOP STOP
1	1	1	0	0	S 7-bit data P STOP
1	1	1	0	1	S 7-bit data P STOP STOP
0	0	—	1	0	S 9-bit data MPB STOP
0	0	—	1	1	S 9-bit data MPB STOP STOP
1	0	—	1	0	S 8-bit data MPB STOP
1	0	—	1	1	S 8-bit data MPB STOP STOP
1	1	—	1	0	S 7-bit data MPB STOP
1	1	—	1	1	S 7-bit data MPB STOP STOP

S: Start bit
STOP: Stop bit
P: Parity bit
MPB: Multi-processor bit

32.3.2 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode

In asynchronous mode, the SCI operates on a base clock with a frequency of 16 times*1 the bit rate.
In reception, the SCI samples the falling edge of the start bit using the base clock, and performs internal synchronization. Since receive data is sampled at the rising edge of the 8th pulse*1 of the base clock, data is latched at the middle*2 of each bit, as shown in Figure 32.6. Thus the reception margin in asynchronous mode is determined by formula (1) below.

$$M = \left| \left(0.5 - \frac{1}{2N}\right) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\ (\%) \quad \cdots \text{Formula (1)}$$

M: Reception margin
N: Ratio of bit rate to clock
(N = 16 when SEMR.ABCSE = 0 and SEMR.ABCS = 0, N = 8 when SEMR.ABCSE = 0 and SEMR.ABCS = 1, N = 6 when SEMR.ABCSE = 1)
D: Duty cycle of clock (D = 0.5 to 1.0)
L: Frame length (L = 9 to 13)
F: Absolute value of clock frequency deviation

Assuming values of F = 0 and D = 0.5 in formula (1), the reception margin is determined by the formula below.

M = {0.5 – 1/(2 × 16)} × 100 (%) = 46.875 (%)

However, this is only the computed value, and a margin of 20% to 30% should be allowed in system design.

Note 1. These are values when the ABCSE and ABCS bits in the SEMR register are 0. When the ABCSE bit is 0 and the ABCS bit is 1, a frequency of 8 times the bit rate is used as a base clock and receive data is sampled at the rising edge of the 4th pulse of the base clock. When the ABCSE bit is 1, a frequency of 6 times the bit rate is used as a base clock and receive data is sampled at the rising edge of the 3rd pulse of the base clock.
Note 2. This is when the SPTR.RTADJ bit is 0.

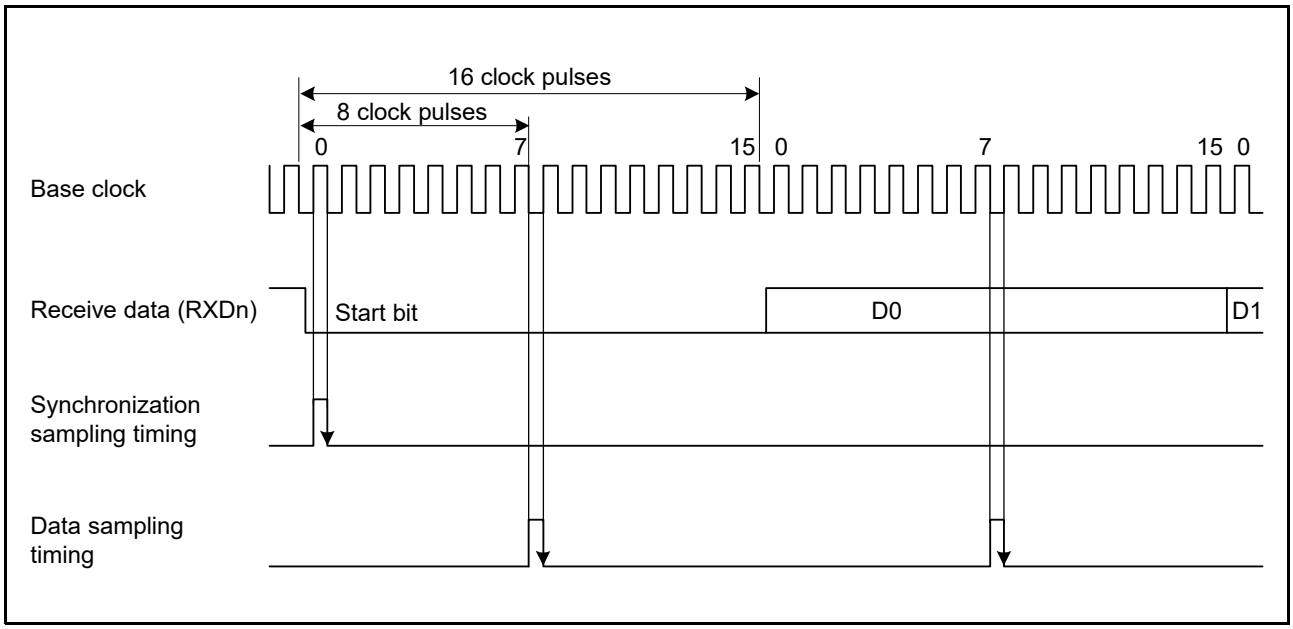


Figure 32.6 Receive Data Sampling Timing in Asynchronous Mode

In SCI1, SCI5, and SCI6, there are functions to adjust the sampling point of the receive data and the transition timing of the transmit data against that the high width and low width of the signal changes affected by the devices on the transmission line.

32.3.2.1 Receive Data Sampling Timing Adjustment

When a waveform such that high width is different from low width because the difference between rising time and falling time is large is received, adjust the timing for data to be sampled at the center of the narrower pulse. When low width is narrower than high width, move the sampling timing forward, and when high width is narrower than low width, move the sampling timing backward.

When the TMGR.RTMG[3:0] bits are set to an offset to the default sampling point and then the SPTR.RTADJ bit is set to 1, received data is sampled at the specified position.

Figure 32.7 shows an example of the sampling timing adjustment.

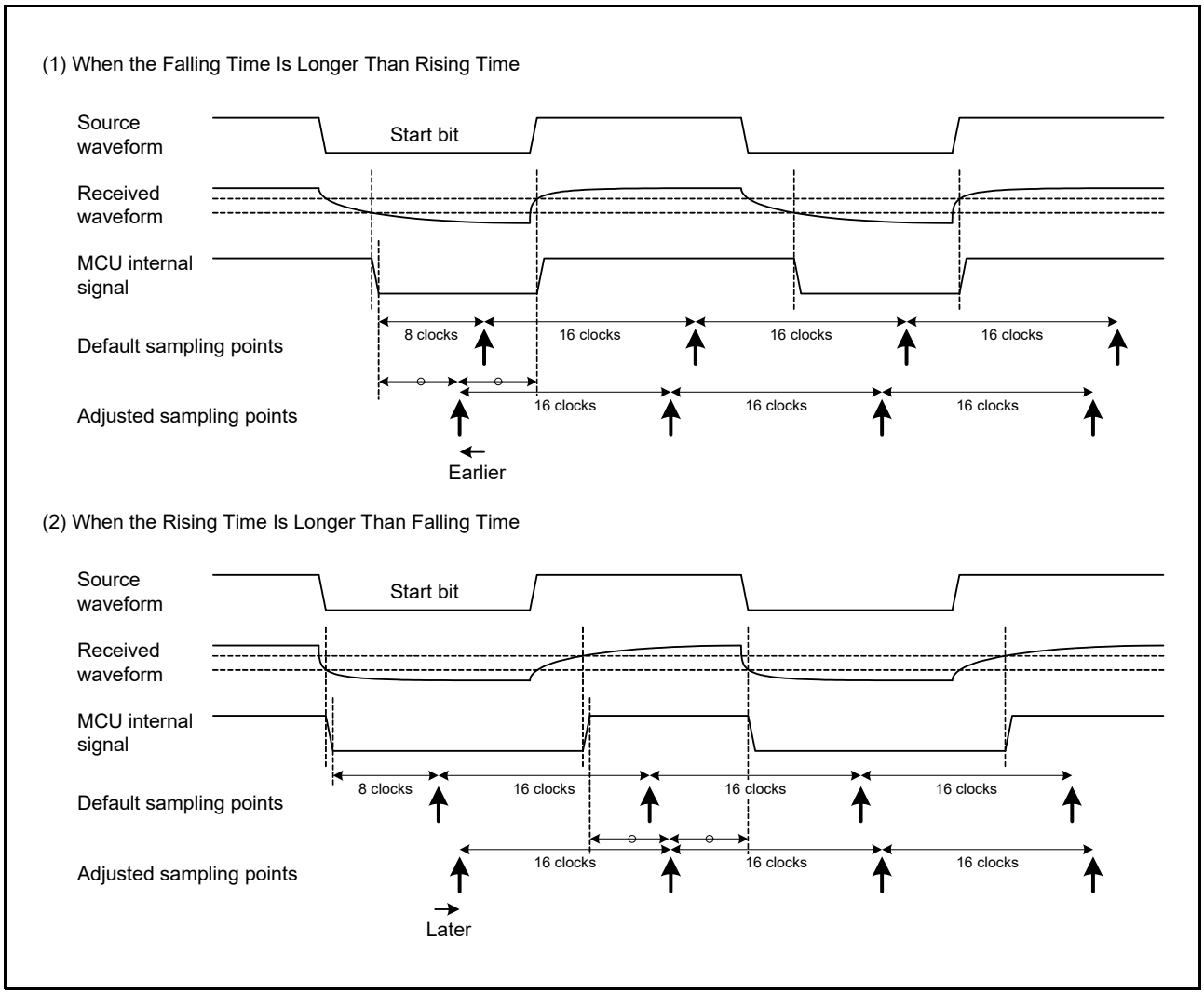


Figure 32.7 Example of Sampling Timing Adjustment (SEMR.ABCSE bit = 0 and SEMR.ABCS bit = 0)

32.3.2.2 Transmit Data Transition Timing Adjustment

When a difference between high width and low width for a waveform transmitted by this MCU arises at the receiver, it is also possible to make a difference between the high width and the low width beforehand at transmission to adjust so that the difference disappears at the receiver. When high width becomes narrower at the receiver, expand the high width of the transmit signal by delaying the falling edges, and when low width becomes narrower at the receiver, expand the low width of the transmit signal by delaying the rising edges.

When the TMGR.TTMG[3:0] bits are set to the transition direction and the delay amount and the SPTR.TTADJ bit is set to 1, transmit data changes at the specified point.

Figure 32.8 shows an example of the transition timing adjustment.

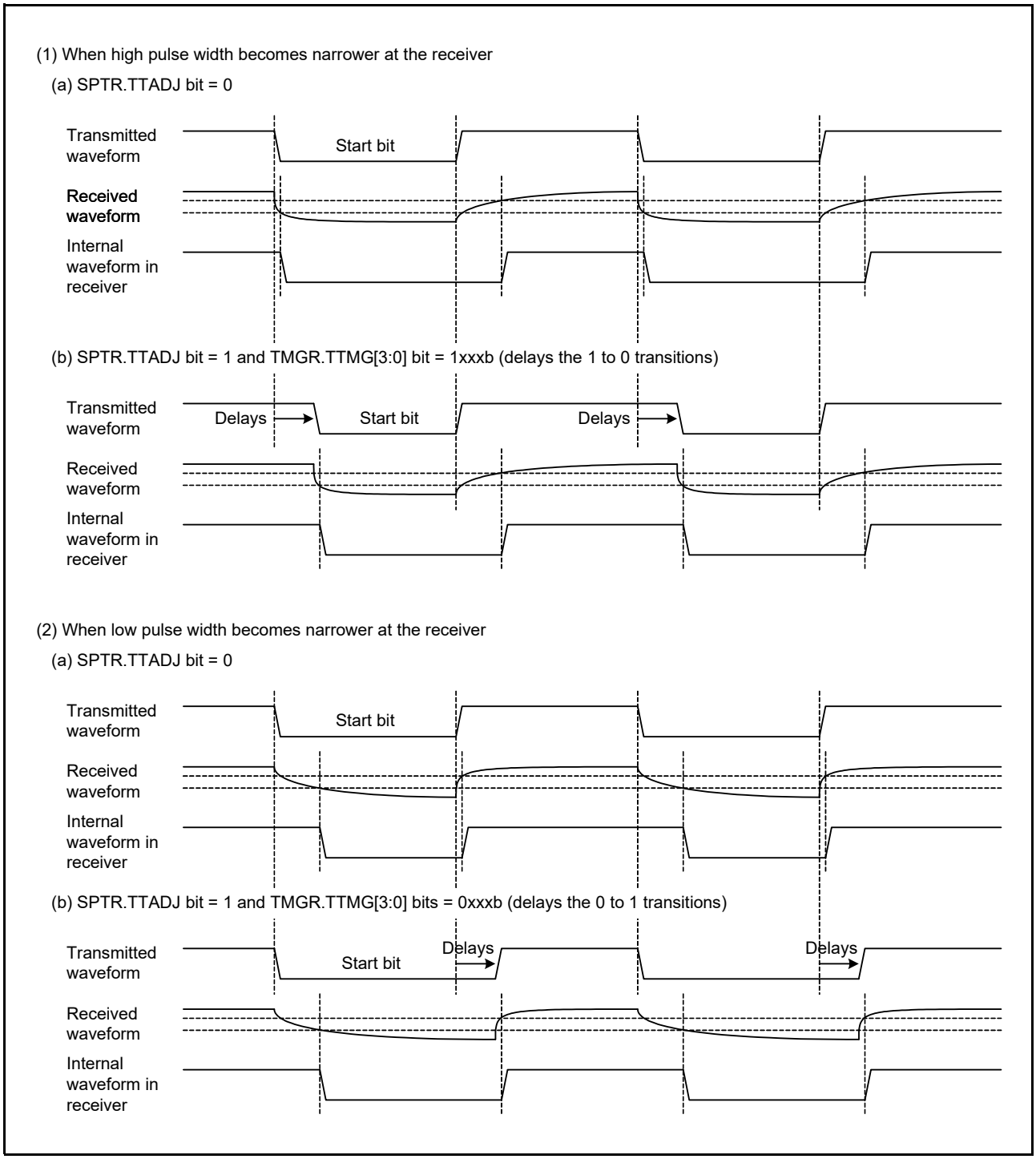


Figure 32.8 Example of Transition Timing Adjustment

32.3.3 Clock

Either an internal clock generated by the on-chip baud rate generator or an external clock input to the SCKn pin can be selected as the SCI's transfer clock, according to the setting of the SMR.CM bit and the SCR.CKE[1:0] bits.
When an external clock is input to the SCKn pin, the clock frequency should be 16 times the bit rate (when SEMR.ABCS bit = 0) and 8 times the bit rate (when SEMR.ABCS bit = 1). In addition, when an external clock is specified, the base clock of TMR0 and TMR1 can be selected by the SCIn.SEMR.ACS0 bit (n = 5, 6, 12).
When the SCI is operated on an internal clock, the clock can be output from the SCKn pin. The frequency of the clock output in this case is equal to the bit rate, and the phase is such that the rising edge of the clock is in the middle of the transmit data, as shown in Figure 32.9.

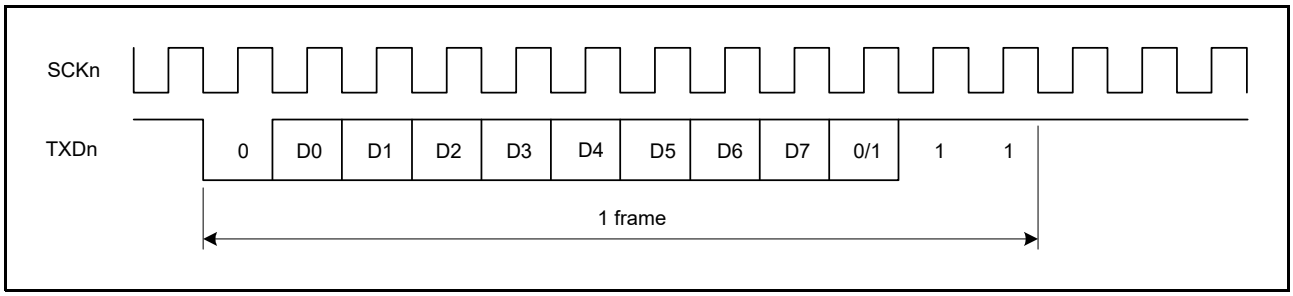


Figure 32.9 Phase Relationship between Output Clock and Transmit Data (Asynchronous Mode: SMR.CHR = 0, PE = 1, MP = 0, STOP = 1)

32.3.4 Double-Speed Mode and Divide-by-6 Mode

The output clock frequency of the on-chip baud rate generator is doubled by setting the SEMR.BGDM bit to 1, enabling high-speed communication at a doubled bit rate. If the SEMR.ABCS bit is set to 1 under the above condition, the number of base clock cycles changes from 16 to 8, so the bit rate becomes four times faster than the initial state.
When the SEMR.ABCSE bit is set to 1, the number of base clock pulses in 1-bit period becomes 6 and the period of the base clock becomes 1/2, where the bit rate becomes 16/3 times faster compared to a case that all of the ABCS, BDGM, and ABCSE bits are set to 0.
As shown by Formula (1) in section 32.3.2, Receive Data Sampling Timing and Reception Margin in Asynchronous Mode, setting the SEMR.ABCS bit to 1 changes the number of cycles to 8, and the sampling interval becomes longer. This causes the reception margin to decrease. Therefore, setting the SEMR.BGDM bit to 1 and the SEMR.ABCS bit to 0 is recommended instead of setting the SEMR.BGDM bit to 0 and the SEMR.ABCS bit to 1 for high-speed operation at a doubled bit rate.

32.3.5 CTS and RTS Functions

The CTS function is the use of input on the CTSn# pin in transmission control. Setting the SPMR.CTSE bit to 1 enables the CTS function. When the CTS function is enabled, placing the low level on the CTSn# pin causes transmission to start.
Applying the high level to the CTS# pin while transmission is in progress does not affect transmission of the current frame, which continues.
In the RTS function, by using the function of output on the RTSn# pin, a low level is output when reception becomes possible. Conditions for output of the low and high level are shown below.

[Conditions for low-level output]
When the following conditions are all satisfied.

- The SCR.RE bit is 1.
- Reception is not in progress.
- There are no received data yet to be read.
- The ORER, FER, and PER flags in the SSR register are all 0.

[Condition for high-level output]
When the conditions for low-level output are not satisfied.

Note that either one of CTS and RTS can be selected.

32.3.6 Data Match Detection

The data match detection function is available in asynchronous mode for SCI1, SCI5, and SCI6.
When the DCCR.DCME bit is set to 1, the received data is compared with the contents of the CDR.CMPD[8:0] bits*1.
Upon a match of the value, a receive data full interrupt (RXI) request is generated.
When the SMR.MP bit is 0, all received data is compared.
When the SMR.MP bit is set to 1 and if the DCCR.IDSEL bit is 1, only the data in which the multiprocessor bit is 1 is compared and data in which the bit is 0 is ignored. When the DCCR.IDSEL bit is 0, all of the received data is compared regardless of the value of the multiprocessor bit.
The received data is not stored or the flag is not updated until the received data matches with the value of the CDR.CMPD[8:0] bits. When the data is matched, the DCCR.DCME bit is automatically set to 0 and the DCMF flag becomes 1. At this time, if the DCCR.IDSEL bit is 1, the SCR.MPIE bit is automatically set to 0. In addition, an receive data full interrupt (RXI) request is generated when the SCR.RIE bit is 1.
When a framing error is found in the matched data, the DCCR.DFER flag becomes 1. When a parity error is found in the matched data, the DCCR.DPER flag becomes 1. The received data matched with the value of the CDR.CMPD[8:0] bits is not stored in the received buffer or the SSR.RDRF flag does not become 1.
After a data match is detected and the DCCR.DCME bit is set to 0, data is normally received.
When the DCCR.DFER or DCCR.DPER flag is 1, a data match is not detected. These flags should be set to 0 before enabling the data match detection function.

Note 1. Only the portion that corresponds to the character length specified by the SMR.CHR and SCMR.CHR1 bits is compared.

Figure 32.10 and Figure 32.11 show examples of data match detection.

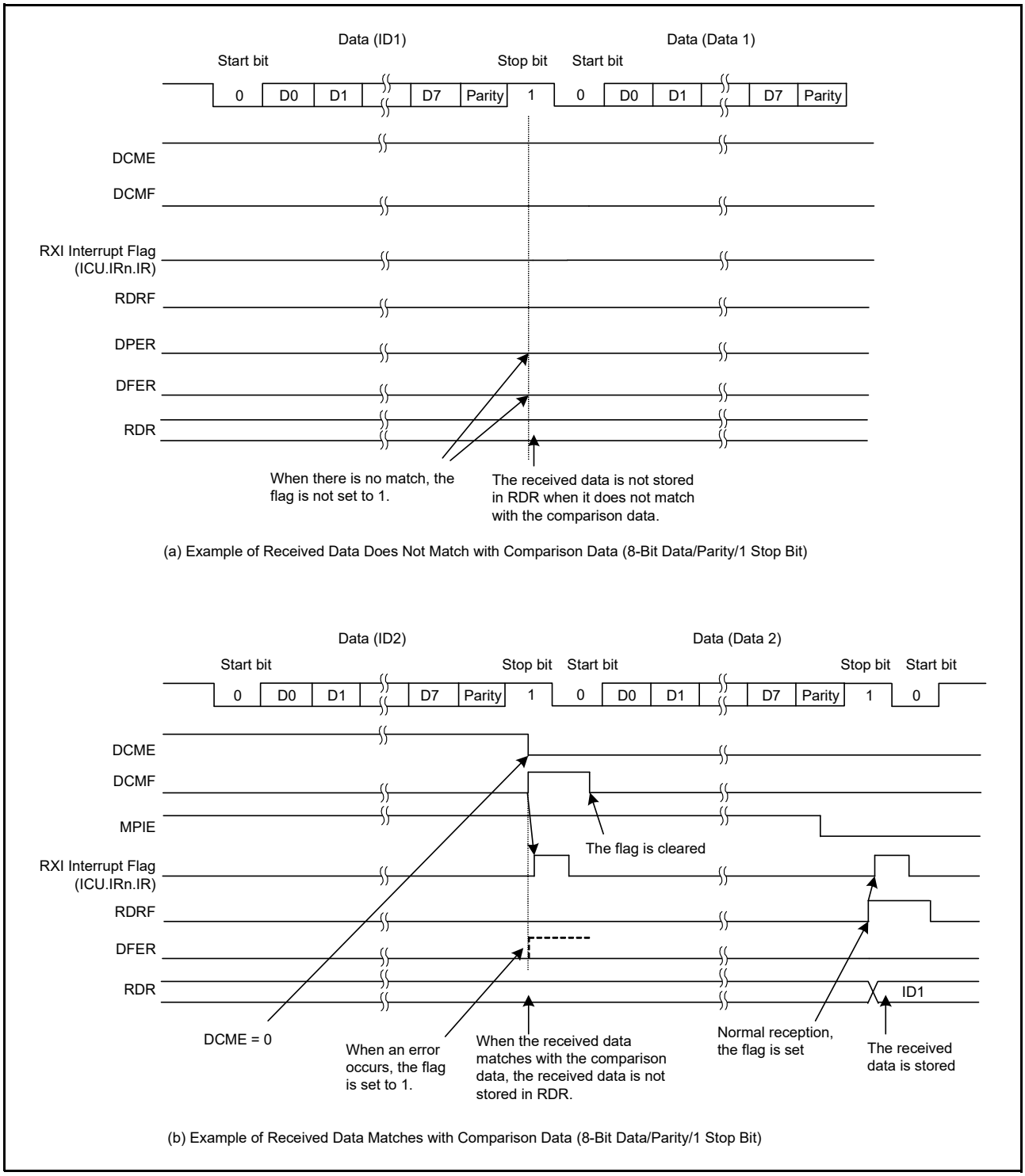


Figure 32.10 Example of Data Match Detection (1) Non Multi-Processor Mode

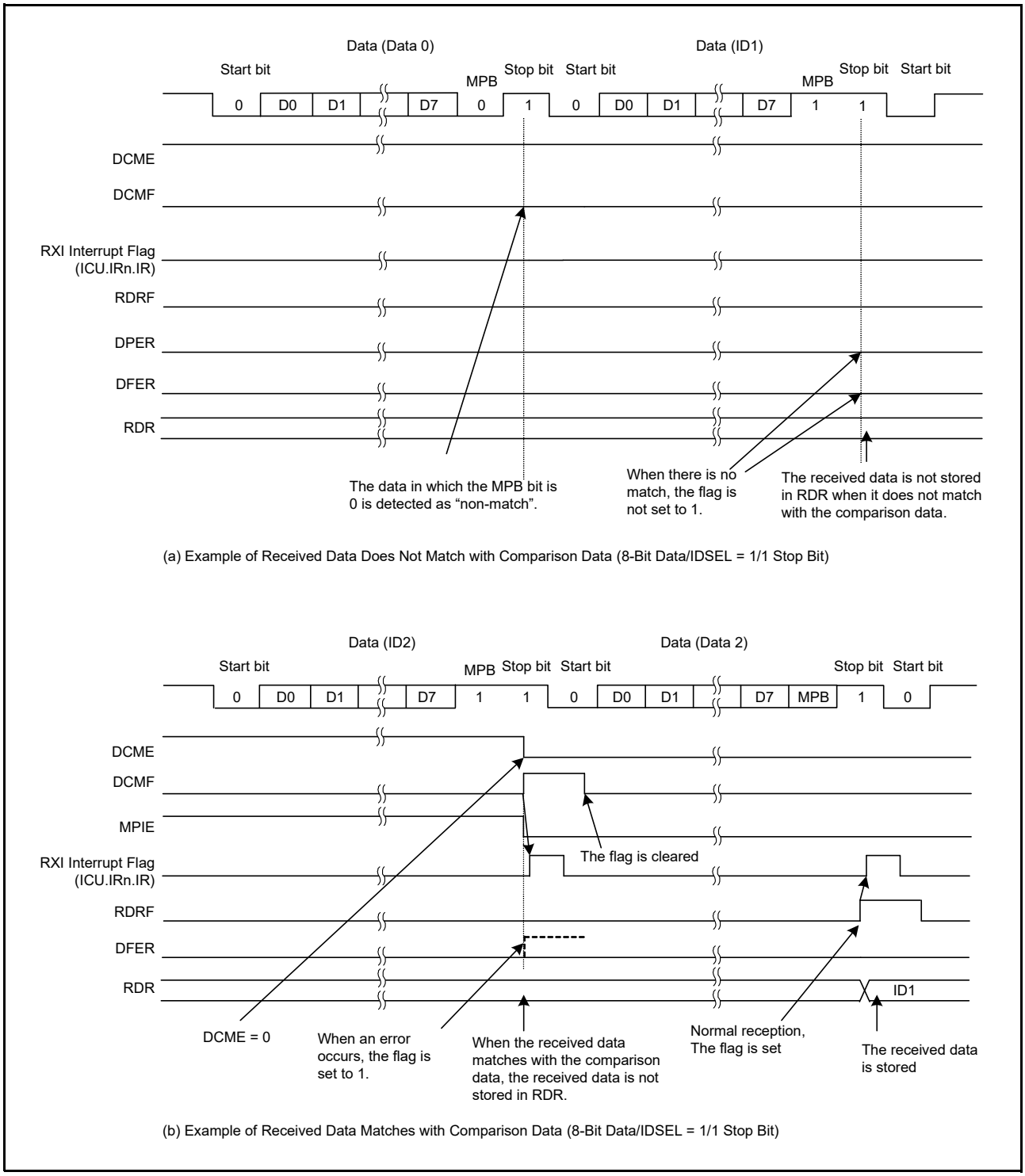


Figure 32.11 Example of Data Match Detection (2) Non Multi-Processor Mode

32.3.7 SCI Initialization (Asynchronous Mode)

Before transmitting and receiving data, start by writing the initial value 00h to the SCR register and then continue through the procedure for SCI given in Figure 32.12. Whenever the operating mode or transfer format is changed, the SCR register must be initialized before the change is made.

When the external clock is used in asynchronous mode, ensure that the clock signal is supplied even during initialization.

Note that setting the SCR.RE bit to 0 initializes neither the ORER, FER, PER, and RDRF flags in the SSR register nor registers RDR, RDRH, and RDRL.

In addition, note that setting bits TIE, TE, and TEIE in the SCR register to 1 simultaneously leads to the generation of a transmit end interrupt (TEI) request before the generation of a transmit data empty interrupt (TXI) request.

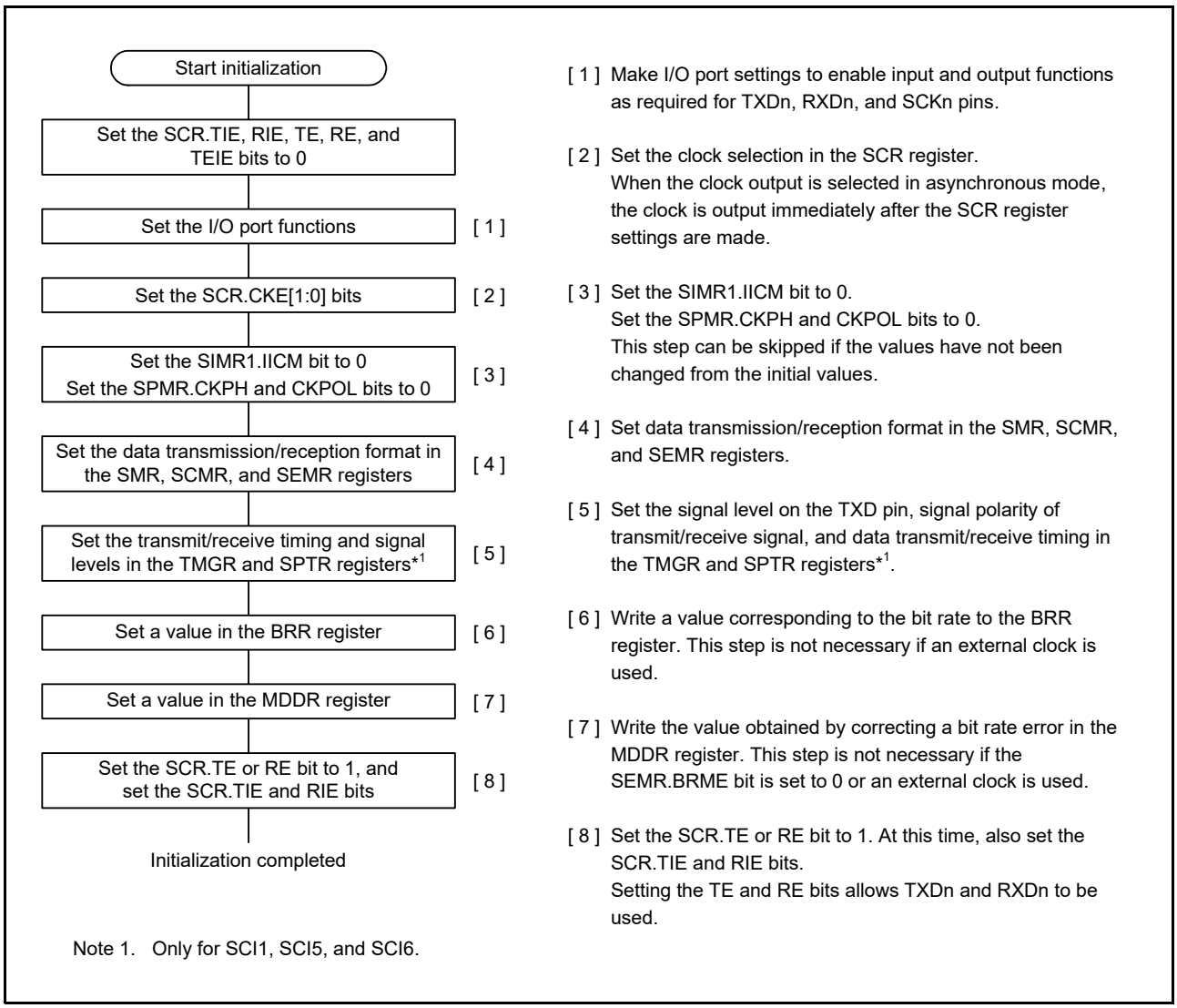


Figure 32.12 Sample SCI Initialization Flowchart (Asynchronous Mode)

Figure 32.13 shows an example of data transmission when the SCI is set to asynchronous mode according to the flow described in Figure 32.12 after a reset. When the pin function is set to the TXD pin, it is still high-impedance because the SCR.TE bit is 0. When the transmit data is written after setting the TE bit to 1, a data transmission starts. After the TE bit is set to 1, one frame of high is output from TXD pin (internal wait time)*1 and then the data transmission starts.

Note 1. This is when the SEMR.ITE bit is 0. When the ITE bit is 1, the internal wait time is not inserted.

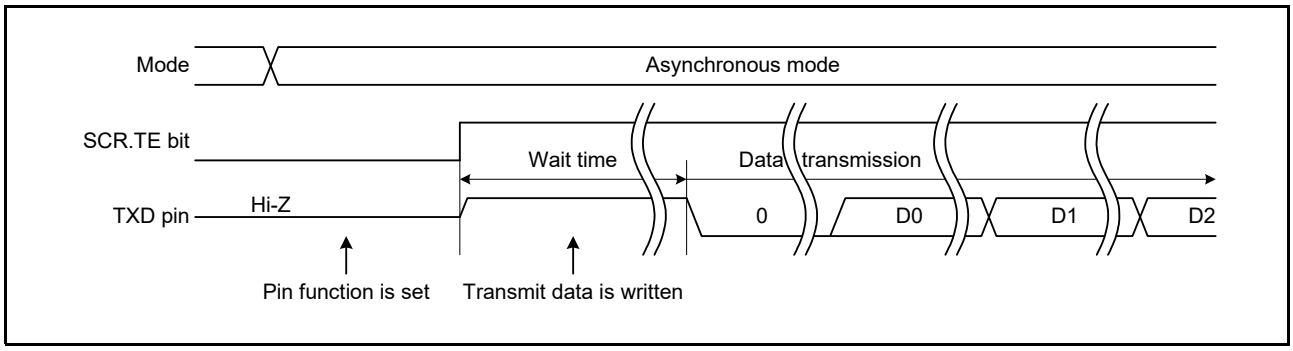


Figure 32.13 Example of Data Transmission Timing in Asynchronous Mode

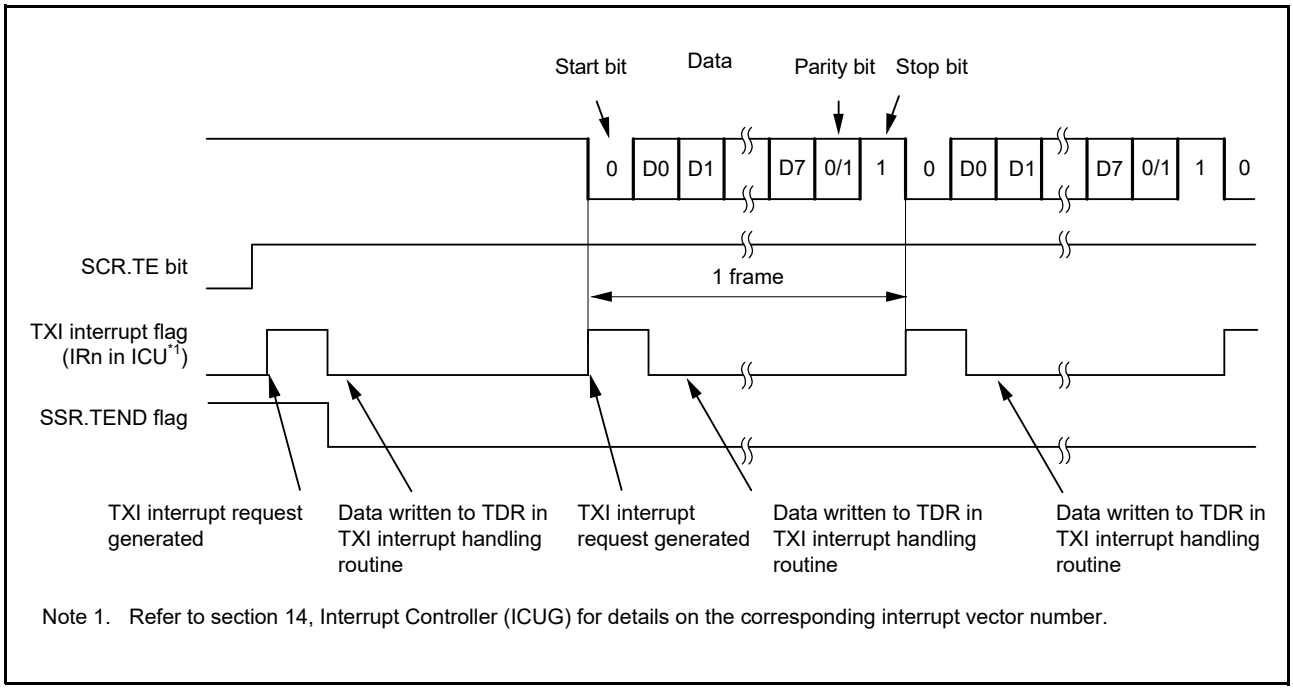
32.3.8 Serial Data Transmission (Asynchronous Mode)

Figure 32.14 to Figure 32.16 show an example of the operation for serial transmission in asynchronous mode.
In serial transmission, the SCI operates as described below.

1. The SCI transfers data from the TDR register*1 to the TSR register when data is written to the TDR register*1 in the TXI interrupt handling routine. The TXI interrupt request at the beginning of transmission is generated when the SCR.TE bit is set to 1 after the SCR.TIE bit is set to 1 or when these 2 bits are set to 1 simultaneously by a single instruction.
2. Transmission starts after the SPMR.CTSE bit is set to 0 (CTS function is disabled) and a low level on the CTSn# pin causes data transfer from the TDR register*1 to the TSR register. If the SCR.TIE bit is 1 at this time, a TXI interrupt request is generated. Continuous transmission is obtainable by writing the next transmit data to the TDR register*1 in the TXI interrupt handling routine before transmission of the current transmit data is completed. When TEI interrupt requests are in use, set the SCR.TIE bit to 0 (a TXI interrupt request is disabled) and the SCR.TEIE bit to 1 (a TEI interrupt request is enabled) after the last of the data to be transmitted are written to the TDR register*1, *2 from the handling routine for TXI requests.
3. Data is sent from the TXDn pin in the following order: start bit, transmit data, parity bit or multi-processor bit (may be omitted depending on the format), and stop bit.
4. The SCI checks for updating of (writing to) the TDR register*3 at the time of stop bit output.
5. When the TDR register*3 is updated, setting of the SPMR.CTSE bit to 0 (CTS function is disabled) or a low level input on the CTSn# pin cause the next transfer of the next transmit data from the TDR register*1 to the TSR register and sending of the stop bit, after which serial transmission of the next frame starts.
6. If the TDR register*3 is not updated, the SCI sets the SSR.TEND flag to 1, sends the stop bit, and then outputs high to put the line in the mark state. If the SCR.TEIE bit is 1 at this time, the SSR.TEND flag is set to 1 and a TEI interrupt request is generated.

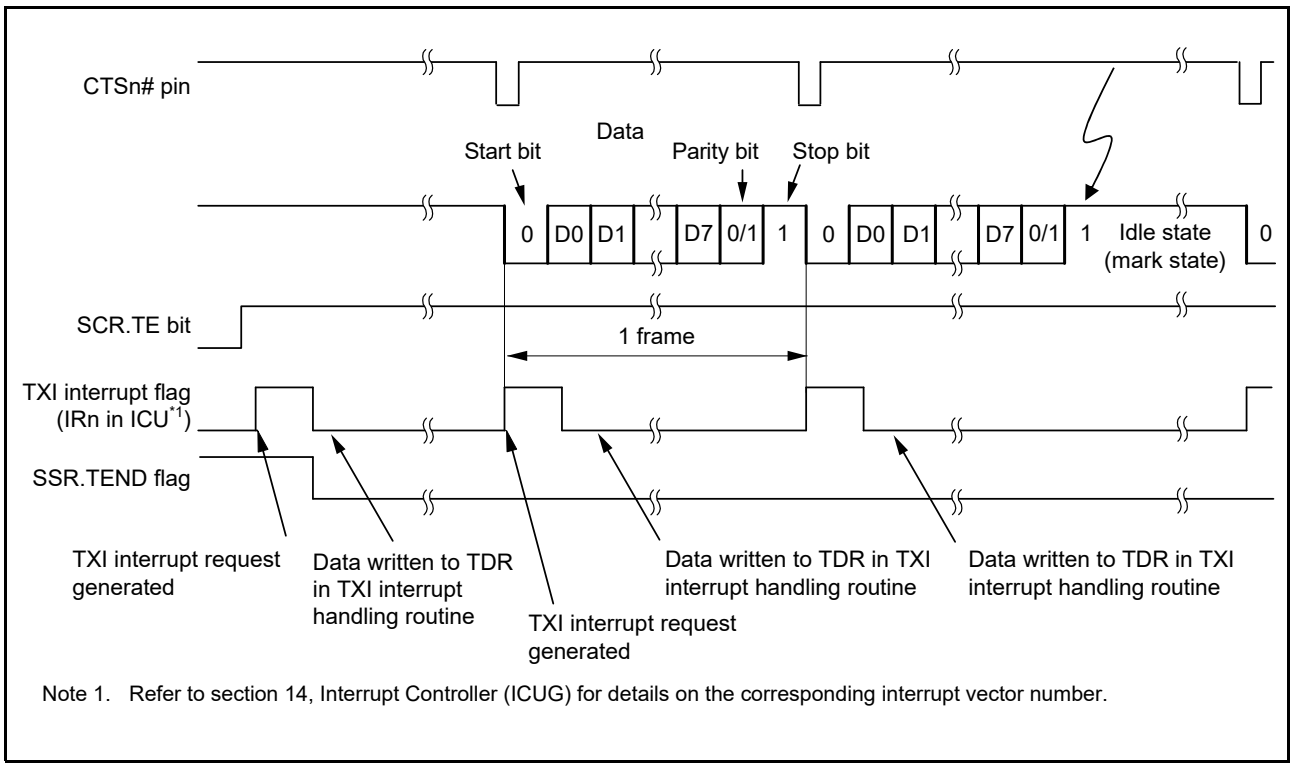
Note 1. Write data not to the TDR register but to the TDRH and TDRL registers when 9-bit data length is selected.
Note 2. Write data in the order from the TDRH register to the TDRL register when 9-bit data length is selected.
Note 3. The SCI checks for updating of the TDRL register only and does not check for updating of the TDRH register when 9-bit data length is selected.

Figure 32.17 shows a sample flowchart for serial transmission in asynchronous mode.



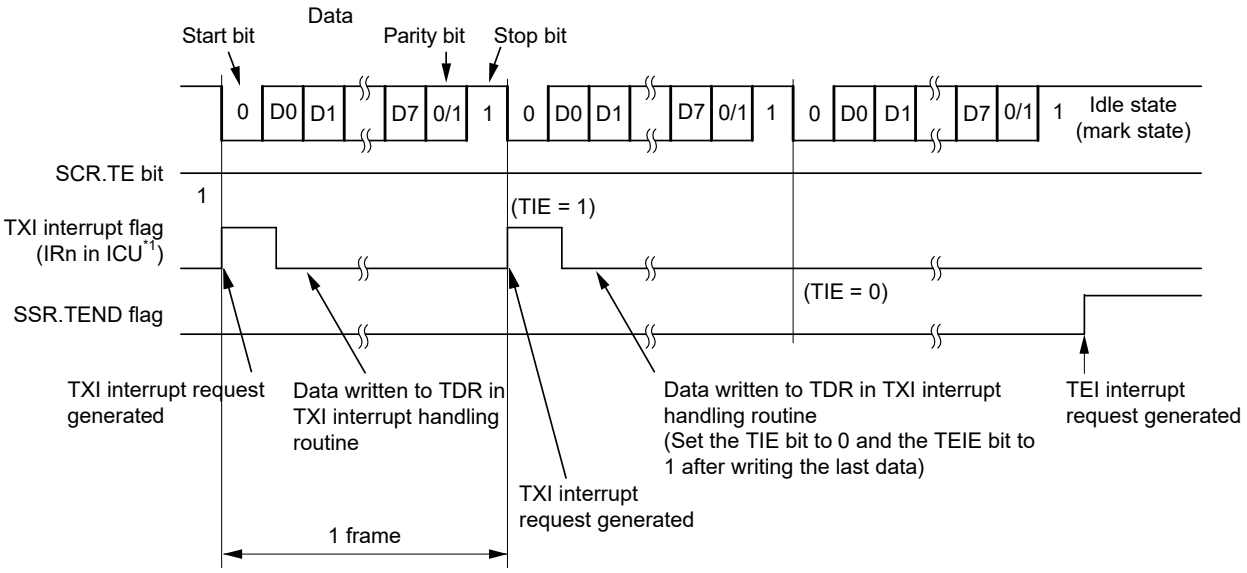
Note 1. Refer to section 14, Interrupt Controller (ICUG) for details on the corresponding interrupt vector number.

Figure 32.14 Example of Operation for Serial Transmission in Asynchronous Mode (1) (with 8-Bit Data, Parity, 1 Stop Bit, CTS Function Not Used, at the Beginning of Transmission)



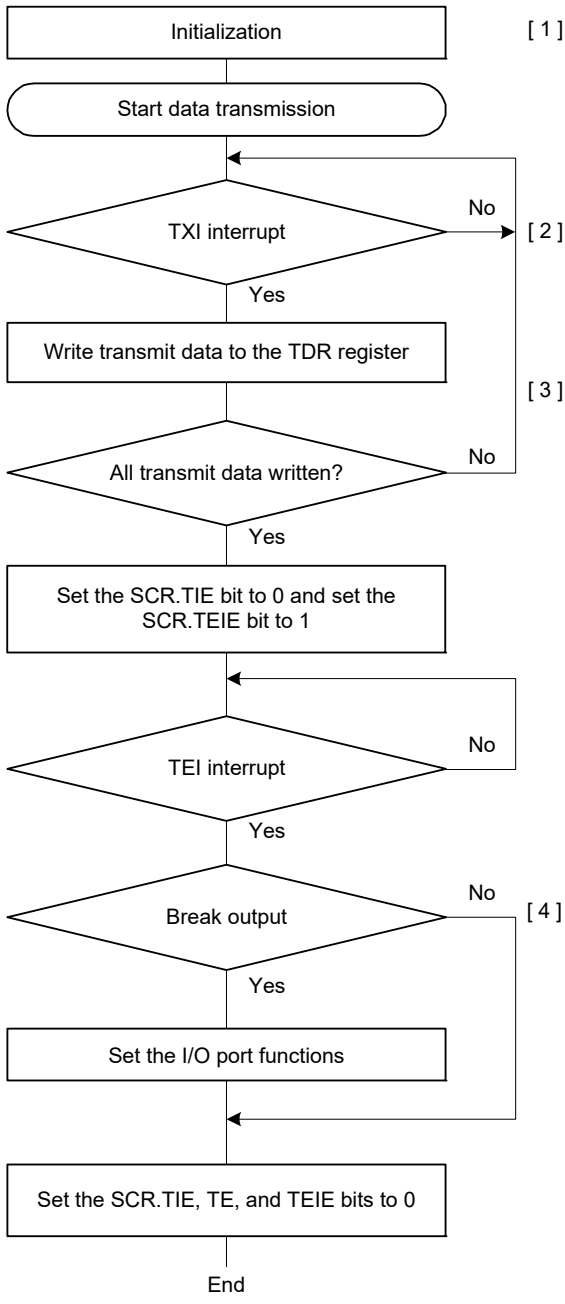
Note 1. Refer to section 14, Interrupt Controller (ICUG) for details on the corresponding interrupt vector number.

Figure 32.15 Example of Operation for Serial Transmission in Asynchronous Mode (2) (with 8-Bit Data, Parity, 1 Stop Bit, CTS Function Used, at the Beginning of Transmission)



Note 1. Refer to section 14, Interrupt Controller (ICUG) for details on the corresponding interrupt vector number.

Figure 32.16 Example of Operation for Serial Transmission in Asynchronous Mode (3) (with 8-Bit Data, Parity, 1 Stop Bit, CTS Function Not Used, from the Middle of Transmission until Transmission Completion)



[1] SCI initialization:
Set data transmission.
After the SCR.TE bit is set to 1, high is output for a frame*1, and transmission is enabled.

[2] Transmit data write to the TDR register by a TXI interrupt request:
When transmit data is transferred from the TDR register to the TSR register, a transmit data empty interrupt (TXI) request is generated.
Write transmit data to the TDR register once in the TXI interrupt handling routine.

[3] Serial transmission continuation procedure:
To continue serial transmission, write transmit data to the TDR register once using a TXI interrupt request.
Transmit data can also be written to the TDR register by activating the DMAC or DTC.
When TEI interrupt requests are in use, set the SCR.TIE bit to 0 and the SCR.TEIE bit to 1 after the last of the data to be transmitted are written to the TDR register.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, set the I/O port function corresponding to the TXDn pin (corresponding to output of the low level), and after switching the TXDn pin to the general I/O port function, set the SCR.TE bit to 0.

Note: The TDR register becomes the TDRH and TDRL registers when 9-bit data length is selected. Write data in the order from the TDRH register to the TDRL register.

Note 1. This is when the SEMR.ITE bit is 0. When the ITE bit is 1, transmission is enabled immediately after the TE bit is set to 1.

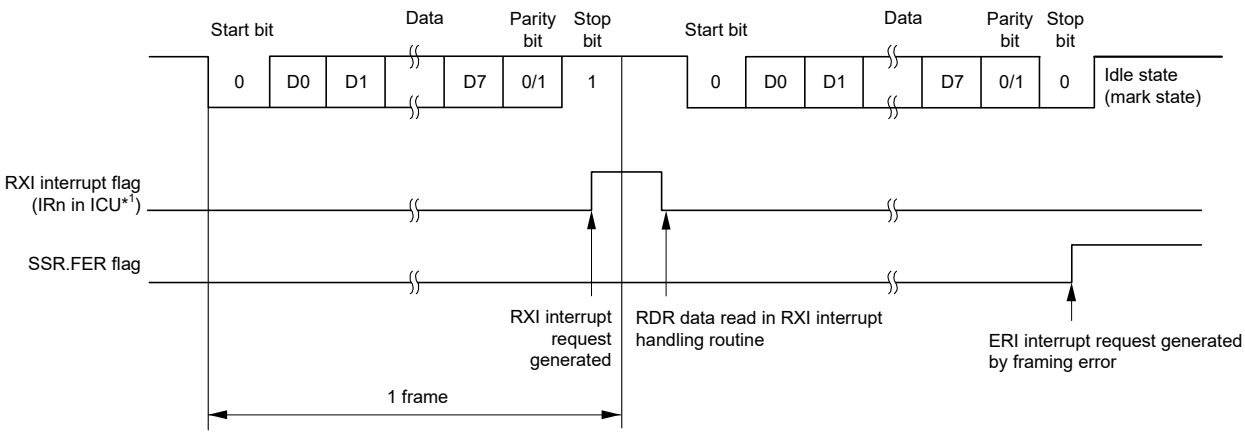
Figure 32.17 Example of Serial Transmission Flowchart in Asynchronous Mode

32.3.9 Serial Data Reception (Asynchronous Mode)

Figure 32.18 and Figure 32.19 show an example of the operation for serial data reception in asynchronous mode. In serial data reception, the SCI operates as described below.

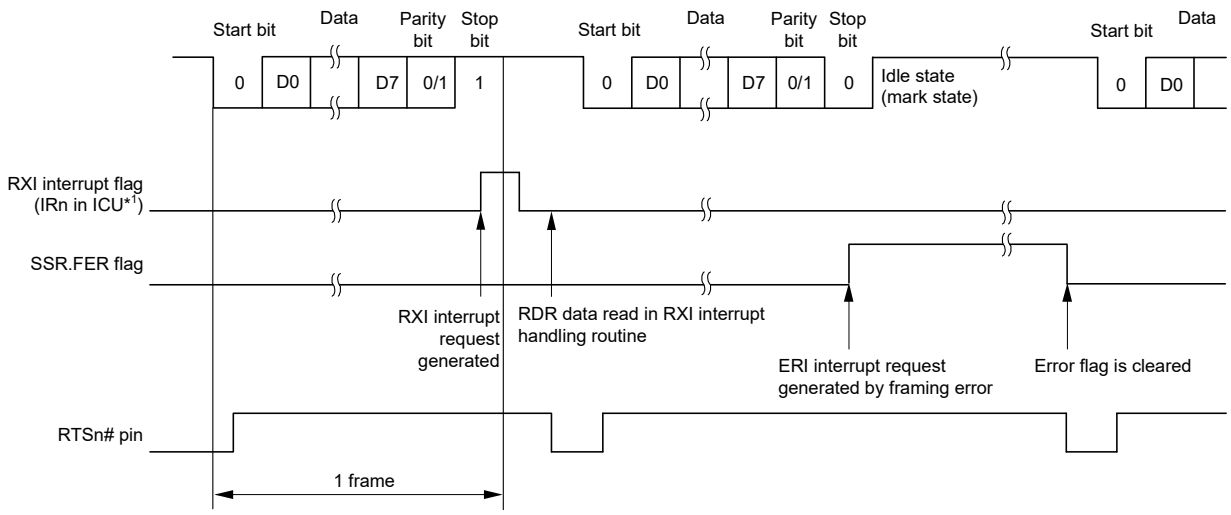
1. When the value of the SCR.RE bit becomes 1, the output signal on the RTSn# pin goes to the low level.
2. When the SCI monitors the communications line and detects a start bit, it performs internal synchronization, stores receive data in the RSR register, and checks the parity bit and stop bit.
3. If an overrun error occurs, the SSR.ORER flag is set to 1. If the SCR.RIE bit is 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to the RDR register*1.
4. If a parity error is detected, the SSR.PER flag is set to 1 and receive data is transferred to the RDR register*1. If the SCR.RIE bit is 1 at this time, an ERI interrupt request is generated.
5. If a framing error (when the stop bit is 0) is detected, the SSR.FER flag is set to 1 and receive data is transferred to the RDR register*1. If the SCR.RIE bit is 1 at this time, an ERI interrupt request is generated.
6. When reception finishes successfully, receive data is transferred to the RDR register*1. If the SCR.RIE bit is 1 at this time, an RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to the RDR register*1 in this RXI interrupt handling routine before reception of the next receive data is completed. Reading the received data that have been transferred to the RDR register*1 causes the RTSn# pin to output the low level.

Note 1. Read data not in the RDR register but in the RDRH and RDRL registers when 9-bit data length is selected.
Note 2. The SCI checks for reading of the RDRL register only and does not check for reading of the RDRH register when 9-bit data length is selected.



Note 1. Refer to section 14, Interrupt Controller (ICUG) for details on the corresponding interrupt vector number.

Figure 32.18 Example of SCI Operation for Serial Reception in Asynchronous Mode (1) (When RTS Function is Not Used) (Example with 8-Bit Data, Parity, 1 Stop Bit)



Note 1. Refer to section 14, Interrupt Controller (ICUG) for details on the corresponding interrupt vector number.

Figure 32.19 Example of SCI Operation for Serial Reception in Asynchronous Mode (2) (When RTS Function is Used) (Example with 8-Bit Data, Parity, 1 Stop Bit)

Table 32.34 lists the states of the status flags in the SSR register and receive data handling when a receive error is detected.

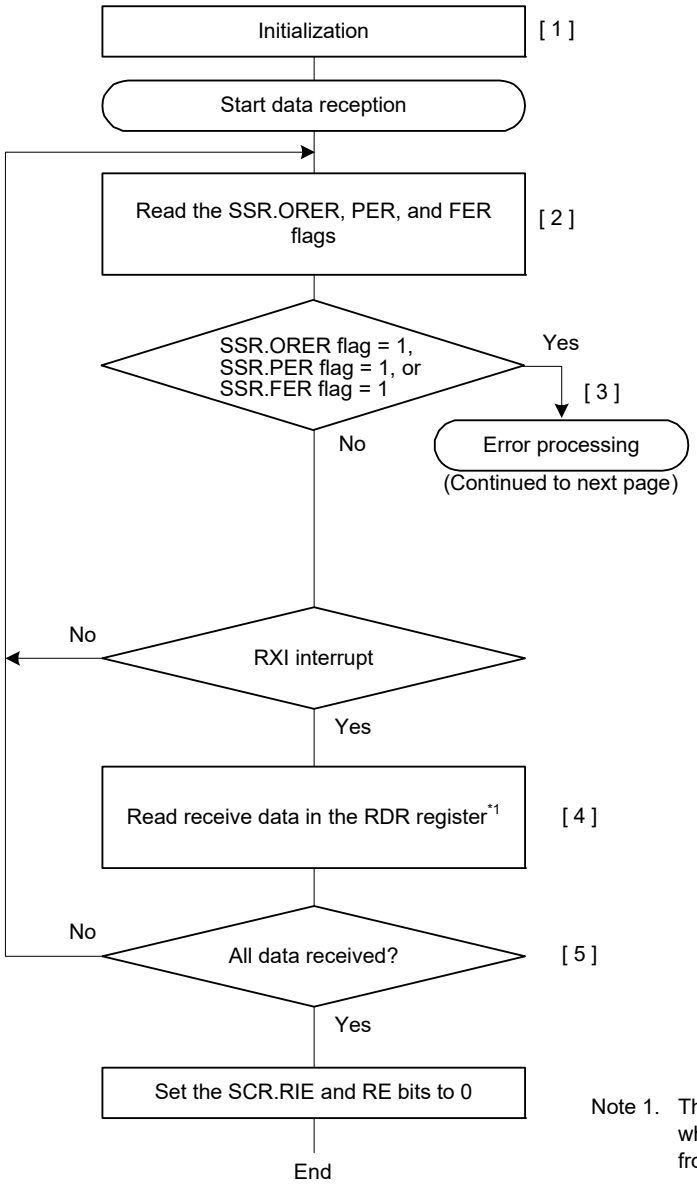
If a receive error is detected, an ERI interrupt request is generated but an RXI interrupt request is not generated. Data reception cannot be resumed while the receive error flag is 1. Accordingly, set the ORER, FER, and PER flags to 0 before resuming reception. Moreover, be sure to read the RDR (or the RDRL) register during overrun error processing. When a reception is forcibly terminated by setting the SCR.RE bit to 0 during operation, read the RDR (or the RDRL) register because received data which has not yet been read may be left in the RDR (or the RDRL) register.

Figure 32.20 and Figure 32.21 show samples of flowcharts for serial data reception.

Table 32.34 Status Flags in the SSR Register and Receive Data Handling

Status Flags in the SSR Register				
ORER	**FER**	**PER**	**Receive Data**	**Receive Error Type**
1	0	0	Lost	Overrun error
0	1	0	Transferred to the RDR register*1	Framing error
0	0	1	Transferred to the RDR register*1	Parity error
1	1	0	Lost	Overrun error + framing error
1	0	1	Lost	Overrun error + parity error
0	1	1	Transferred to the RDR register*1	Framing error + parity error
1	1	1	Lost	Overrun error + framing error + parity error

Note 1. Read data not in the RDR register but in the RDRH and RDRL registers when 9-bit data length is selected.



[1] SCI initialization:
Set data reception.

[2] [3] Receive error processing and break detection:
If a receive error occurs, an ERI interrupt is generated. An error is identified by reading the ORER, PER, and FER flags in the SSR register. After performing the appropriate error processing, be sure to set the ORER, PER, and FER flags to 0. Reception cannot be resumed if any of these flags is set to 1. In the case of a framing error, a break can be detected by reading the value of the input port corresponding to the RXDn pin.

[4] Read the receive data in the RDR register once in the RXI interrupt handling routine.

[5] Serial reception continuation procedure:
To continue serial reception, before the stop bit of the current frame is received, read data from the RDR register in the RXI interrupt handling routine.
The RDR data can also be read by activating the DMAC or DTC.

Note 1. The RDR register becomes the RDRH and RDRL registers when 9-bit data length is selected. Read data in the order from the RDRH register to the RDRL register.

Figure 32.20 Example Flowchart of Serial Reception in Asynchronous Mode (1)

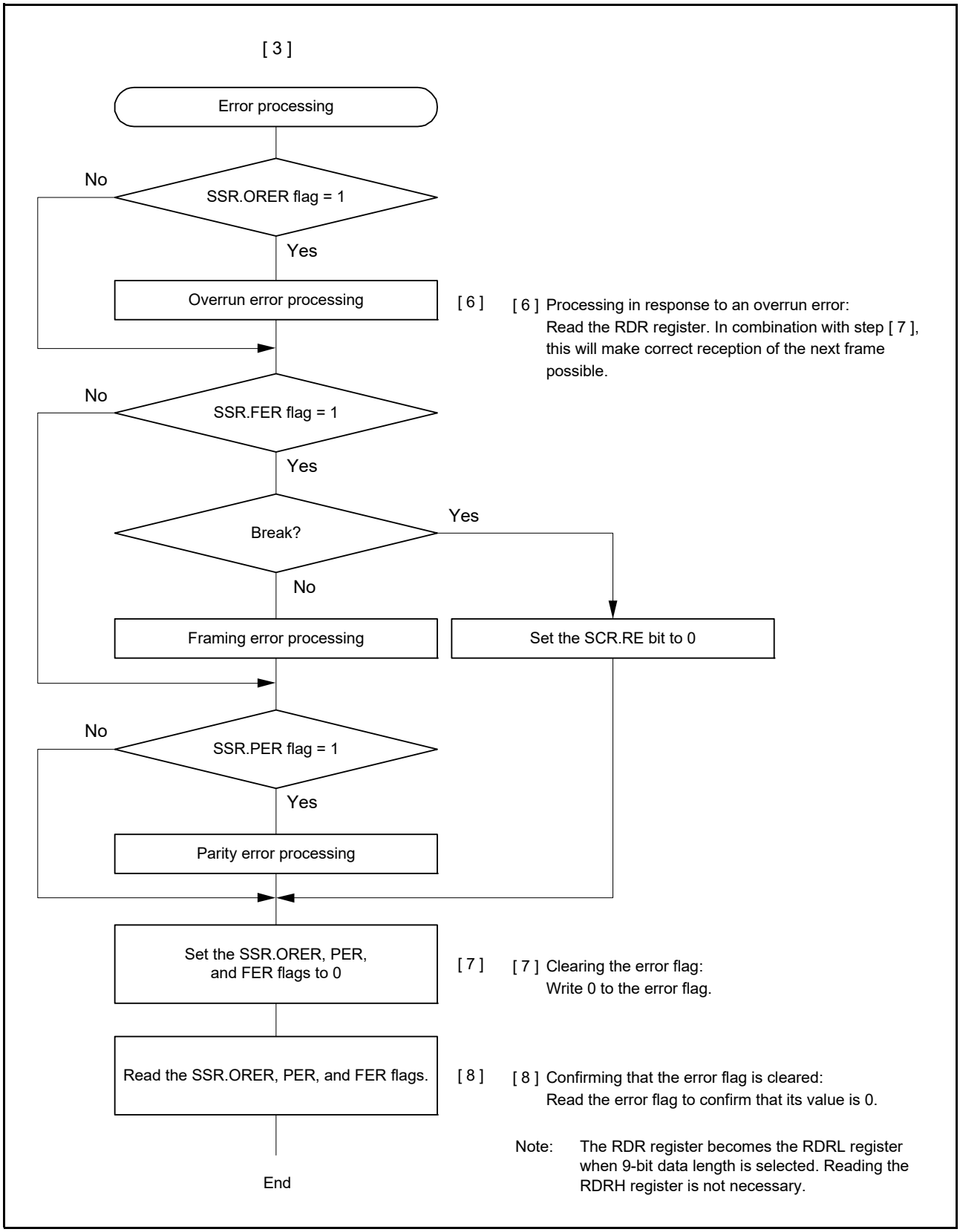


Figure 32.21 Example Flowchart of Serial Reception in Asynchronous Mode (2)

32.4 Multi-Processor Communications Function

Using the multi-processor communication functions enables to transmit and receive data by sharing a communication line between multiple processors by using asynchronous serial communication in which the multi-processor bit is added. In multi-processor communication, a unique ID code is allocated to each receiving station. Serial communication cycles consist of an ID transmission cycle to specify the receiving station and a data transmission cycle to transmit data to the specified receiving station. The multi-processor bit is used to distinguish between the ID transmission cycle and the data transmission cycle. When the multi-processor bit is set to 1, it indicates the ID transmission cycle and when the multi-processor bit is set to 0, it indicates the data transmission cycle. Figure 32.22 shows an example of communication between processors by using a multi-processor format. First, a transmitting station transmits communication data in which the multi-processor bit set to 1 is added to the ID code of the receiving station. Next, the transmitting station transmits the communication data in which the multi-processor bit set to 0 is added to the transmit data. Upon receiving the communication data in which the multi-processor bit is set to 1, the receiving station compares the received ID with the ID of the receiving station itself and if the two match, receives the communication data that is subsequently transmitted. If the received ID does not match with the ID of the receiving station, the receiving station skips the communication data until again receiving the communication data in which the multi-processor bit is set to 1.

For supporting this function, the SCI provides the SCR.MPIE bit. When the MPIE bit is set to 1, transfer of receive data from the RSR register to the RDR register (the RDRH and RDRL registers when 9-bit data length is selected), detection of a receive error, and setting the respective status flags RDRF, ORER, and FER in the SSR register are disabled until reception of data in which the multi-processor bit is set to 1. Upon receiving a reception character in which the multi-processor bit is set to 1, the SSR.MPB bit is set to 1 and the SCR.MPIE bit is automatically cleared, thus returning to a normal reception operation. During this time, an RXI interrupt is generated if the SCR.RIE bit is 1.

When the multi-processor format is specified, specification of the parity bit is disabled. Apart from this, there is no difference from the operation in the normal asynchronous mode. A clock which is used for the multi-processor communication is also the same as the clock used in the normal asynchronous mode.

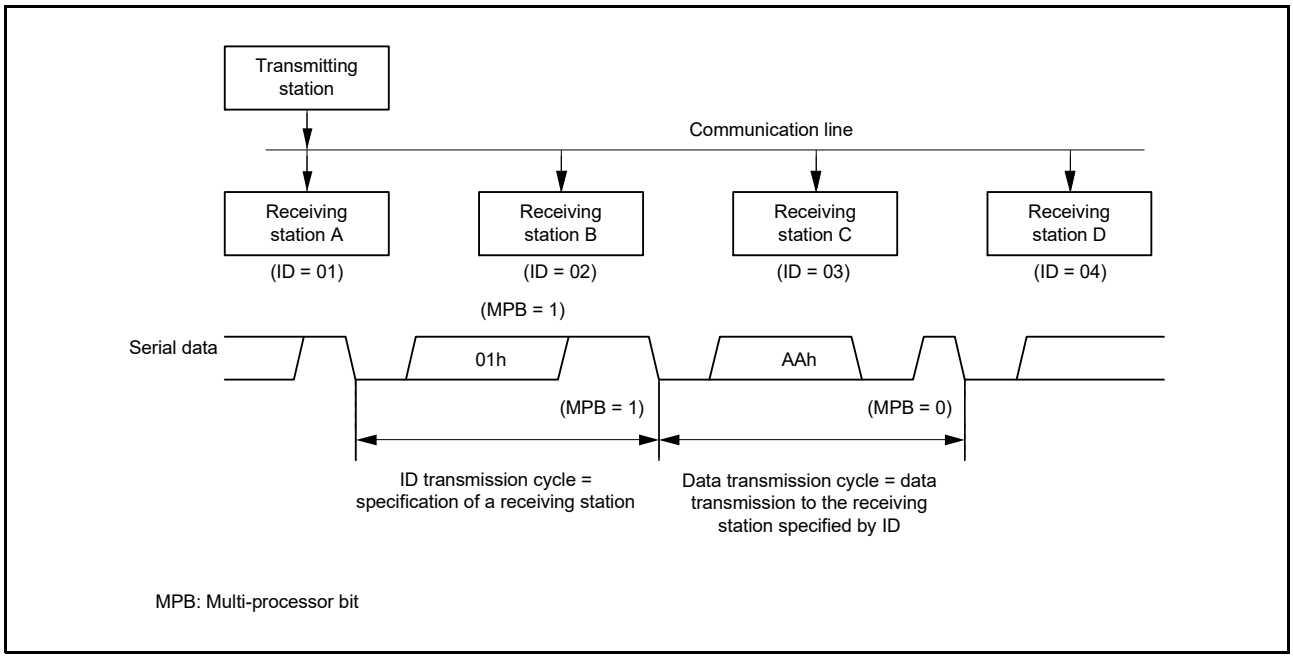


Figure 32.22 An Example of Communication using the Multi-Processor Format (Example of Transmission of Data AAh to Receiving Station A)

32.4.1 Multi-Processor Serial Data Transmission

Figure 32.23 is a sample flowchart of multi-processor data transmission. In the ID transmission cycle, the ID should be transmitted with the SSR.MPBT bit set to 1. In the data transmission cycle, the data should be transmitted with the MPBT bit set to 0. The other operations are the same as the operations in asynchronous mode.

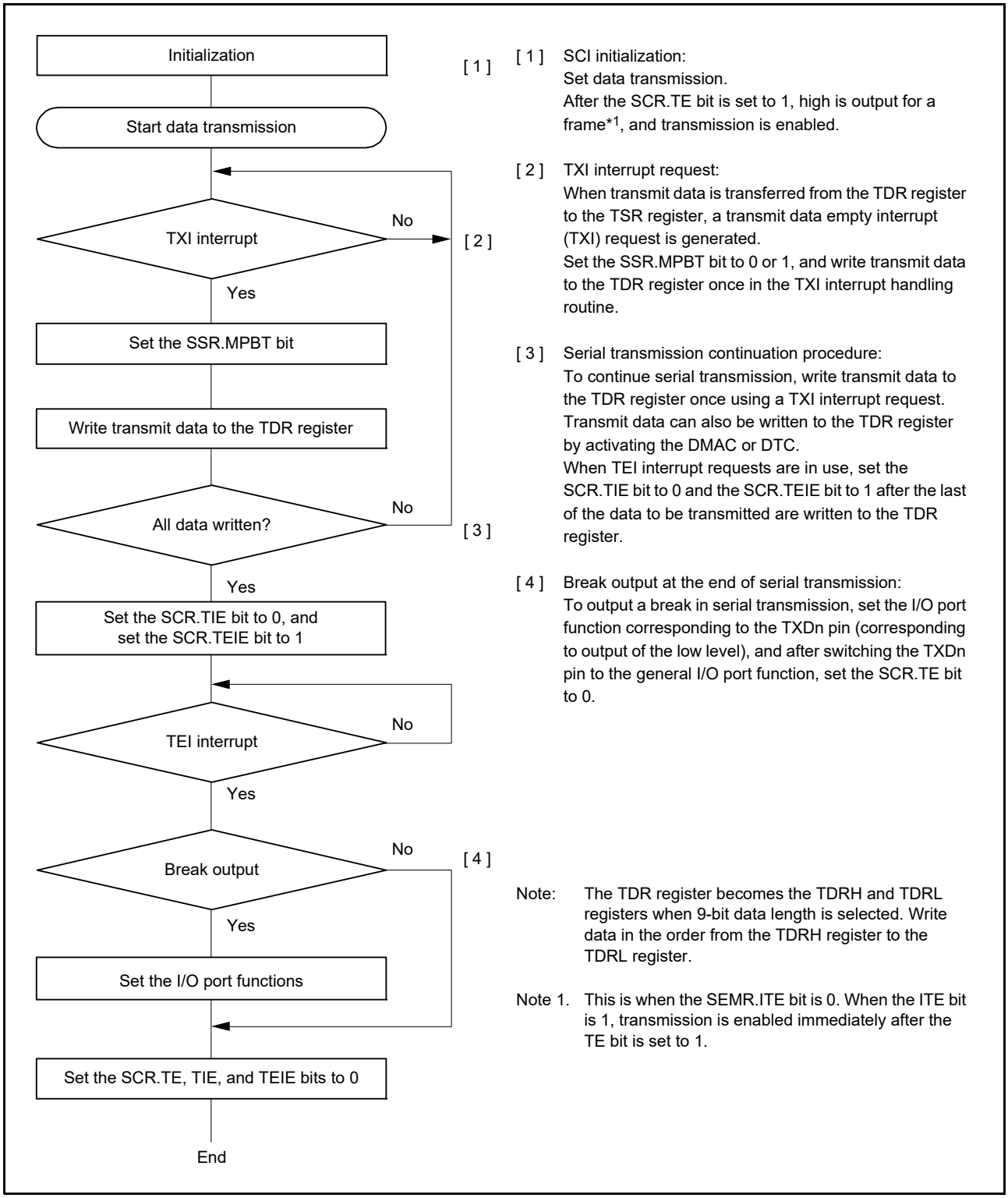
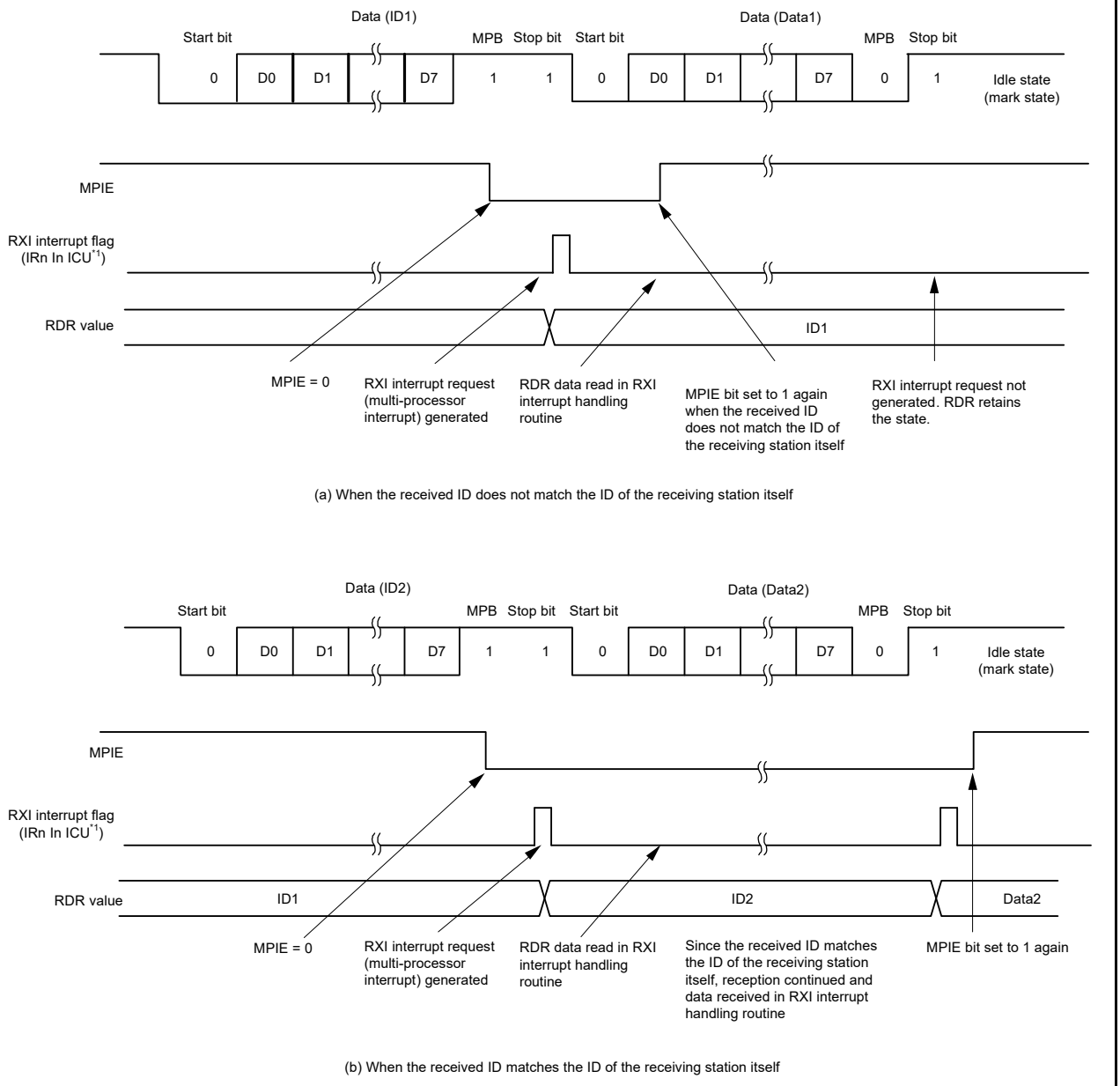


Figure 32.23 Example of Multi-Processor Serial Transmission Flowchart

32.4.2 Multi-Processor Serial Data Reception

Figure 32.25 and Figure 32.26 are sample flowcharts of multi-processor data reception. When the SCR.MPIE bit is set to 1, reading the communication data is skipped until reception of the communication data in which the multi-processor bit is set to 1. When the communication data in which the multi-processor bit is set to 1 is received, the received data is transferred to the RDR register (the RDRH and RDRL registers when 9-bit data length is selected). During this time, the RXI interrupt request is generated. The other operations are the same as the operations in asynchronous mode.
Figure 32.24 is the example of operation for reception.



Note 1. Refer to section 14, Interrupt Controller (ICUG) for details on the corresponding interrupt vector number.

Figure 32.24 Example of SCI Reception (8-Bit Data/Multi-Processor Bit/1 Stop Bit)

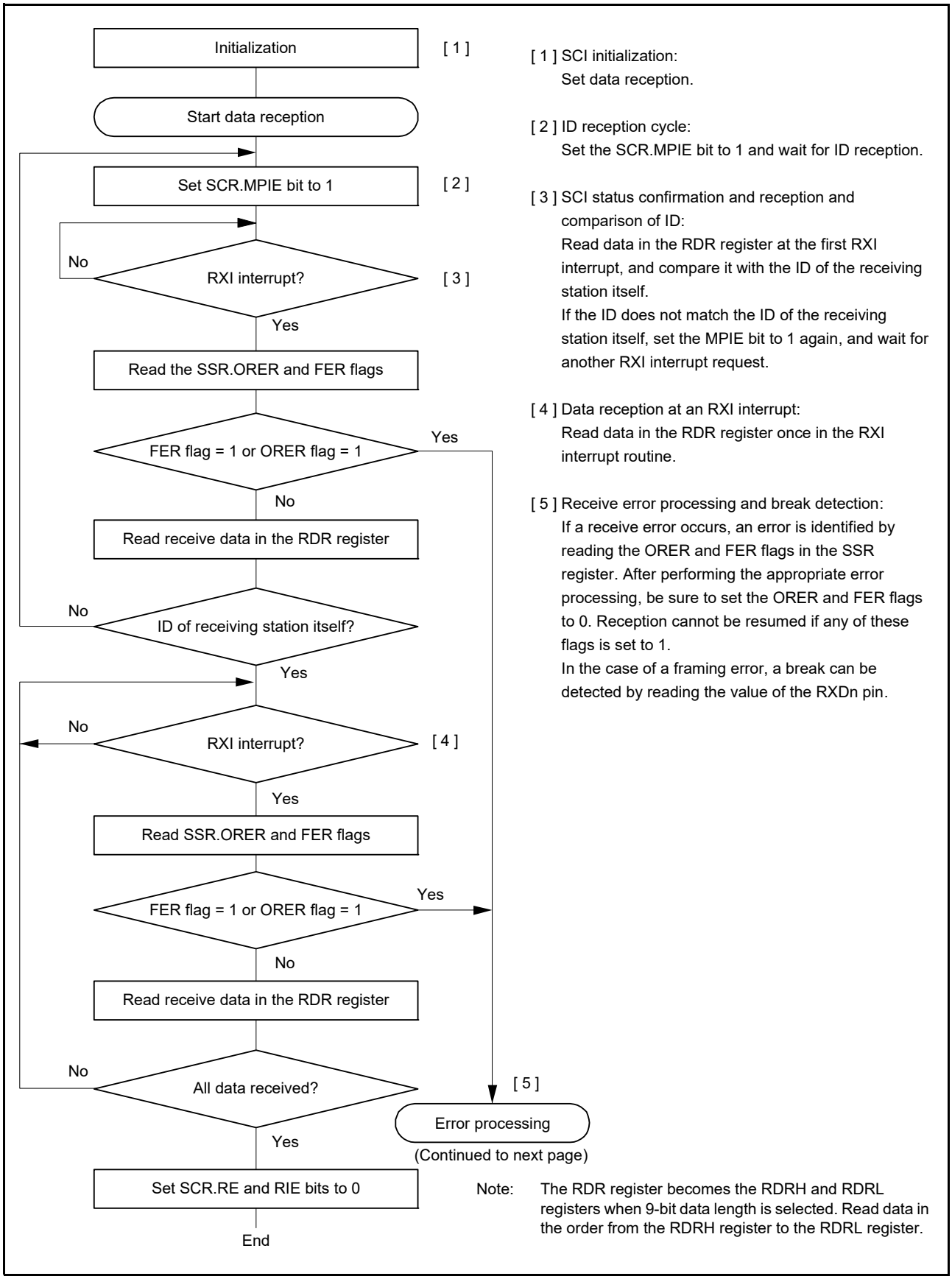


Figure 32.25 Example of Multi-Processor Serial Reception Flowchart (1)

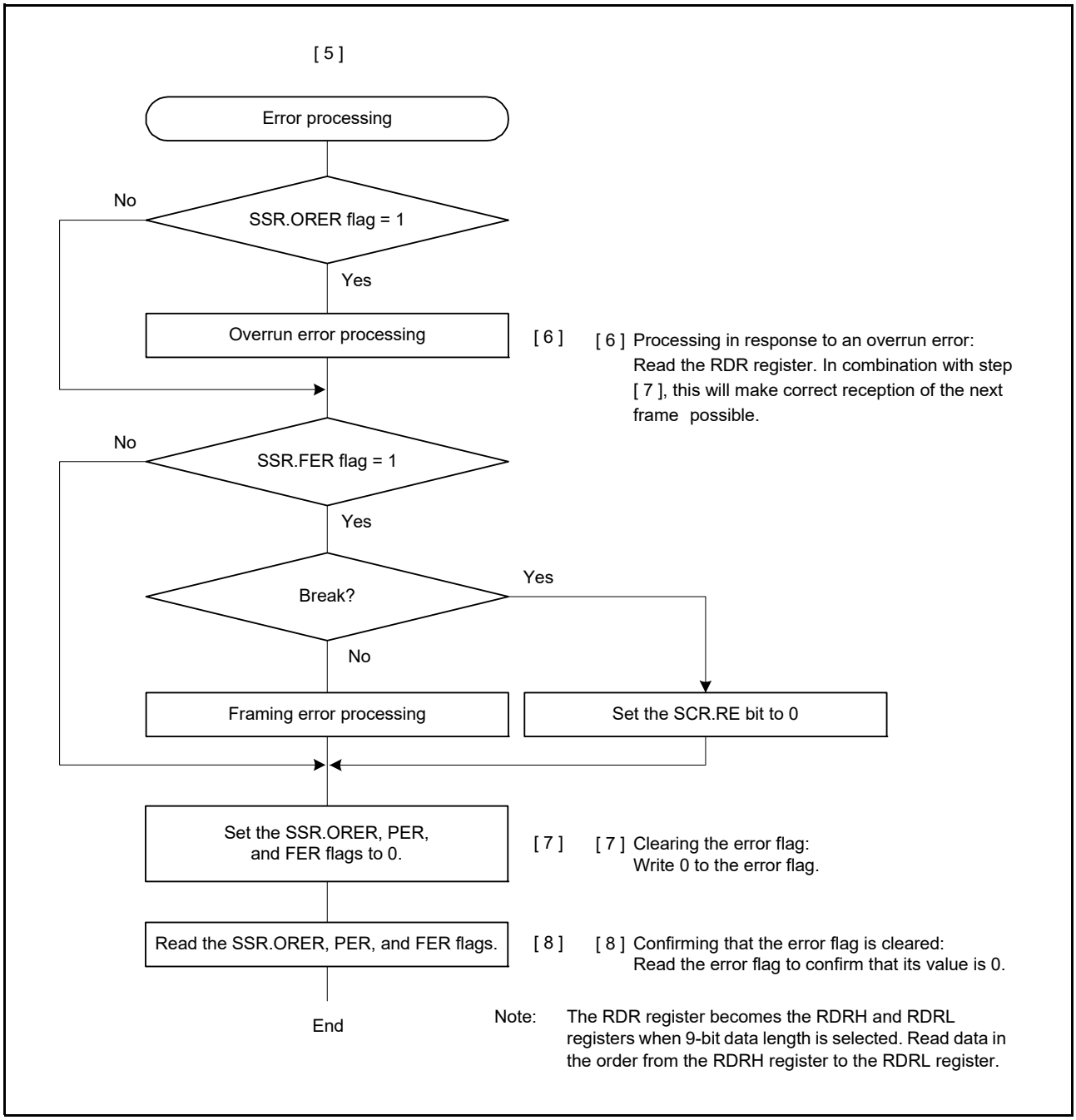


Figure 32.26 Example of Multi-Processor Serial Reception Flowchart (2)

32.5 Operation in Clock Synchronous Mode

Figure 32.27 shows the data format for clock synchronous serial data communications.
In clock synchronous mode, data is transmitted or received in synchronization with clock pulses. One character in transfer data consists of 8-bit data. In clock synchronous mode, no parity bit can be added.
In data transmission, the SCI outputs data from one falling edge of the synchronization clock to the next. In data reception, the SCI receives data in synchronization with the rising edge of the synchronization clock. After 8-bit data is output, the communication line holds the last bit output state.
Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communications by use of a common clock. Both the transmitter and the receiver also have a double-buffered structure, so that the next transmit data can be written during transmission or the previous receive data can be read during reception, enabling continuous data transfer.

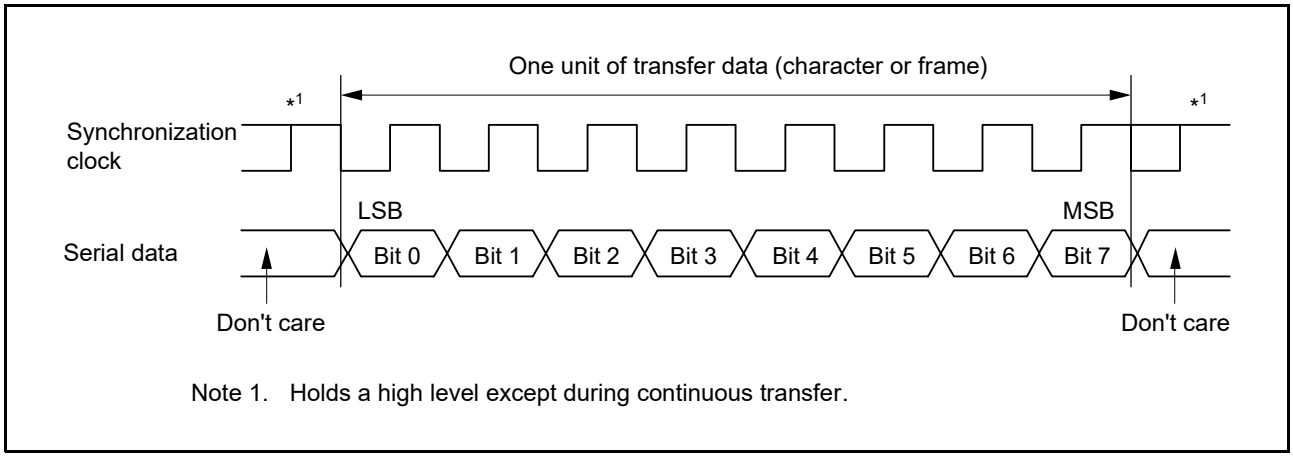


Figure 32.27 Data Format in Clock Synchronous Serial Communications (LSB First)

32.5.1 Clock

Either an internal clock generated by the on-chip baud rate generator or an external synchronization clock input at the SCKn pin can be selected, according to the setting of the SCR.CKE[1:0] bits.
When the SCI is operated on an internal clock, the synchronization clock is output from the SCKn pin. Eight synchronization clock pulses are output in the transfer of one character, and when no transfer is performed the clock is held high. However, when only data reception is performed while the CTS function is disabled, the synchronization clock output is started at the same time when the SCR.RE bit set to 1. The synchronization clock is stopped at the high level when an overrun error occurs or the SCR.RE bit is set to 0.
When only data reception is performed and the CTS function is enabled, the clock output is not started even when the SCR.RE bit set to 1 if the CTSn# pin input is high when the SCR.RE bit is 0. The synchronization clock output is started when the SCR.RE bit is set to 1 and the CTSn# pin input is low. After that, if the CTSn# pin input is high on completion of the frame reception, the synchronization clock output is stopped at the high level. If the CTSn# pin input continues to be low, the synchronization clock is stopped at the high level when an overrun error occurs or the SCR.RE bit is set to 0.

32.5.2 CTS and RTS Functions

In the CTS function, CTSn# pin input is used to control reception/transmission start when the clock source is the internal clock. Setting the SPMR.CTSE bit to 1 enables the CTS function.
When the CTS function is enabled, placing the low level on the CTSn# pin causes reception/transmission to start. Applying the high level to the CTS# pin while reception/transmission are in progress does not affect reception/transmission of the current frame, which continues.
In the RTS function, RTSn# pin output is used to request reception/transmission start when the clock source is an external synchronizing clock. A low level is output when serial communications become possible. Conditions for output of the low and high level are shown below.

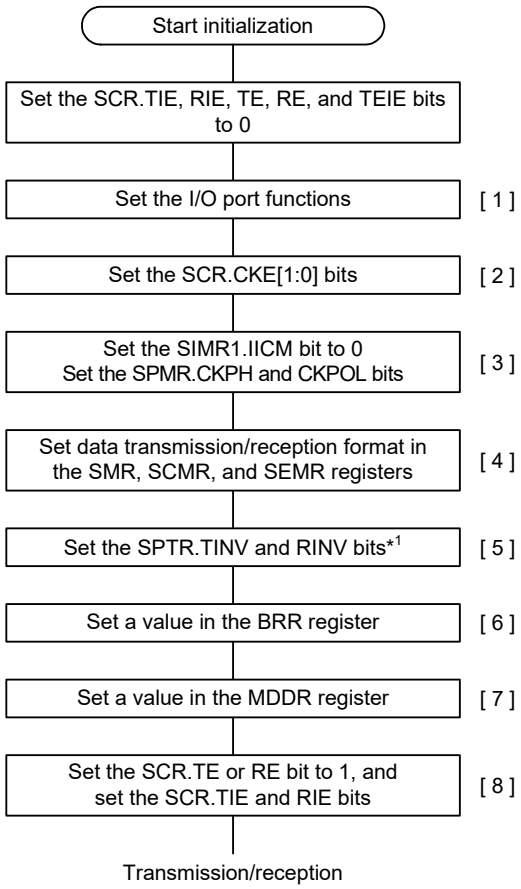
[Conditions for low-level output]
When the following conditions are all satisfied.

- The SCR.RE or SCR.TE bit is 1.
- Transmission or reception is not in progress.
- There are no received data yet to be read (when the SCR.RE bit is 1).
- Untransmitted data is present (when the SCR.TE bit is 1).
- The SSR.ORER flag is 0.

[Condition for high-level output]
The conditions for low-level output have not been satisfied.

32.5.3 SCI Initialization (Clock Synchronous Mode)

Before transmitting and receiving data, start by writing the initial value 00h to the SCR register and then continue through the procedure for SCI given in Figure 32.28. Whenever the operating mode or transfer format is changed, the SCR register must be initialized before the change is made.
Note that setting the SCR.RE bit to 0 initializes neither the ORER, FER, and PER flags in the SSR register nor the RDR register.



[1] Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.

[2] Set the clock selection in the SCR register.
When an internal clock is selected, the SCK pin functions as the clock output pin.

[3] Set the SIMR1.IICM bit to 0.
Set the SPMR.CKPH and CKPOL bits.
Step [3] can be skipped if the values have not been changed from the initial values.

[4] Set the data transmission/reception format in the SMR, SCMR, and SEMR registers.

[5] Set the polarities of transmit/receive signals by the SPTR.TINV and RINV bits*1.

[6] Write a value corresponding to the bit rate to the BRR register. This step is not necessary if an external clock is used.

[7] Write the value obtained by correcting a bit rate error in the MDDR register. This step is not necessary if the SEMR.BRME bit is set to 0 or an external clock is used.

[8] Set the SCR.TE or RE bit to 1. At this time, also set the SCR.TIE and RIE bits.
Setting the TE and RE bits allows TXDn and RXDn to be used.

Note: In simultaneous transmit and receive operations, the SCR.TE and RE bits should both be set to 0 or 1 simultaneously.
Note 1. Only for SCI1, SCI5, and SCI6.

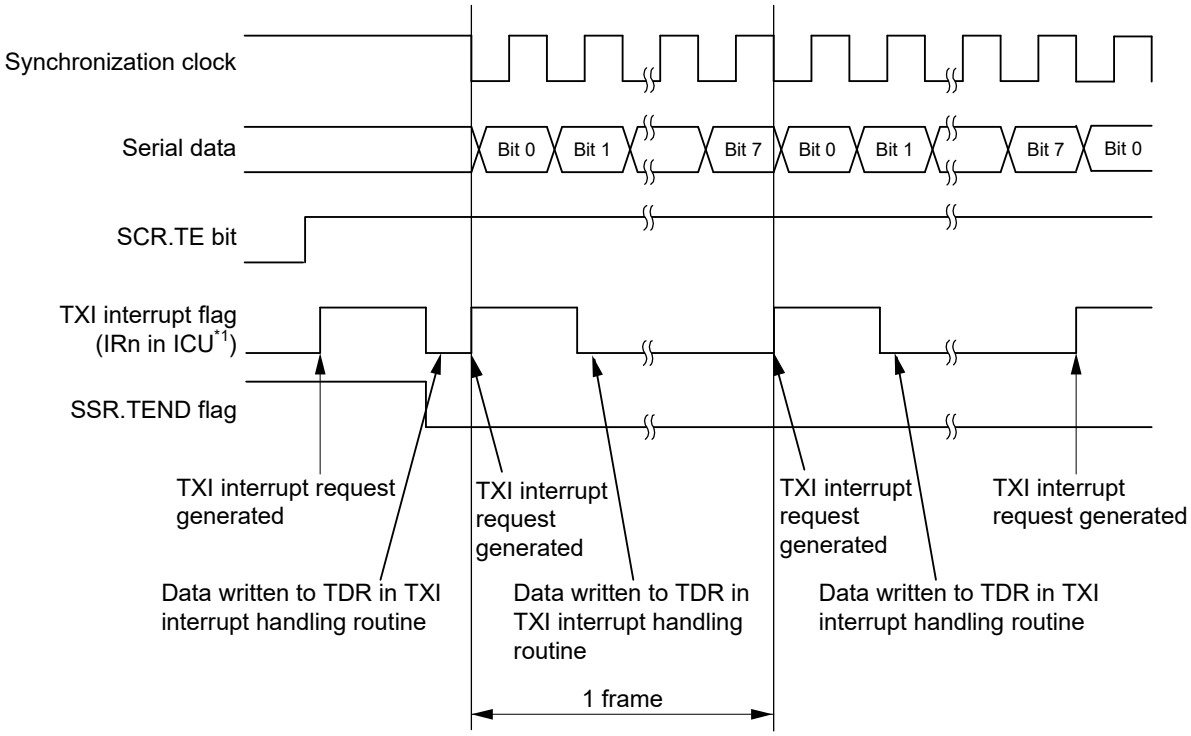
Figure 32.28 Example of SCI Initialization Flowchart (Clock Synchronous Mode)

32.5.4 Serial Data Transmission (Clock Synchronous Mode)

Figure 32.29, Figure 32.30, and Figure 32.31 show an example of the operation for serial transmission in clock synchronous mode.
In serial data transmission, the SCI operates as described below.

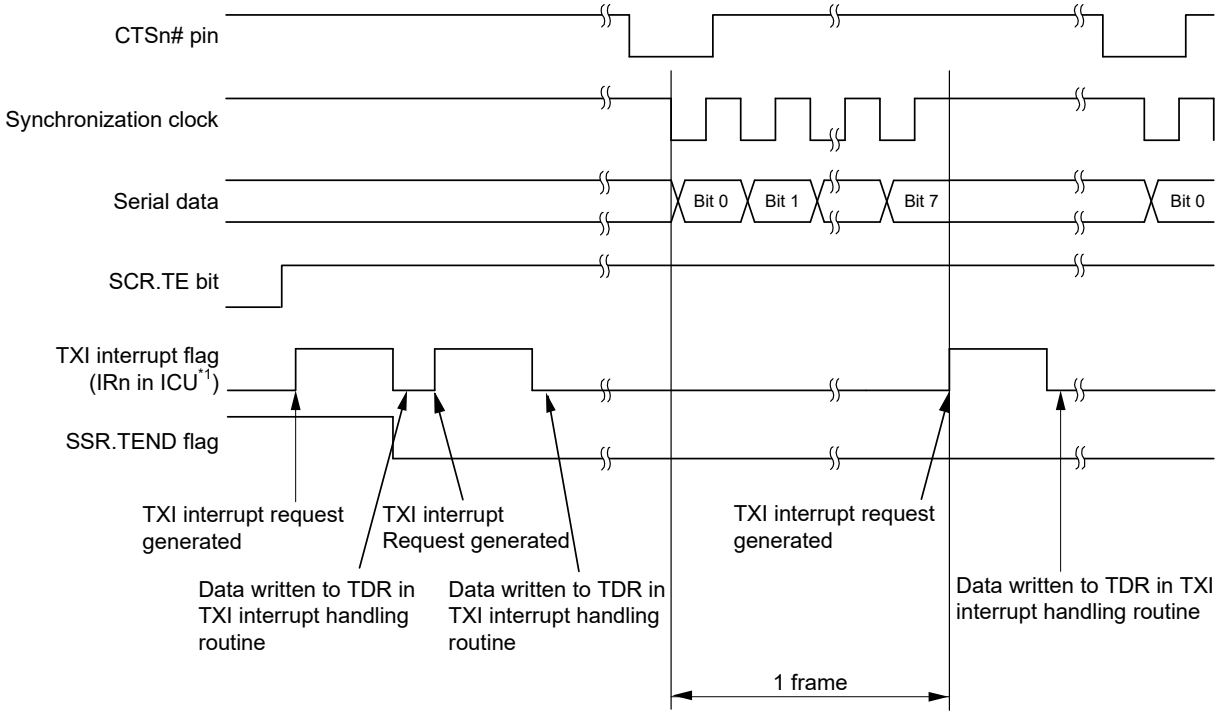
1. The SCI transfers data from the TDR register to the TSR register when data is written to the TDR register in the TXI interrupt handling routine. The TXI interrupt request at the beginning of transmission is generated when the SCR.TE bit is set to 1 after the SCR.TIE bit is set to 1 or when these 2 bits are set to 1 simultaneously by a single instruction.
2. After transferring data from the TDR register to the TSR register, the SCI starts transmission. When the SCR.TIE bit is set to 1 at this time, a TXI interrupt request is generated. Continuous transmission is enabled by writing the next transmit data to the TDR register in this TXI interrupt handling routine before transmission of the current transmit data has finished. When TEI interrupt requests are in use, set the SCR.TIE bit to 0 (a TXI interrupt request is disabled) and the SCR.TEIE bit to 1 (a TEI interrupt request is enabled) after the last of the data to be transmitted are written to the TDR register from the handling routine for TXI requests.
3. 8-bit data is sent from the TXDn pin in synchronization with the output clock when clock output mode has been specified and in synchronization with the input clock when use of an external clock has been specified. Output of the clock signal is suspended until the input CTS signal is at the low level while the SPMR.CTSE bit is 1 (CTS function is enabled).
4. The SCI checks for updating of (writing to) the TDR register at the time of the last bit output.
5. When TDR is updated, the next transmit data is transferred from the TDR register to the TSR register, and serial transmission of the next frame is started.
6. If the TDR register is not updated, set the SSR.TEND flag to 1 and the TXDn pin retains the output state of the last bit. If the SCR.TEIE bit is 1 at this time, a TEI interrupt request is generated. The SCKn pin is held high.

Figure 32.32 shows a sample flowchart of serial data transmission.
Transmission will not start while a receive error flag (ORER, FER, or PER in the SSR register) is set to 1. Be sure to set the receive error flags to 0 before starting transmission. Note that setting the SCR.RE bit to 0 does not clear the receive error flags.



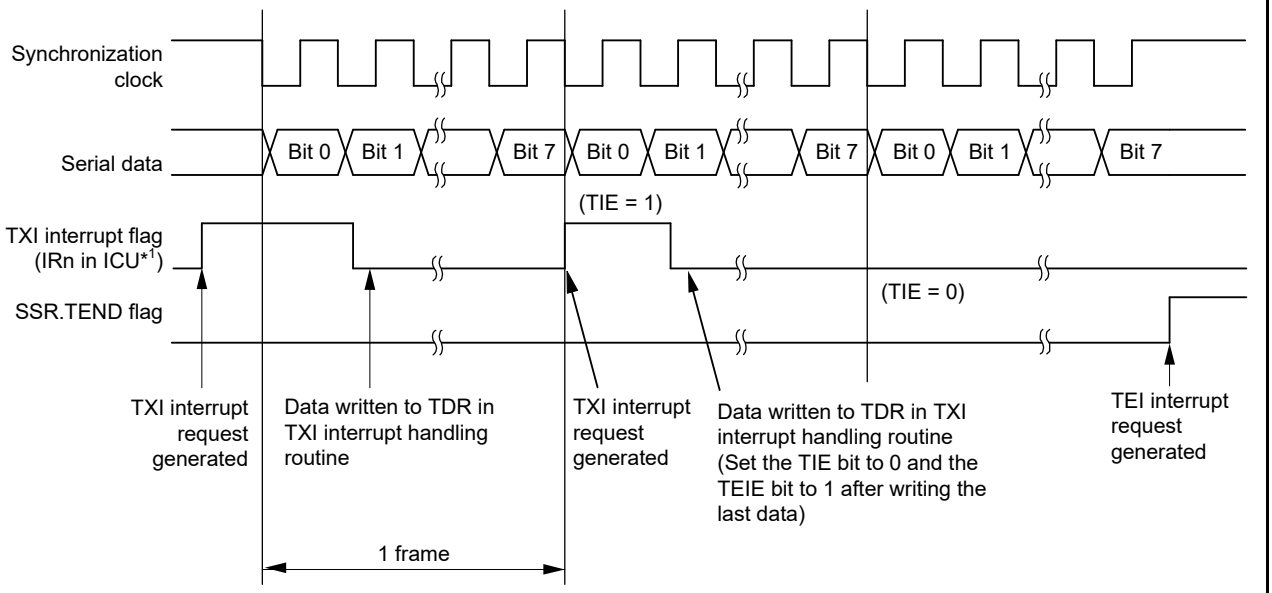
Note 1. Refer to section 14, Interrupt Controller (ICUG) for details on the corresponding interrupt vector number.

Figure 32.29 Example of Serial Data Transmission in Clock Synchronous Mode When the CTS Function is Not Used at the Beginning of Transmission



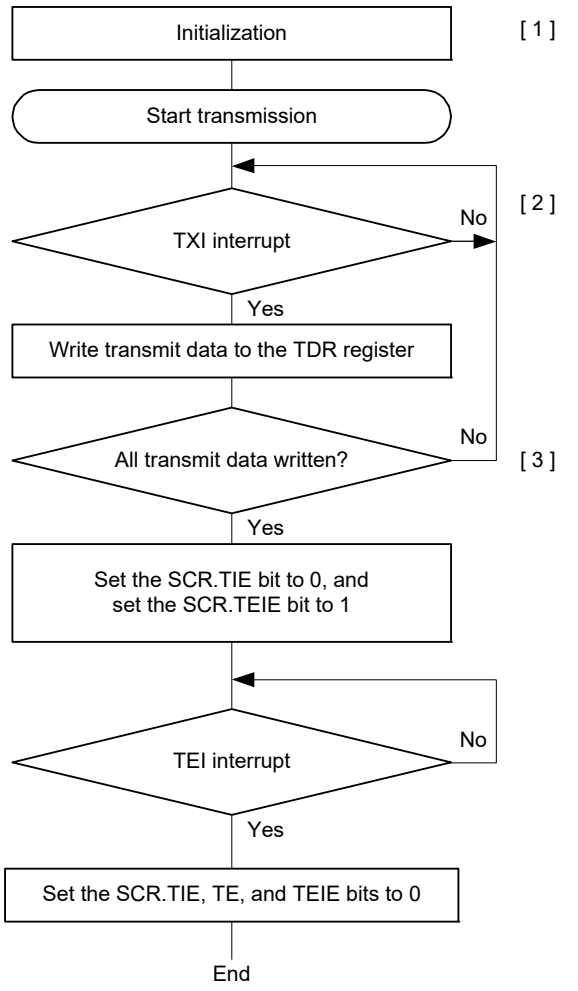
Note 1. Refer to section 14, Interrupt Controller (ICUG) for details on the corresponding interrupt vector number.

Figure 32.30 Example of Serial Data Transmission in Clock Synchronous Mode When the CTS Function is Used at the Beginning of Transmission



Note 1. Refer to section 14, Interrupt Controller (ICUG) for details on the corresponding interrupt vector number.

Figure 32.31 Example of Serial Data Transmission in Clock Synchronous Mode from the Middle of Transmission until Transmission Completion



[1] SCI initialization:
Set data transmission.

[2] Writing transmit data write to the TDR register by a TXI interrupt request:
When transmit data is transferred from the TDR register to the TSR register, a transmit data empty interrupt (TXI) request is generated.
Transmit data is written to the TDR register once from the handling routine for TXI requests.

[3] Serial transmission continuation procedure:
To continue serial transmission, write transmit data to the TDR register upon accepting a transmit data empty interrupt (TXI) request. Transmit data can also be written to the TDR register by activating the DMAC or DTC by the TXI interrupt request.
When TEI interrupt requests are in use, set the SCR.TIE bit to 0 and the SCR.TEIE bit to 1 after the last of the data to be transmitted are written to the TDR register.

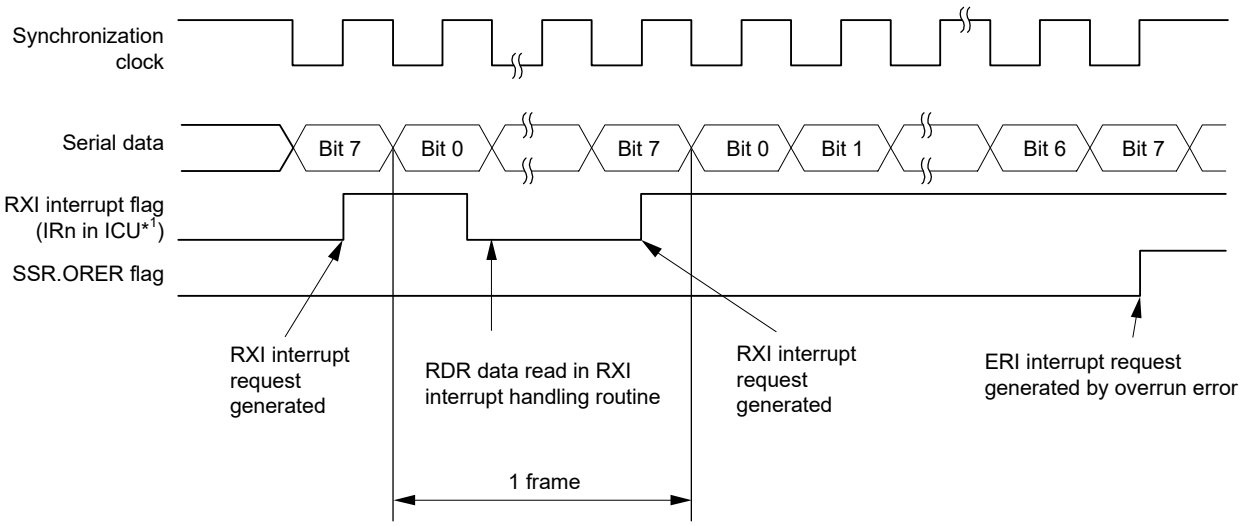
Note: When the external clock is in use (the value of the SCR.CKE[1:0] bits is 10b or 11b), the rising edge on the SCK pin for the last bit sets the SSR.TEND flag to 1. Setting the SCR.TE bit to 0 immediately after this may lead to insufficient received-data hold time on the receiver side.

Figure 32.32 Example Flowchart of Serial Transmission in Clock Synchronous Mode

32.5.5 Serial Data Reception (Clock Synchronous Mode)

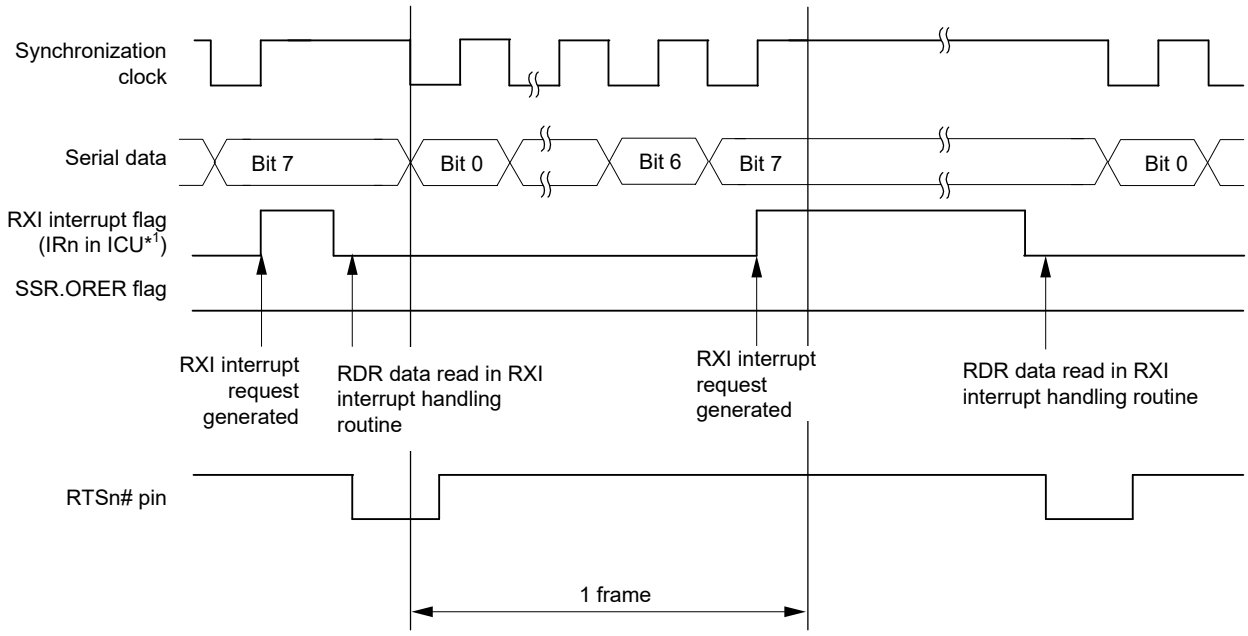
Figure 32.33 and Figure 32.34 show an example of SCI operation for serial reception in clock synchronous mode. In serial data reception, the SCI operates as described below.

1. The value of the SCR.RE bit becoming 1 places the signal output on the RTSn# pin at the low level (when the RTS function is in use).
2. The SCI performs internal initialization and starts receiving data in synchronization with a synchronization clock input or output, and stores the receive data in the RSR register.
3. If an overrun error occurs, the SSR.ORER flag is set to 1. If the SCR.RIE bit is 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to the RDR register.
4. When reception finishes successfully, receive data is transferred to the RDR register. If the SCR.RIE bit is 1 at this time, an RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to the RDR register in this RXI interrupt handling routine before reception of the next receive data is completed. Reading out the received data that have been transferred to the RDR register causes the RTSn# pin to output the low level (when the RTS function is in use).



Note 1. Refer to section 14, Interrupt Controller (ICUG) for details on the corresponding interrupt vector number.

Figure 32.33 Example of Operation for Serial Reception in Clock Synchronous Mode (1) (When RTS Function is Not Used)

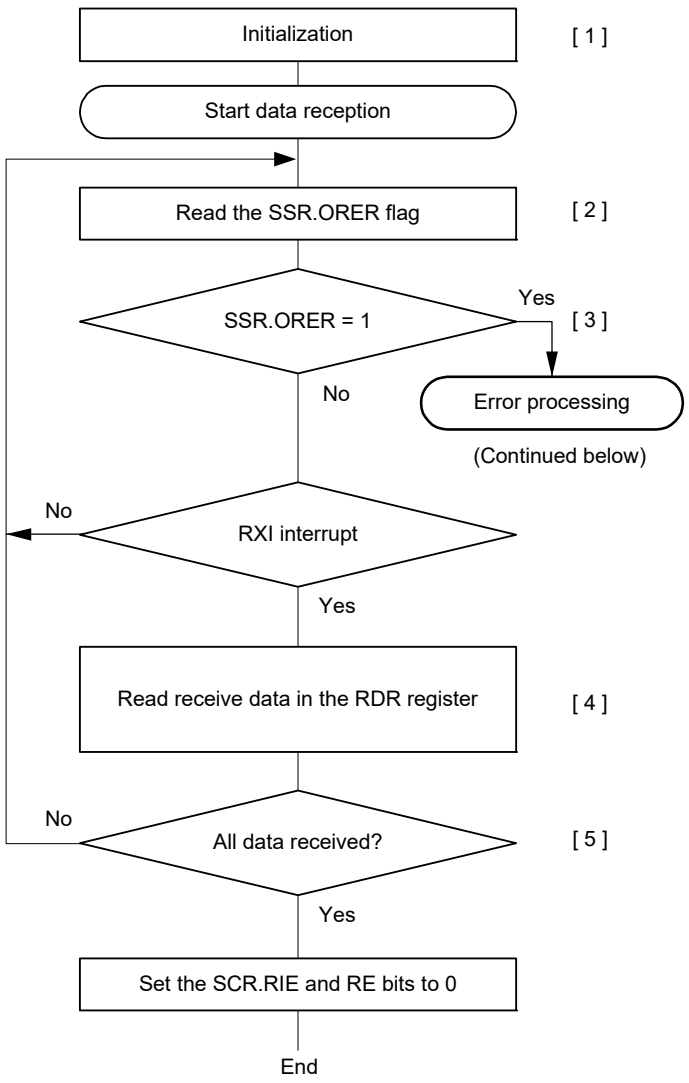


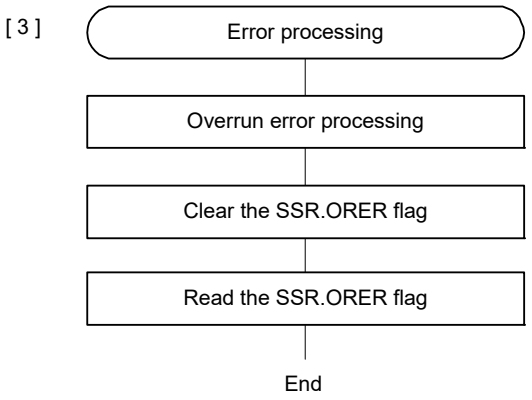
Note 1. Refer to section 14, Interrupt Controller (ICUG) for details on the corresponding interrupt vector number.

Figure 32.34 Example of Operation for Serial Reception in Clock Synchronous Mode (2) (When RTS Function is Used)

Data transfer cannot be resumed while a receive error flag is 1. Accordingly, clear the ORER, FER, and PER flags in the SSR register before resuming reception. Moreover, be sure to read the RDR register during overrun error processing. When a reception is forcibly terminated by setting the SCR.RE bit to 0 during operation, read the RDR register because received data which has not yet been read may be left in the RDR register.

Figure 32.35 shows a sample flowchart for serial data reception.





[1] SCI initialization:
Make input port-pin settings for pins to be used as RXDn pins.

[2] [3] Receive error processing:
If a receive error occurs, read the SSR.ORER flag, perform the relevant error processing, and then set the ORER flag to 0. Data reception cannot be resumed while the ORER flag is 1.

[4] Read the receive data in the RDR register once in the receive data full interrupt (RXI) request handling routine.

[5] Serial reception continuation procedure:
To continue serial reception, before the MSB (bit 7) of the current frame is received, finish reading the receive data in the RDR register. The RDR data can also be read by activating the DMAC or DTC by an RXI interrupt request.

[6] Processing in response to an overrun error:
Read the RDR register. In combination with step [7], this will make correct reception of the next frame possible.

[7] Clearing the error flag:
Write 0 to the error flag.

[8] Confirming that the error flag is cleared:
Read the error flag to confirm that its value is 0.

Figure 32.35 Example Flowchart of Serial Reception in Clock Synchronous Mode

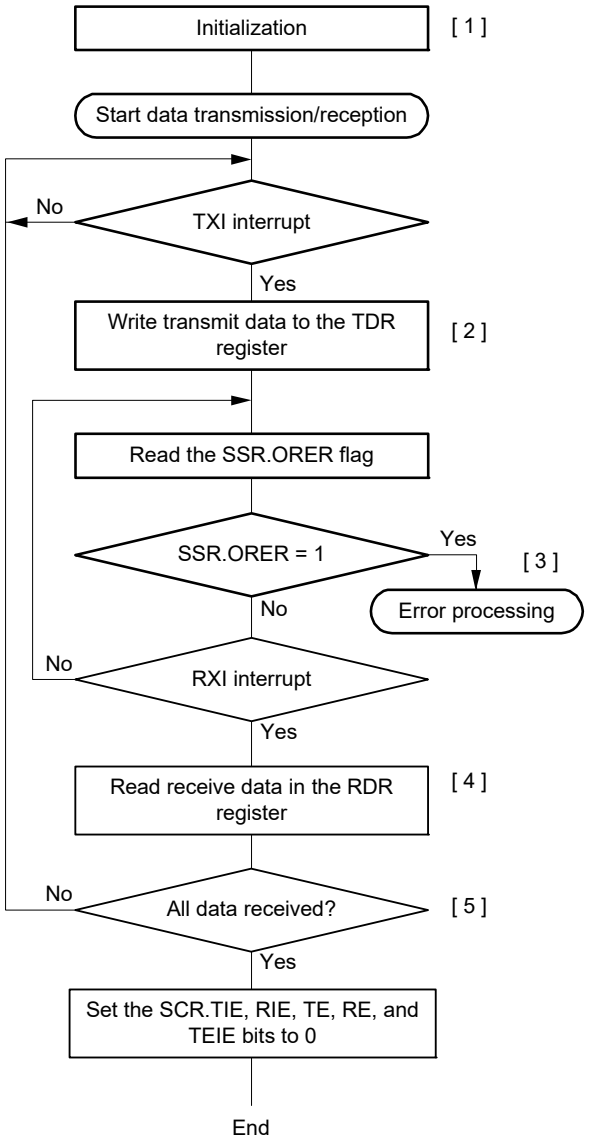
32.5.6 Simultaneous Serial Data Transmission and Reception (Clock Synchronous Mode)

Figure 32.36 shows a sample flowchart for simultaneous serial transmit and receive operations in clock synchronous mode.

After initializing the SCI, the following procedure should be used for simultaneous serial data transmit and receive operations.

To switch from transmit mode to simultaneous transmit and receive mode, check that the SCI has finished transmission by reading that the SSR.TEND flag is 1, and then initialize the SCR register. Then set the TIE, RIE, TE, and RE bits in the SCR register to 1 simultaneously by a single instruction.

To switch from receive mode to simultaneous transmit and receive mode, check that the SCI has finished reception, and then set the RIE and RE bits to 0. Then check that the receive error flags (ORER, FER, and PER in the SSR register) are 0, and then set the TIE, RIE, TE, and RE bits in the SCR register to 1 simultaneously by a single instruction.



[1] SCI initialization:
The TXDn pin can act as the output pin for transmitted data and the RXDn pin can act as the input pin for received data at the same time.

[2] Transmit data write:
Write transmit data to the TDR register once in the TXI interrupt handling routine.

[3] Receive error processing:
If a receive error occurs, read the SSR.ORER flag, perform the relevant error processing, and then set the ORER flag to 0. Data reception cannot be resumed while the ORER flag is 1.

[4] Reading receive data:
Read the receive data in the RDR register once in the RXI interrupt request handling routine.

[5] Serial transmission/reception continuation procedure:
To continue serial transmission and reception, before the MSB (bit 7) of the current frame is received, finish reading the receive data in the RDR register by the RXI interrupt. Also, before the MSB (bit 7) of the current frame is transmitted, write data to the TDR register by the TXI interrupt. Transmit data can also be written to the TDR register by activating the DMAC or DTC by a transmit data empty interrupt (TXI) request. Similarly, the RDR data can also be read by activating the DMAC or DTC by a receive data full interrupt (RXI) request.

Note: When switching from transmit or receive operation to simultaneous transmit and receive operations first set the TIE, RIE, TE, RE, and TEIE bits in the SCR register to 0, and then set TIE, RIE, TE, and RE bits to 1 simultaneously.

Figure 32.36 Example Flowchart of Simultaneous Serial Transmission and Reception in Clock Synchronous Mode

32.6 Operation in Smart Card Interface Mode

The SCI supports smart card (IC card) interfaces conforming to ISO/IEC 7816-3 (standard for Identification Cards), as an extended function of the SCI.
Smart card interface mode can be selected using the appropriate register.

32.6.1 Sample Connection

Figure 32.37 shows a sample connection between a smart card (IC card) and this MCU.
As in the figure, since this MCU communicates with an IC card using a single transmission line, interconnect the TXDn and RXDn pins and pull up the data transmission line to VCC using a resistor.
Setting the TE and RE bits in the SCR register to 1 with an IC card disconnected enables closed transmission/reception allowing self-diagnosis.
To supply an IC card with the clock pulses generated by the SCI, input the SCKn pin output to the CLK pin of an IC card.
The output port of the this MCU can be used to output a reset signal.

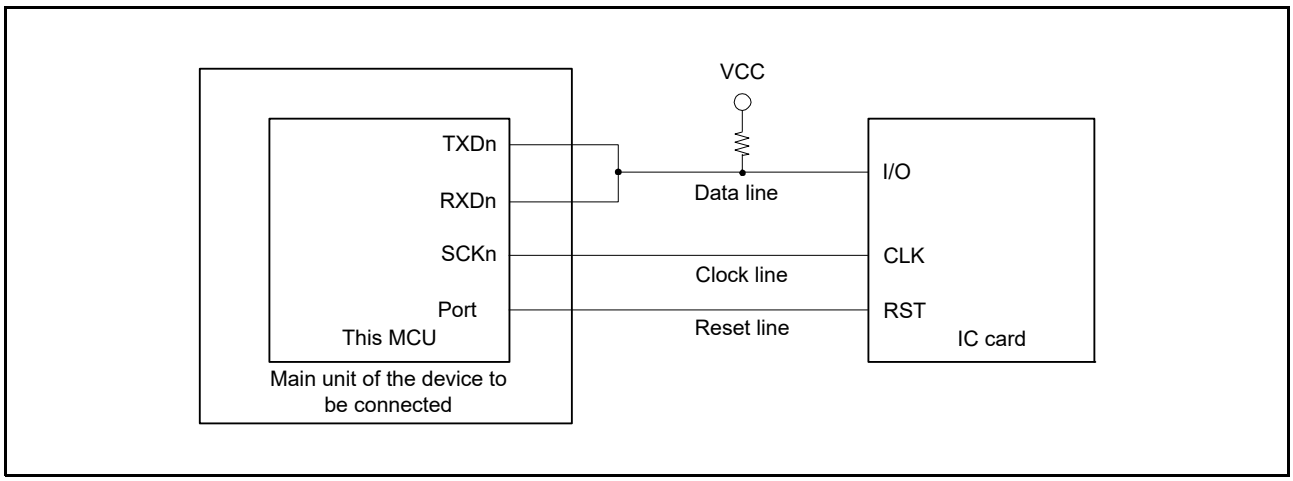


Figure 32.37 Sample Connection with a Smart Card (IC Card)

32.6.2 Data Format (Except in Block Transfer Mode)

Figure 32.38 shows the data transfer formats in smart card interface mode.

- One frame consists of 8-bit data and a parity bit in asynchronous mode.
- During transmission, at least 2 etu (elementary time unit: time required for transferring 1 bit) is secured as a guard time from the end of the parity bit until the start of the next frame.
- If a parity error is detected during reception, a low-level error signal is output for 1 etu after 10.5 etu has passed from the start bit.
- If an error signal is sampled during transmission, the same data is automatically retransmitted after at least 2 etu.

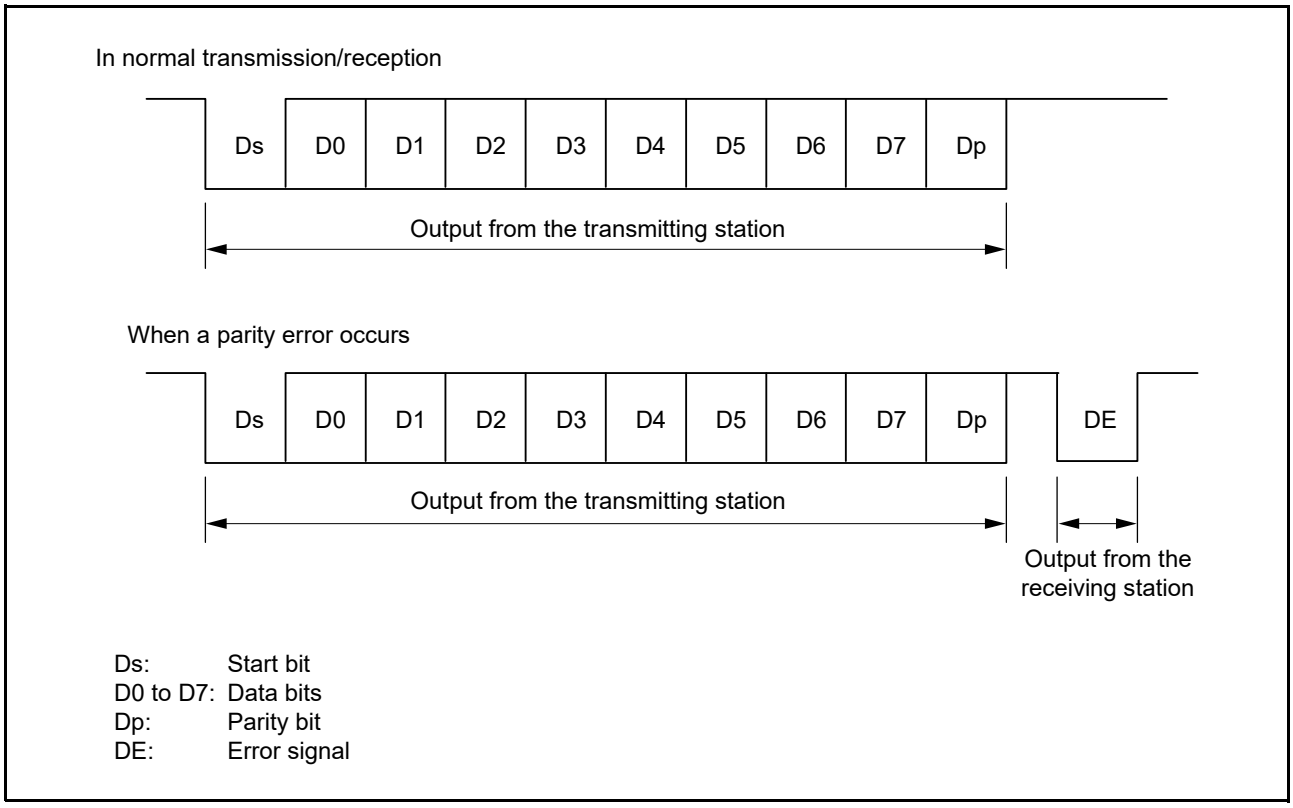


Figure 32.38 Data Formats in Smart Card Interface Mode

For communications with IC cards of the direct convention type and inverse convention type, follow the procedure below.

(1) Direct Convention Type

For the direct convention type, logic levels 1 and 0 correspond to states Z and A, respectively, and data is transferred with LSB first as the start character, as shown in Figure 32.39. Therefore, data in the start character in the figure is 3Bh. When using the direct convention type, write 0 to both the SDIR and SINV bits in the SCMR register. Write 0 to the SMR.PM bit in order to use even parity, which is prescribed by the smart card standard.

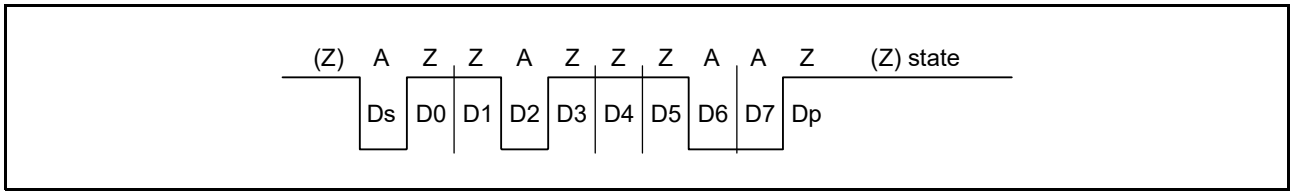


Figure 32.39 Direct Convention (SCMR.SDIR bit = 0, SCMR.SINV bit = 0, SMR.PM bit = 0)

(2) Inverse Convention Type

For the inverse convention type, logic levels 1 and 0 correspond to states A and Z, respectively and data is transferred with MSB first as the start character, as shown in Figure 32.40. Therefore, data in the start character in the figure is 3Fh. When using the inverse convention type, write 1 to both the SDIR and SINV bits in the SCMR register. The parity bit is logic level 0 to produce even parity, which is prescribed by the smart card standard, and corresponds to state Z. Since the SINV bit of the this MCU only inverts data bits D7 to D0, write 1 to the SMR.PM bit to invert the parity bit for both transmission and reception.

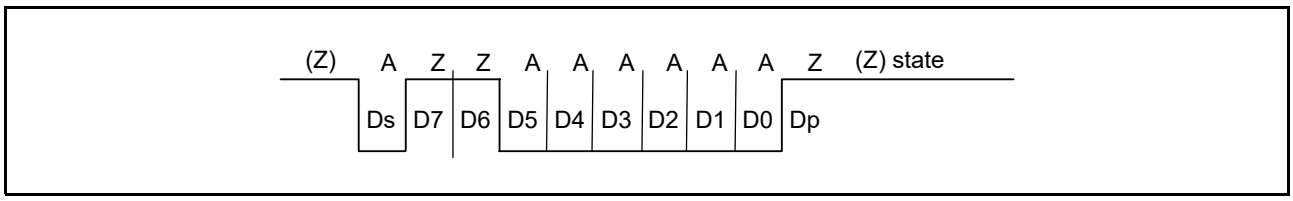


Figure 32.40 Inverse Convention (SCMR.SDIR bit = 1, SCMR.SINV bit = 1, SMR.PM bit = 1)

32.6.3 Block Transfer Mode

Block transfer mode is different from non-block transfer mode in the following respects.

- Even if a parity error is detected during reception, no error signal is output. Since the SSR.PER flag is set by error detection, clear the PER flag before receiving the parity bit of the next frame.
- During transmission, at least 1 etu is secured as a guard time from the end of the parity bit until the start of the next frame.
- Since the same data is not retransmitted during transmission, the SSR.TEND flag is set 11.5 etu after transmission start.
- In block transfer mode, the SSR.ERS flag indicates the error signal status as in non-block transfer mode, but the flag is read as 0 because no error signal is transferred.

32.6.4 Receive Data Sampling Timing and Reception Margin

Only the base clock generated by the on-chip baud rate generator can be used as a transmit/receive clock in smart card interface mode.

In this mode, the SCI can operate on a base clock with a frequency of 32, 64, 372, 256, 93, 128, 186, or 512 times the bit rate according to the settings of the SCMR.BCP2 bit and the SMR.BCP[1:0] bits.

For data reception, the falling edge of the start bit is sampled with the base clock to perform synchronization. Receive data is sampled on the 16th, 32nd, 186th, 128th, 46th, 64th, 93rd, and 256th rising edges of the base clock so that it can be latched at the middle of each bit as shown in Figure 32.41. The reception margin here is determined by the following formula.

$$M = \left| \left(0.5 - \frac{1}{2N}\right) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\ (\%)$$

M: Reception margin (%)
N: Ratio of bit rate to clock (N = 32, 64, 372, 256)
D: Duty cycle of clock (D = 0 to 1.0)
L: Frame length (L = 10)
F: Absolute value of clock frequency deviation

Assuming values of F = 0, D = 0.5, and N = 372 in the above formula, the reception margin is determined by the formula below.

M = {0.5 – 1/(2 × 372)} × 100 (%) = 49.866 (%)

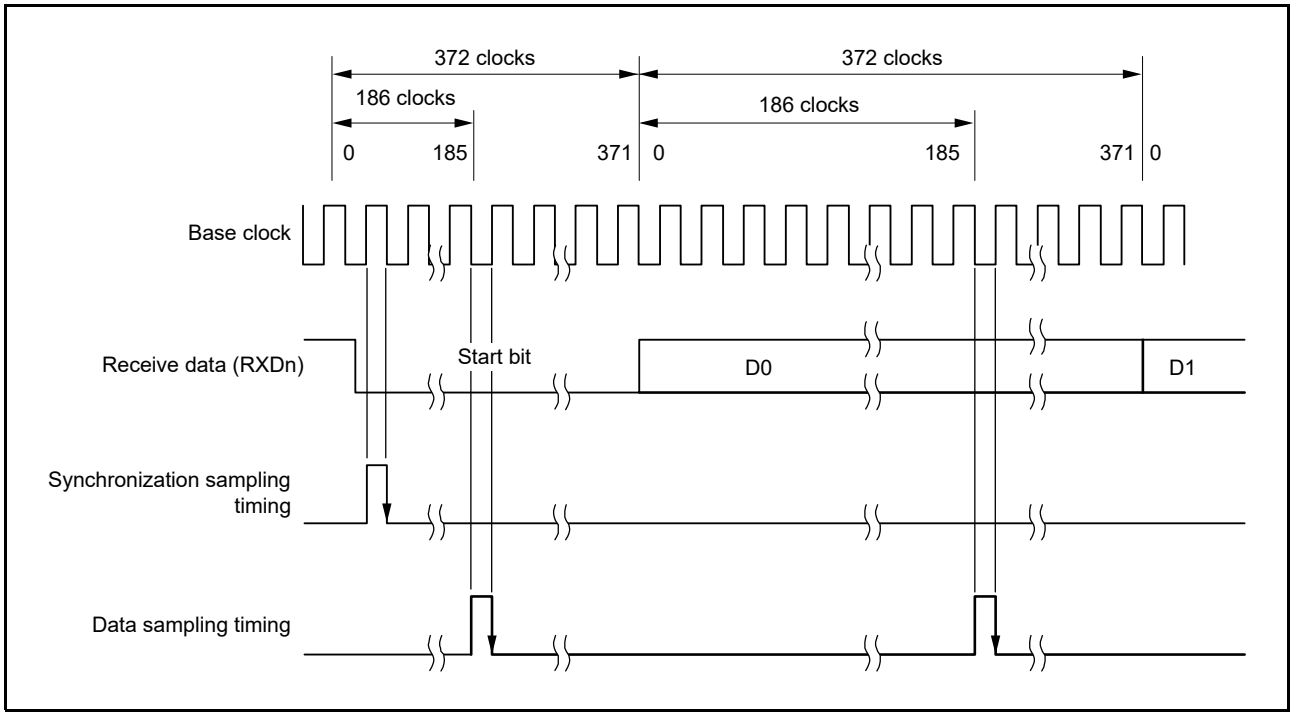
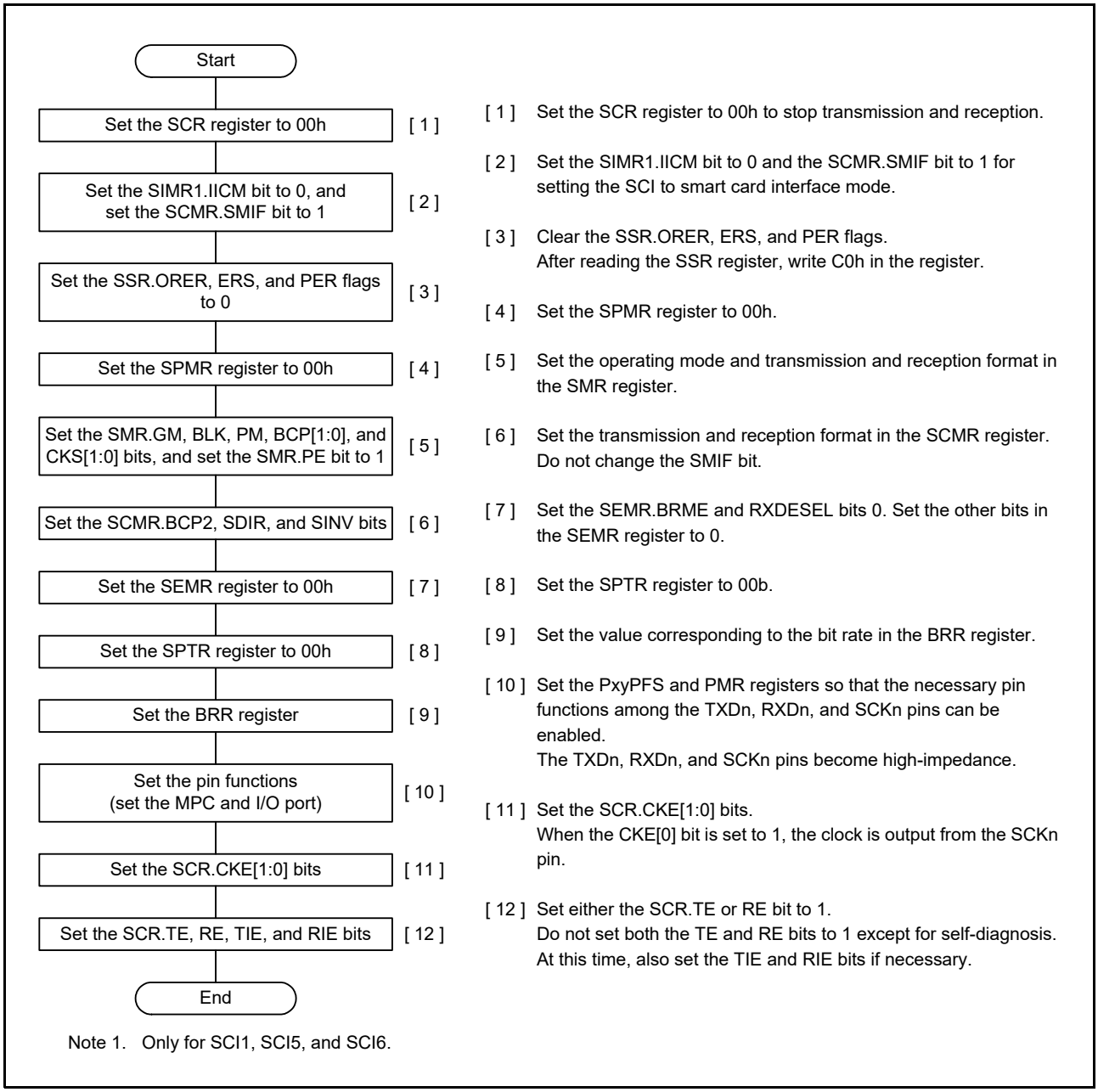


Figure 32.41 Receive Data Sampling Timing in Smart Card Interface Mode (When Clock Frequency is 372 Times the Bit Rate)

32.6.5 SCI Initialization (Smart Card Interface Mode)

Initialize the SCI following the example of flowchart shown in Figure 32.42.
Initialize the SCR and SSR registers before switching from transmit mode to receive mode and vice versa. When not changing the bit rate, it is not necessary to set the CKE[1:0] bits to 00b. Even if the RE bit is set to 0, the RDR register is not initialized.

To change receive mode to transmit mode, first check that reception has completed, and then execute steps [1] and [3] in Figure 32.42. Set TE = 1 and RE = 0 at step [11]. Reception completion can be verified by reading the RXI request, SSR.ORER, or SSR.PER flag.
To change transmit mode to receive mode, first check that transmission has completed, and then execute steps [1] and [3] in Figure 32.42. Set TE = 0 and RE = 1 at step [11]. Transmission completion can be verified by reading the SSR.TEND flag.



Note 1. Only for SCI1, SCI5, and SCI6.

Figure 32.42 Example of SCI Initialization Flowchart (Smart Card Interface Mode)

Figure 32.43 shows an example of data transmission when the SCI is set to smart card interface mode according to the flow described in Figure 32.42 after a reset. When the pin functions are set to the SCK and TXD pins, they are still high-impedance because the SCR.CKE[0] and SCR.TE bits are 0. When the CKE[0] bit is set to 1, clock is output from the SCK pin. When the transmit data is written after setting the TE bit to 1, a data transmission starts. After the TE bit is set to 1, one frame of high-impedance is output from TXD pin (internal wait time) and then the data transmission starts. In smart card interface mode, the clock is continuously output while the CKE[0] bit is set to 1 (clock output) even if both the TE and RE bits are set to 0.

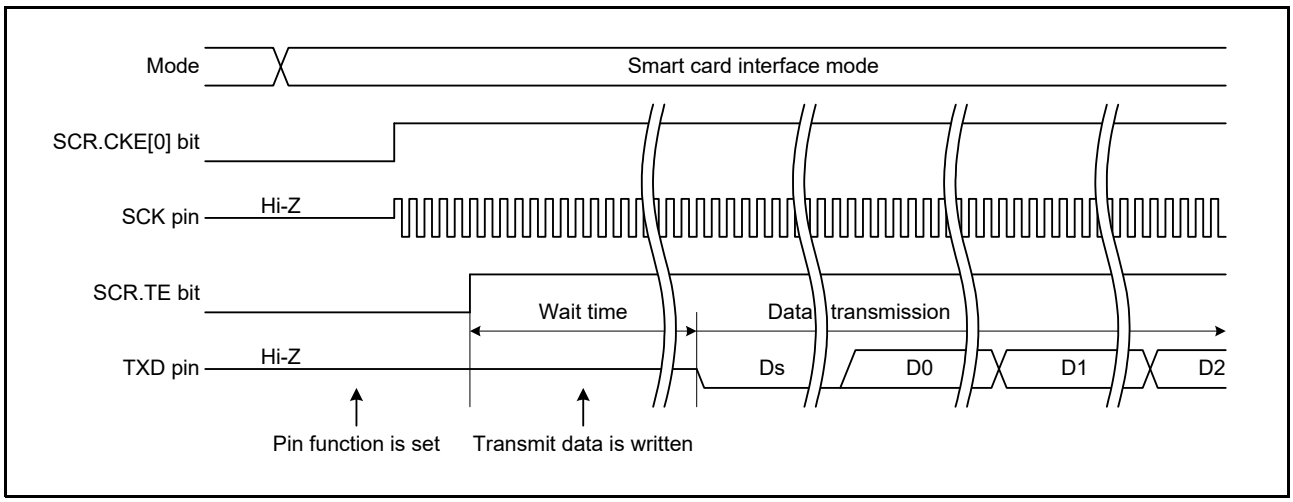


Figure 32.43 Example of Data Transmission Timing in Smart Card Interface Mode

32.6.6 Serial Data Transmission (Except in Block Transfer Mode)

Serial data transmission in smart card interface mode (except in block transfer mode), in that an error signal is sampled and data can be retransmitted, is different from that in non-smart card interface mode. Figure 32.44 shows the data retransmit operation during transmission.

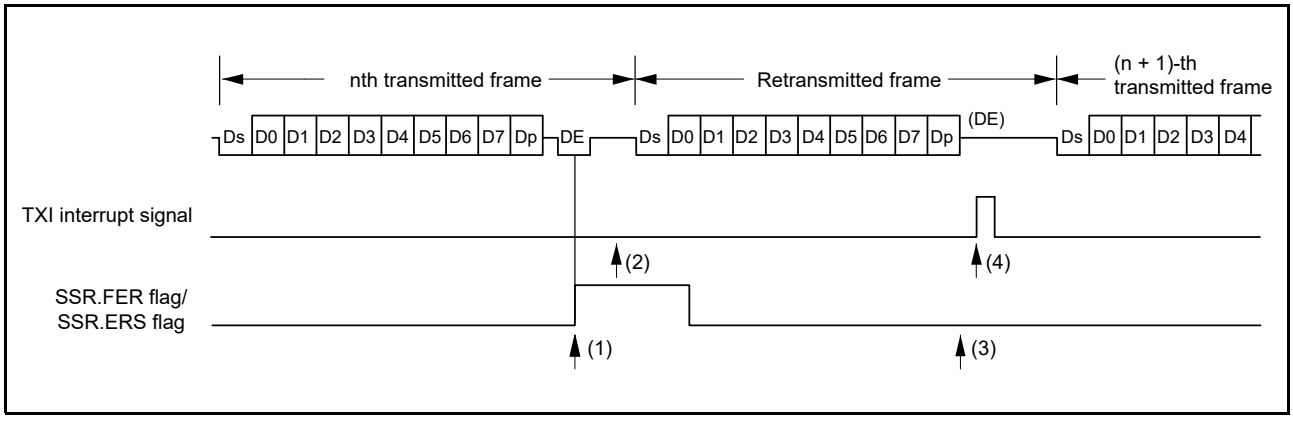


Figure 32.44 Data Retransmit Operation in SCI Transmit Mode

(1) When an error signal from the receiver end is sampled after one-frame data has been transmitted, the SSR.ERS flag is set to 1. If the SCR.RIE bit is 1 at this time, an ERI interrupt request is generated. Clear the ERS flag before the next parity bit is sampled.

(2) For a frame in which an error signal is received, the SSR.TEND flag is not set. Data is retransferred from the TDR register to the TSR register allowing automatic data retransmission.

(3) If no error signal is returned from the receiver, the ERS flag is not set to 1.

(4) In this case, the SCI judges that transmission of one-frame data (including retransmission) has been completed, and the TEND flag is set. If the SCR.TIE bit is 1 at this time, a TXI interrupt request is generated. Writing transmit data to the TDR register starts transmission of the next data.

Figure 32.45 shows a sample flowchart of serial transmission.

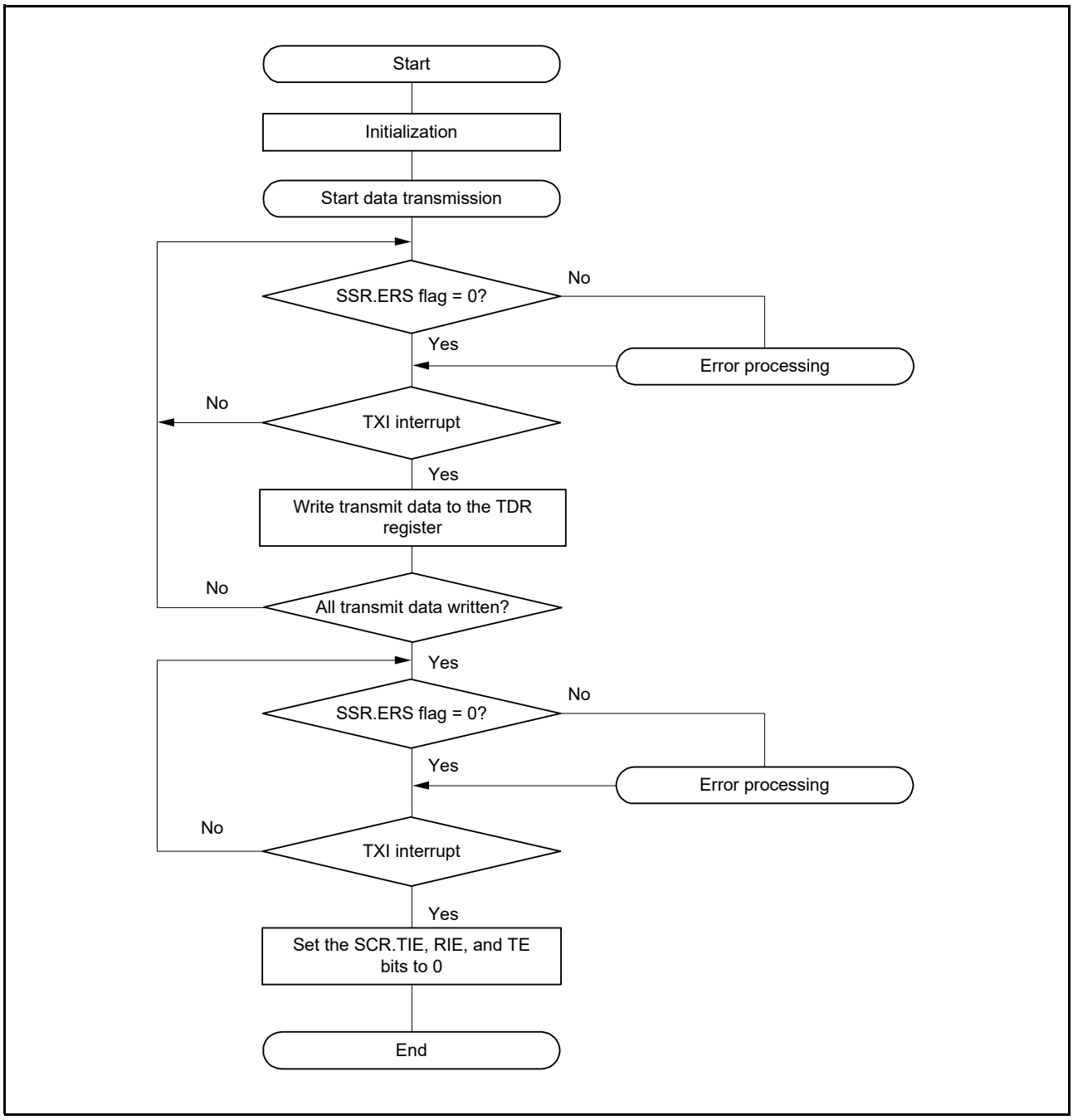


Figure 32.45 Sample Smart Card Interface Transmission Flowchart

All the processing steps are automatically performed using a TXI interrupt request to activate the DTC or DMAC. When the SSR.TEND flag is set to 1 in transmission, if the SCR.TIE bit is 1, a TXI interrupt request is generated. The DTC or DMAC is activated by a TXI interrupt request if the TXI interrupt request is specified as a source of DTC or DMAC activation beforehand, allowing transfer of transmit data. The TEND flag is automatically set to 0 when the DTC or DMAC transfers the data.

If an error occurs, the SCI automatically retransmits the same data. During this retransmission, the TEND flag is kept to 0 and the DTC or DMAC is not activated. Therefore, the SCI and DTC or DMAC automatically transmit the specified number of bytes, including retransmission in the case of error occurrence. However, since the ERS flag is not automatically cleared, set the RIE bit to 1 beforehand to enable an ERI interrupt request to be generated at error occurrence, and clear the ERS flag.

When transmitting/receiving data using the DTC or DMAC, be sure to make settings to enable the DTC or DMAC before making SCI settings.
For DTC or DMAC settings, refer to section 18, Data Transfer Controller (DTCb), section 17, DMA Controller (DMACAa).

Note that the SSR.TEND flag is set in different timings depending on the SMR.GM bit setting. Figure 32.46 shows the TEND flag generation timing.

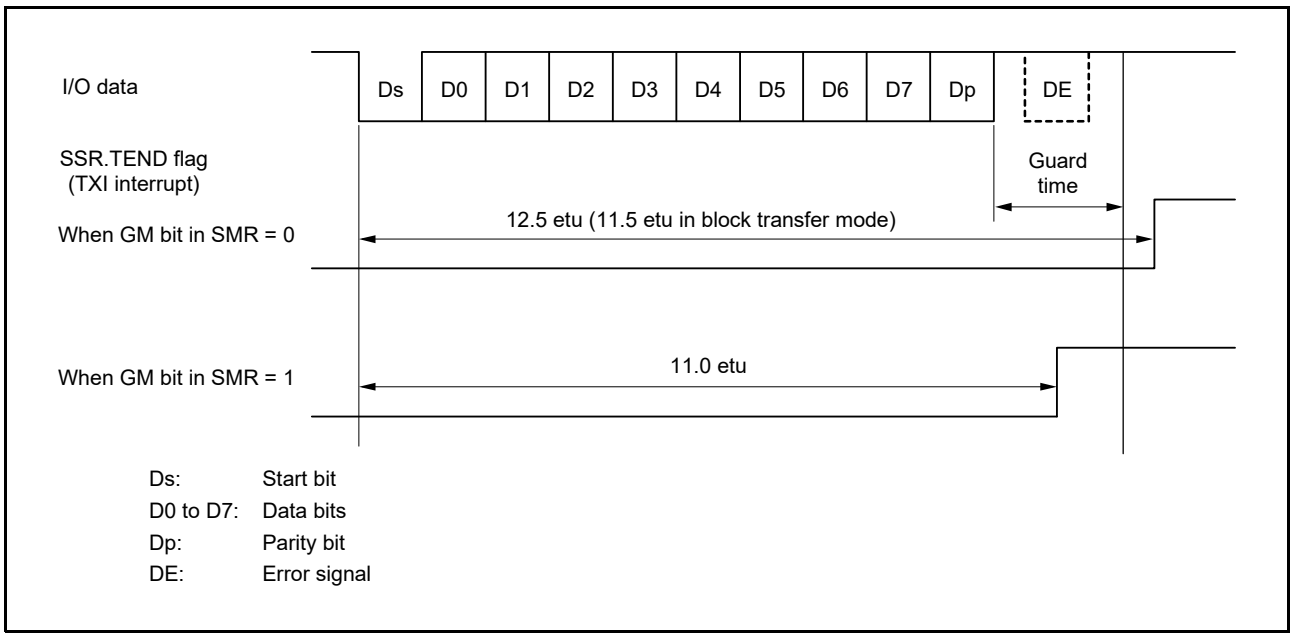


Figure 32.46 SSR.TEND Flag Generation Timing during Transmission

32.6.7 Serial Data Reception (Except in Block Transfer Mode)

Serial data reception in smart card interface mode is similar to that in non-smart card interface mode. Figure 32.47 shows the data retransmit operation in receive mode.

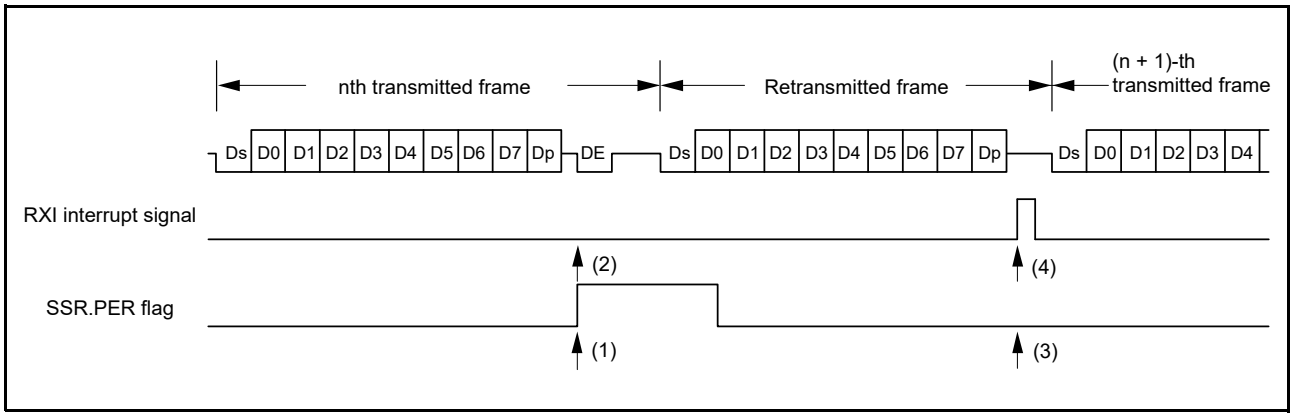


Figure 32.47 Data Retransmit Operation in SCI Receive Mode (Data Retransmit Operation during Reception)

(1) If a parity error is detected in receive data, the SSR.PER flag is set to 1. When the SCR.RIE bit is 1 at this time, an ERI interrupt request is generated. Clear the PER flag before the next parity bit is sampled.
(2) For a frame in which a parity error is detected, no RXI interrupt is generated.
(3) When no parity error is detected, the SSR.PER flag is not set to 1.
(4) In this case, data is determined to have been received successfully. When the SCR.RIE bit is 1, an RXI interrupt request is generated.

Figure 32.48 shows a sample flowchart for serial data reception.

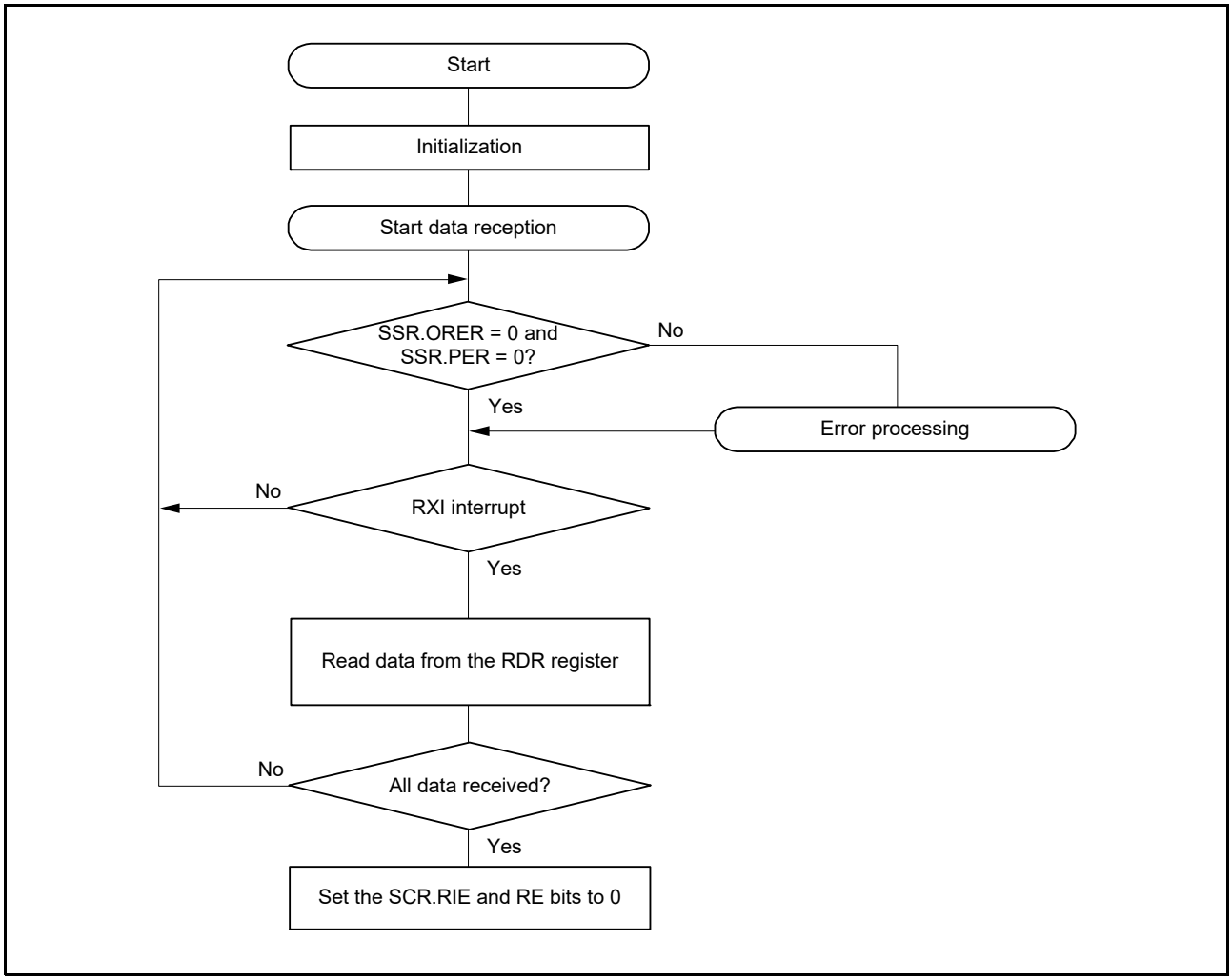


Figure 32.48 Sample Smart Card Interface Reception Flowchart

All the processing steps are automatically performed using an RXI interrupt request to activate the DTC or DMAC.
In reception, setting the RIE bit to 1 allows an RXI interrupt request to be generated. The DTC or DMAC is activated by an RXI interrupt request if the RXI interrupt request is specified as a source of DTC or DMAC activation beforehand, allowing transfer of receive data.
If an error occurs during reception and either the ORER or PER flag in the SSR register is set to 1, a receive error interrupt (ERI) request is generated. Clear the error flag after the error occurrence. If an error occurs, the DTC or DMAC is not activated and receive data is skipped. Therefore, the number of bytes of receive data specified in the DTC or DMAC is transferred.
Even if a parity error occurs and the PER flag is set to 1 during reception, receive data is transferred to the RDR register, thus allowing the data to be read.
When a reception is forcibly terminated by setting the SCR.RE bit to 0 during operation, read the RDR register because the received data which has not yet been read may be left in the RDR register.

Note 1. For operations in block transfer mode, refer to section 32.3, Operation in Asynchronous Mode.

32.6.8 Clock Output Control

Clock output can be fixed to high or low using the SCR.CKE[1:0] bits when the SMR.GM bit is 1. When the CKE[1:0] bits are set to 01b (clock output), the base clock is output from the SCK pin. For the settings of the base clock frequency (bit rate), refer to section 32.2.11, Bit Rate Register (BRR). When the CKE[1:0] bits are set to 00b (output fixed low) or 10b (output fixed to high), the SCK pin can be fixed to low or high.

Figure 32.49 shows a timing chart when the clock output is controlled.

If changing the CKE[1:0] bits while the SMR.GM bit is 0 (non-GSM mode), a pulse of unexpected width may output from SCK pin because the result is immediately reflected to the SCK pin.

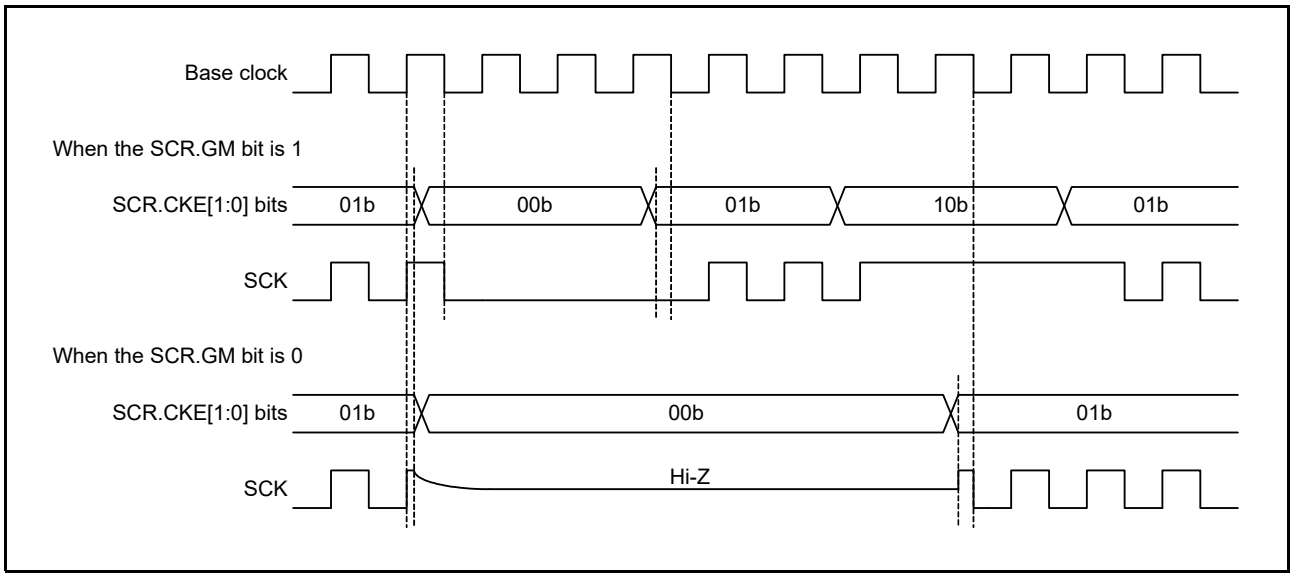


Figure 32.49 Clock Output Control

32.7 Operation in Simple I2C Mode

Simple I2C-bus format is composed of 8 data bits and an acknowledge bit. By continuing into a slave-address frame after a start condition or restart condition, a master device is able to specify a slave device as the partner for communications. The currently specified slave device remains valid until a new slave device is specified or a stop condition is satisfied.
The 8 data bits in all frames are transmitted in order from the MSB.
The I2C-bus format and timing of the I2C-bus are shown in Figure 32.50 and Figure 32.51.

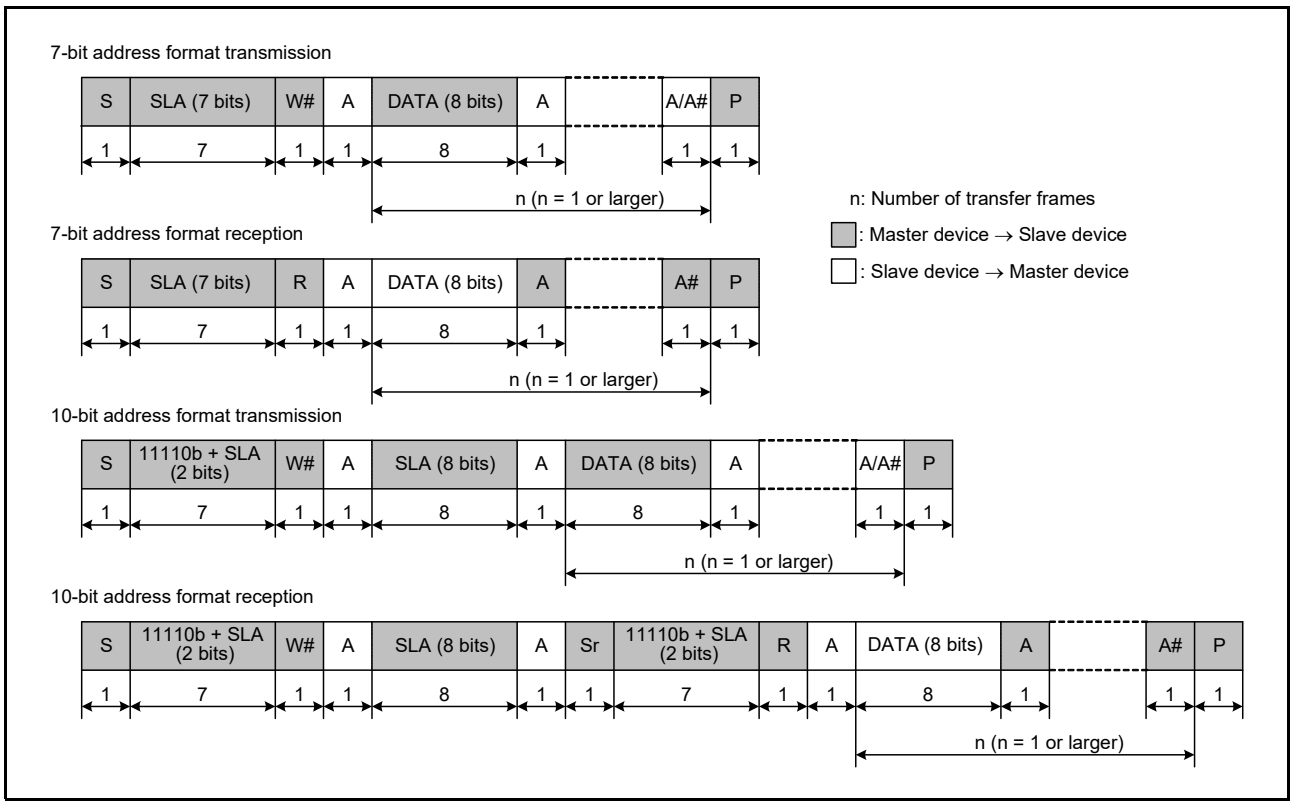


Figure 32.50 I2C-bus Format

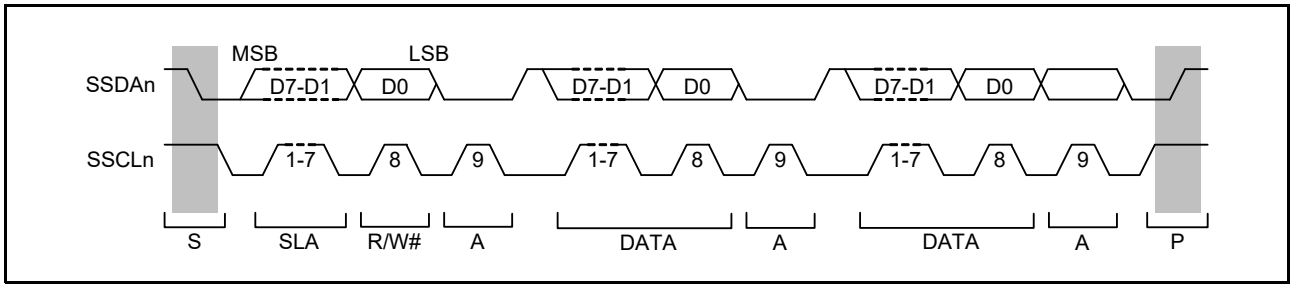


Figure 32.51 I2C-bus Timing (When SLA is 7 Bits)

S: Indicates a start condition, i.e. the master device changing the level on the SSDAn line from the high to the low level while the SSCLn line is at the high level.
SLA: Indicates a slave address, by which the master device selects a slave device.
R/W#: Indicates the direction of transfer (reception or transmission). The value 1 corresponds to transfer from the slave device to the master device and 0 corresponds to transfer from the master device to the slave device.
A/A#: Indicates an acknowledge bit. This is returned by the slave device for master transmission and by the master device for master reception. Return of the low level indicates ACK and return of the high level indicates NACK.
Sr: Indicates a restart condition, i.e. the master device changing the level on the SSDAn line from the high to the low level while the SSCLn line is at the high level and after the setup time has elapsed.
DATA: Indicates the data being received or transmitted.
P: Indicates a stop condition, i.e. the master device changing the level on the SSDAn line from the low to the high level while the SSCLn line is at the high level.

32.7.1 Generation of Start, Restart, and Stop Conditions

Writing 1 to the SIMR3.IICSTAREQ bit causes the generation of a start condition. The generation of a start condition proceeds through the following operations.

- The level on the SSDAn line falls (from the high level to the low level) and the SSCLn line is kept in the released state.
- The hold time for the start condition is secured as half of a bit period at the bit rate determined by the setting of the BRR.
- The level on the SSCLn line falls (from the high level to the low level), the SIMR3.IICSTAREQ bit is set (to 0), and a start-condition generated interrupt is output.

Writing 1 to the SIMR3.IICRSTAREQ bit causes the generation of a start condition. The generation of a start condition proceeds through the following operations.

- The SSDAn line is released and the SSCLn line is kept at the low level.
- The period at low level for the SSCLn line is secured as half of a bit period at the bit rate determined by the setting of the BRR.
- The SSCLn line is released (transition from the low to the high level).
- Once the high level on the SSCLn line is detected, the setup time for the restart condition is secured as half of a bit period at the bit rate determined by the setting of the BRR.
- The level on the SSDAn line falls (from the high level to the low level).
- The hold time for the restart condition is secured as half of a bit period at the bit rate determined by the setting of the BRR.
- The level on the SSCLn line falls (from the high level to the low level), the SIMR3.IICRSTAREQ bit is set (to 0), and a restart-condition generated interrupt is output.

Writing 1 to the SIMR3.IICSTPREQ bit causes the generation of a stop condition. The generation of a stop condition proceeds through the following operations.

- The level on the SSDAn line falls (from the high level to the low level) and the SSCLn line is kept at the low level.
- The period at low level for the SSCLn line is secured as half of a bit period at the bit rate determined by the setting of the BRR.
- The SSCLn line is released (transition from the low to the high level).
- Once the high level on the SSCLn line is detected, the setup time for the stop condition is secured as half of a bit period at the bit rate determined by the setting of the BRR.
- The SSDAn is released (transition from the low to the high level), the SIMR3.IICSTPREQ bit is set (to 0), and a stop-condition generated interrupt is output.

Figure 32.52 shows the timing of operations in the generation of start, restart, and stop conditions.

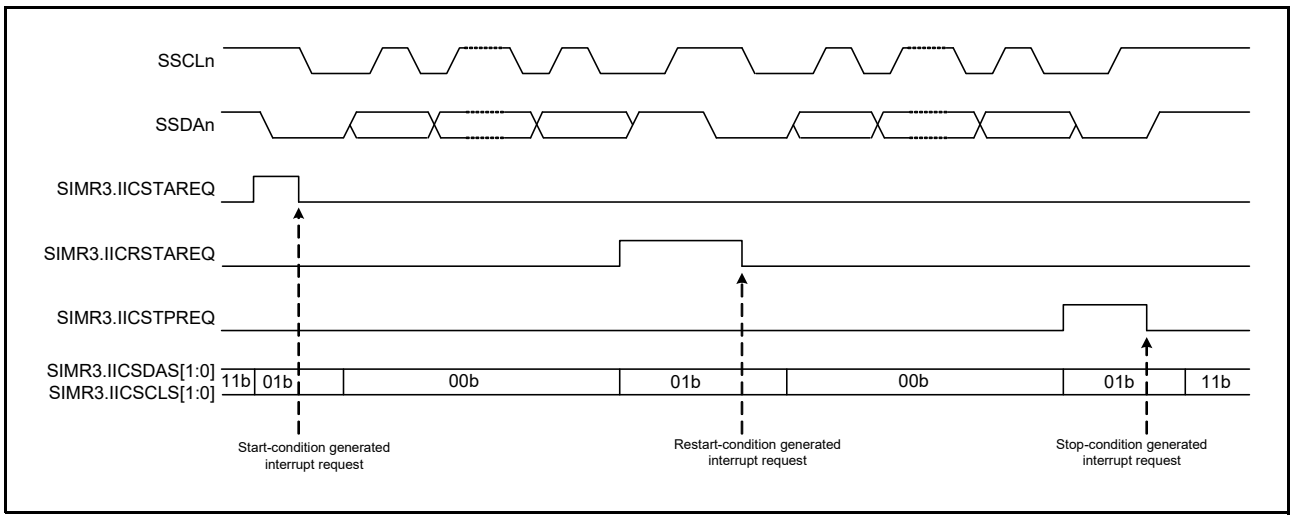


Figure 32.52 Timing of Operations in the Generation of Start, Restart, and Stop Conditions

32.7.2 Clock Synchronization

The SSCLn line may be placed at the low level in the case of a wait inserted by a slave device as the other side of transfer. Setting the SIMR2.IICCSC bit to 1 applies control to obtain synchronization when the levels of the internal SSCLn clock signal and the level being input on the SSCLn pin differ.

When the SIMR2.IICCSC bit is set to 1, the level of the internal SSCLn clock signal changes from low to high, counting to determine the period at high level is stopped while the low level is being input on the SSCLn pin, and counting to determine the period at high level starts after the transition of the input on the SSCLn pin to the high level. The interval from this time until counting to determine the period at high level starts on the transition of the SSCLn pin to the high level is the total of the delay of SSCLn output, delay for noise filtering of the input on the SSCLn pin (2 or 3 cycles of sampling clock for the noise filter), and delay for internal processing (1 or 2 cycles of PCLK). The period at high level of the internal SSCLn clock is extended even if other devices are not placing the low level on the SSCLn line. If the SIMR2.IICCSC bit is 1, synchronization is obtained for the transmission and reception of data by taking the logical AND of the input on the SSCLn pin and the internal SSCLn clock. If the SIMR2.IICCSC bit is 0, synchronization with the internal SSCLn clock is obtained for the transmission and reception of data.

If a slave device inserts a period of waiting into the interval until the transition of the internal SSCLn clock signal from the low to the high level after a request for the generation of a start, restart, or stop condition is issued, the time until generation is prolonged by that period.

If a slave device inserts a period of waiting after the transition of the internal SSCLn clock signal from the low to the high level, although the generation-completed interrupt is issued without stopping the period of waiting, generation of the condition itself is not guaranteed. Figure 32.53 shows an example of operations to synchronize the clocks.

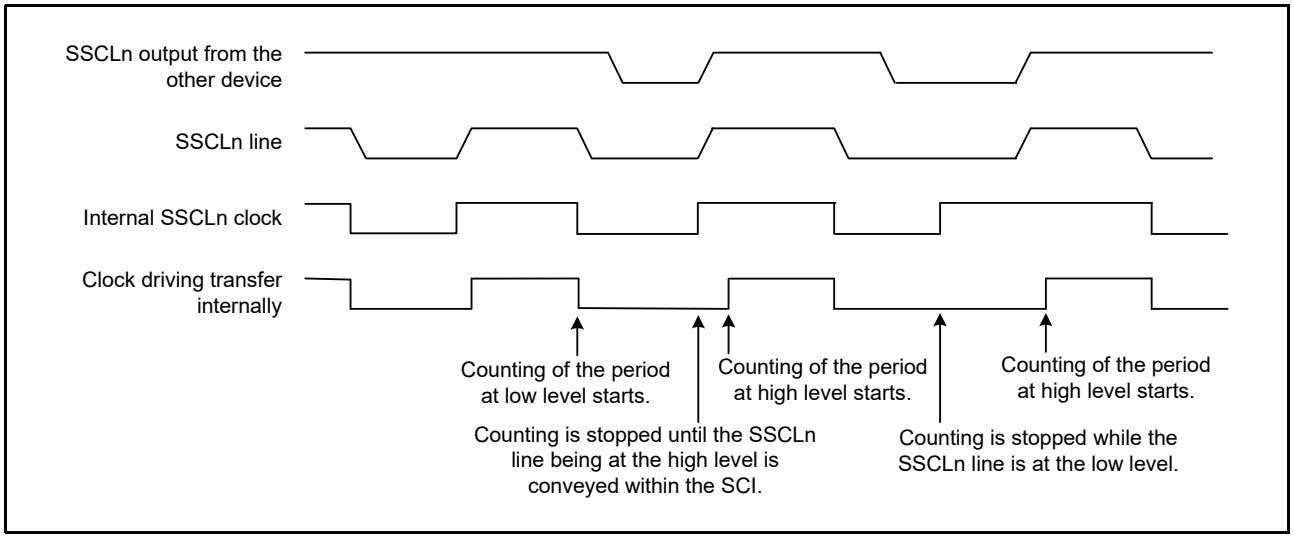


Figure 32.53 Example of Operations for Clock Synchronization

32.7.3 SSDA Output Delay

The SIMR1.IICDL[4:0] bits can be used to set a delay for output on the SSDAn pin relative to falling edges of output on the SSCLn pin. Delay-time settings from 0 to 31 are selectable, representing periods of the corresponding numbers of cycles of the clock signal from the on-chip baud rate generator (derived by frequency-dividing the base clock, PCLK, by the divisor selected by the SMR.CKS[1:0] bits). A delay for output on the SSDAn pin is for the start condition/restart condition/stop condition signal, 8-bit transmit data, and an acknowledge bit.

If the SSDA output delay is shorter than the time for the level on the SSCLn pin to fall, the change of the output on the SSDAn pin will start while the output level on the SSCLn pin is falling, creating a possibility of erroneous operation for slave devices. Ensure that settings for the delay of output on the SSDAn pin are for times greater than the time output on the SSCLn pin takes to fall (300 ns for I^2C-bus in normal mode and fast mode).

Figure 32.54 shows the timing of delays in SSDA output.

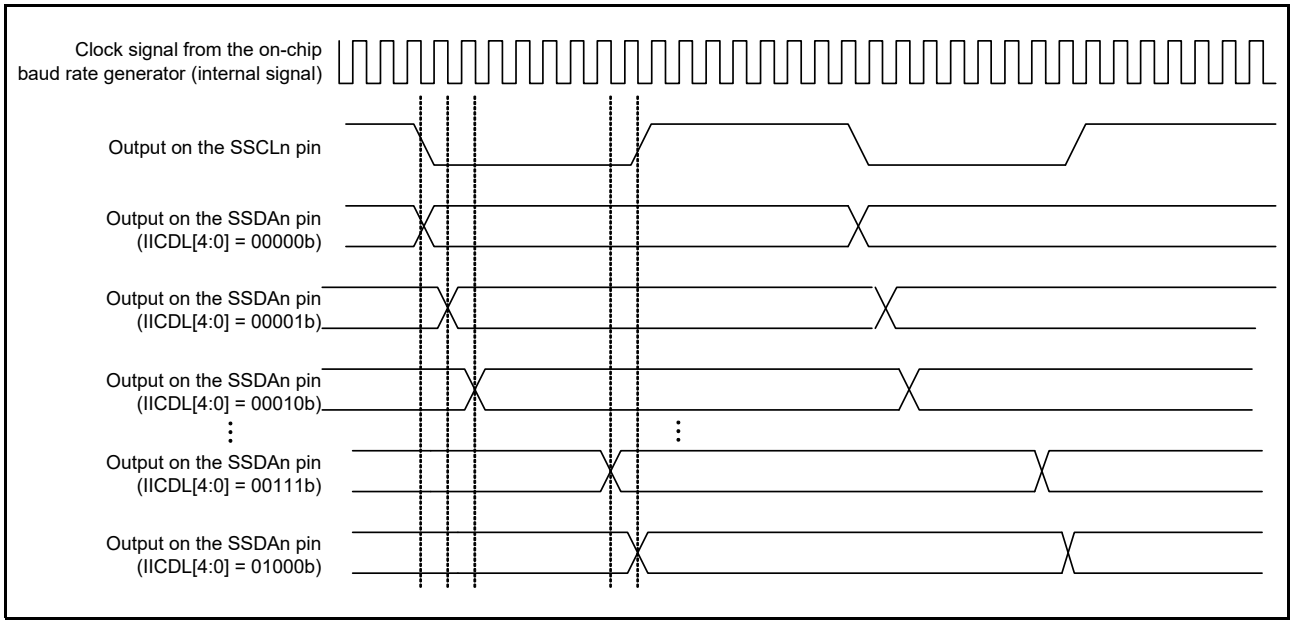


Figure 32.54 Timing of Delays in SSDA Output

32.7.4 SCI Initialization (Simple I2C Mode)

Before transferring data, write the initial value (00h) to the SCR register and initialize the interface following the example shown in Figure 32.55.
When changing the operating mode, transfer format, and so on, be sure to set the SCR register to its initial value before proceeding with the changes.
In simple I2C mode, the open-drain setting for the communication ports should be made on the port side.

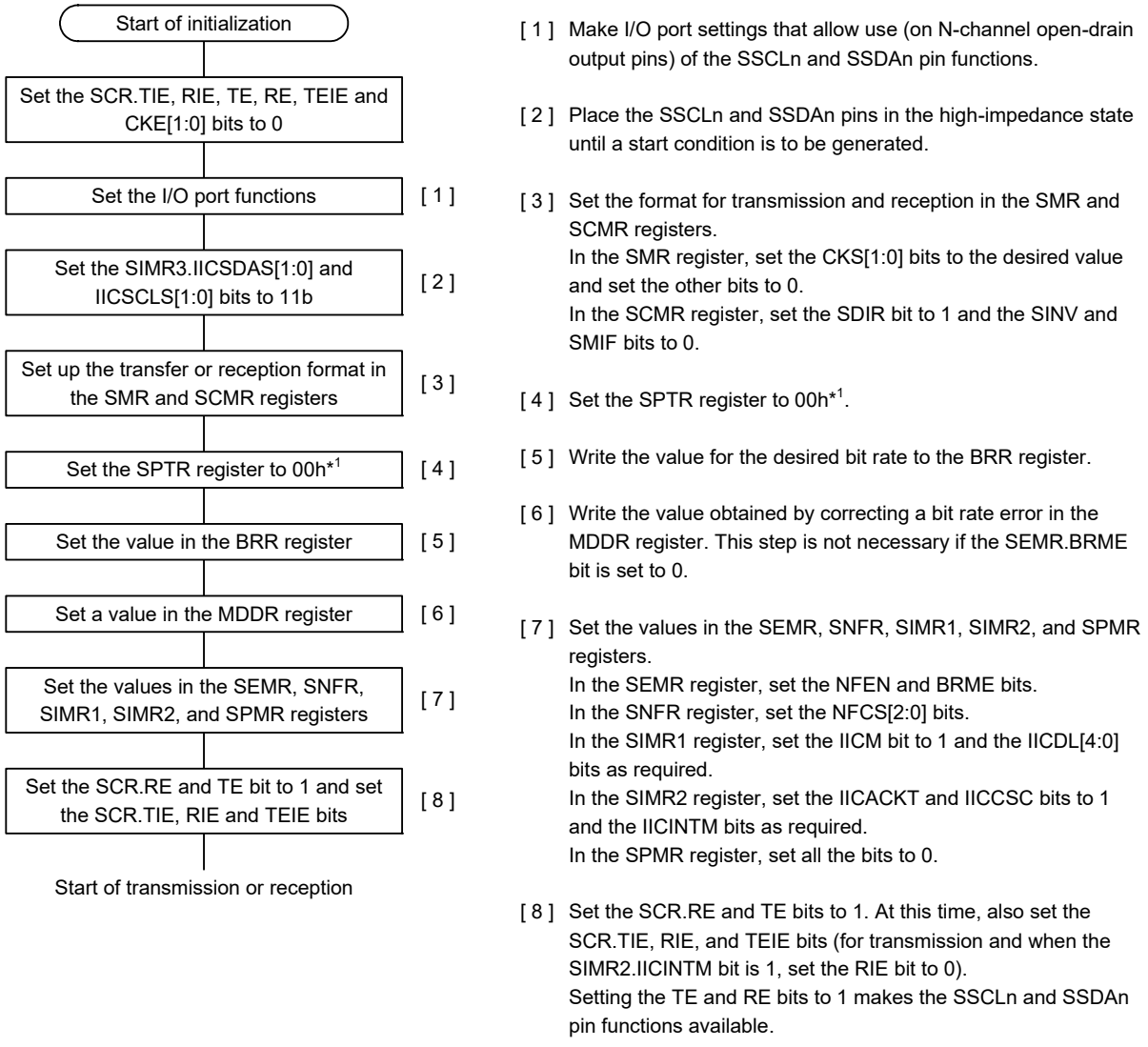


Figure 32.55 Example of the Flowchart of SCI Initialization (for Simple I2C Mode)

32.7.5 Operation in Master Transmission (Simple I2C Mode)

Figure 32.56 and Figure 32.57 show examples of operations in master transmission and Figure 32.58 is a flowchart showing the procedure for data transmission. Refer to Table 32.39 for more information on the STI interrupt.
When 10-bit slave addresses are in use, steps [3] and [4] in Figure 32.58 are repeated twice.
In simple I2C mode, the transmit data empty interrupt (TXI) is generated when communication of one frame is completed, unlike the TXI interrupt request generation timing during clock synchronous transmission.

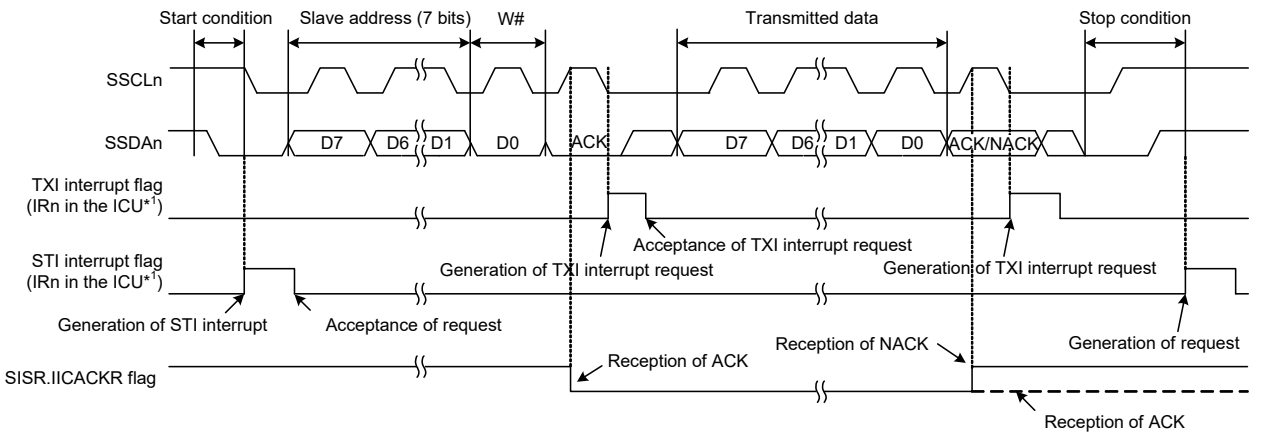


Figure 32.56 Example 1 of Operations for Master Transmission in Simple I2C-bus Mode (with 7-Bit Slave Addresses, Transmission Interrupts, and Reception Interrupts in Use)

When the SIMR2.IICINTM bit is set to 0 (use ACK/NACK interrupts) during master transmission, the DTC or DMAC is activated by the ACK interrupt as the trigger and necessary number of data bytes are transmitted. When the NACK is received, error processing, such as transmission stop and retransmission, is performed by the NACK interrupt as the trigger.

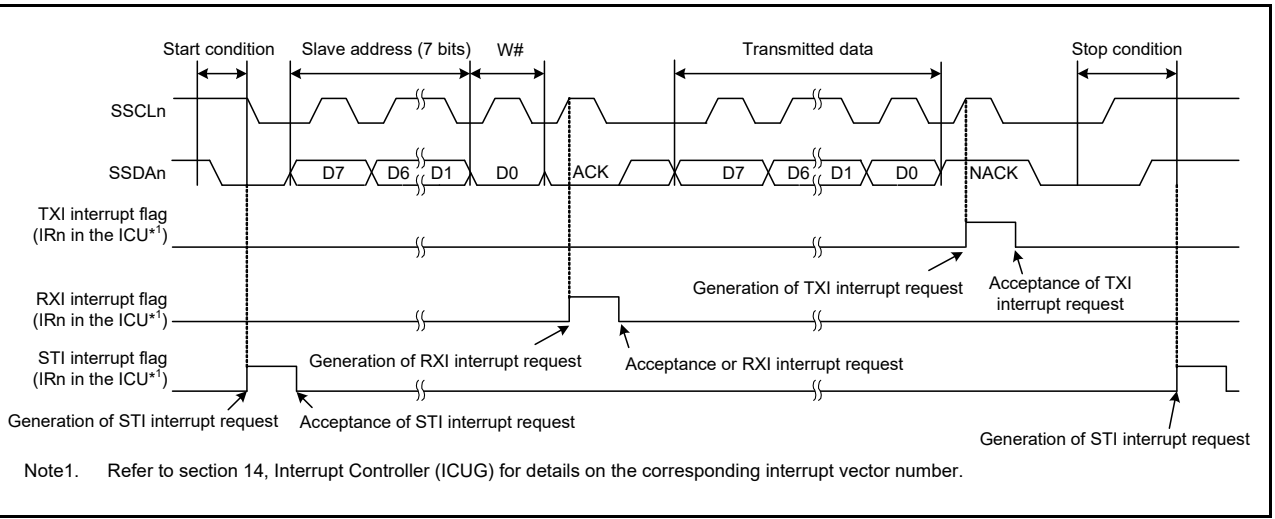
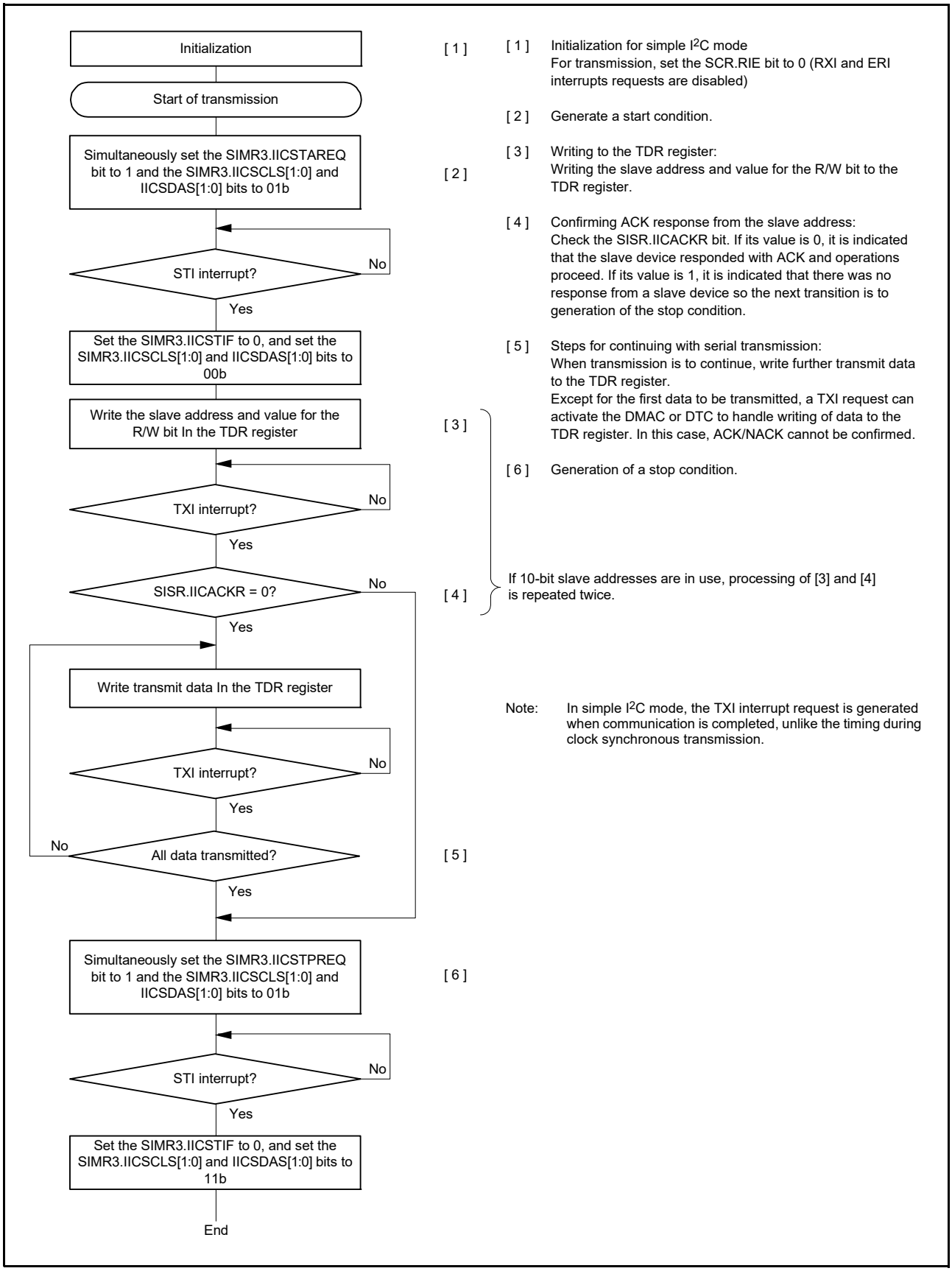


Figure 32.57 Example 2 of Operations for Master Transmission in Simple I2C-bus Mode (with 7-Bit Slave Addresses, ACK Interrupts, and NACK Interrupts in Use)



[1] Initialization for simple I²C mode
For transmission, set the SCR.RIE bit to 0 (RXI and ERI interrupts requests are disabled)

[2] Generate a start condition.

[3] Writing to the TDR register:
Writing the slave address and value for the R/W bit to the TDR register.

[4] Confirming ACK response from the slave address:
Check the SISR.IICACKR bit. If its value is 0, it is indicated that the slave device responded with ACK and operations proceed. If its value is 1, it is indicated that there was no response from a slave device so the next transition is to generation of the stop condition.

[5] Steps for continuing with serial transmission:
When transmission is to continue, write further transmit data to the TDR register.
Except for the first data to be transmitted, a TXI request can activate the DMAC or DTC to handle writing of data to the TDR register. In this case, ACK/NACK cannot be confirmed.

[6] Generation of a stop condition.

[3] to [4] If 10-bit slave addresses are in use, processing of [3] and [4] is repeated twice.

Note: In simple I²C mode, the TXI interrupt request is generated when communication is completed, unlike the timing during clock synchronous transmission.

Figure 32.58 Example of the Procedure for Master Transmission Operations in Simple I²C Mode (with Transmission Interrupts and Reception Interrupts in Use)

32.7.6 Master Reception (Simple I2C Mode)

Figure 32.59 shows an example of operations in simple I2C mode master reception and Figure 32.60 is a flowchart showing the procedure for master reception.

The value of the SIMR2.IICINTM bit is assumed to be 1 (use reception and transmission interrupts).

In simple I2C mode, the transmit data empty interrupt (TXI) is generated when communication of one frame is completed, unlike the TXI interrupt request generation timing during clock synchronous transmission.

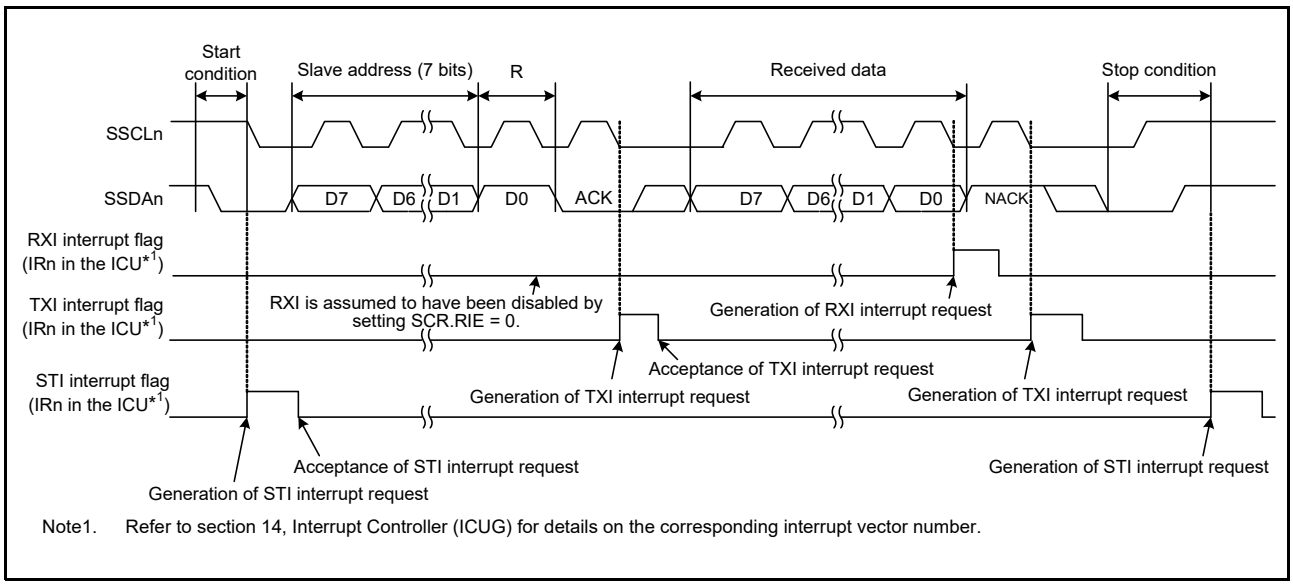
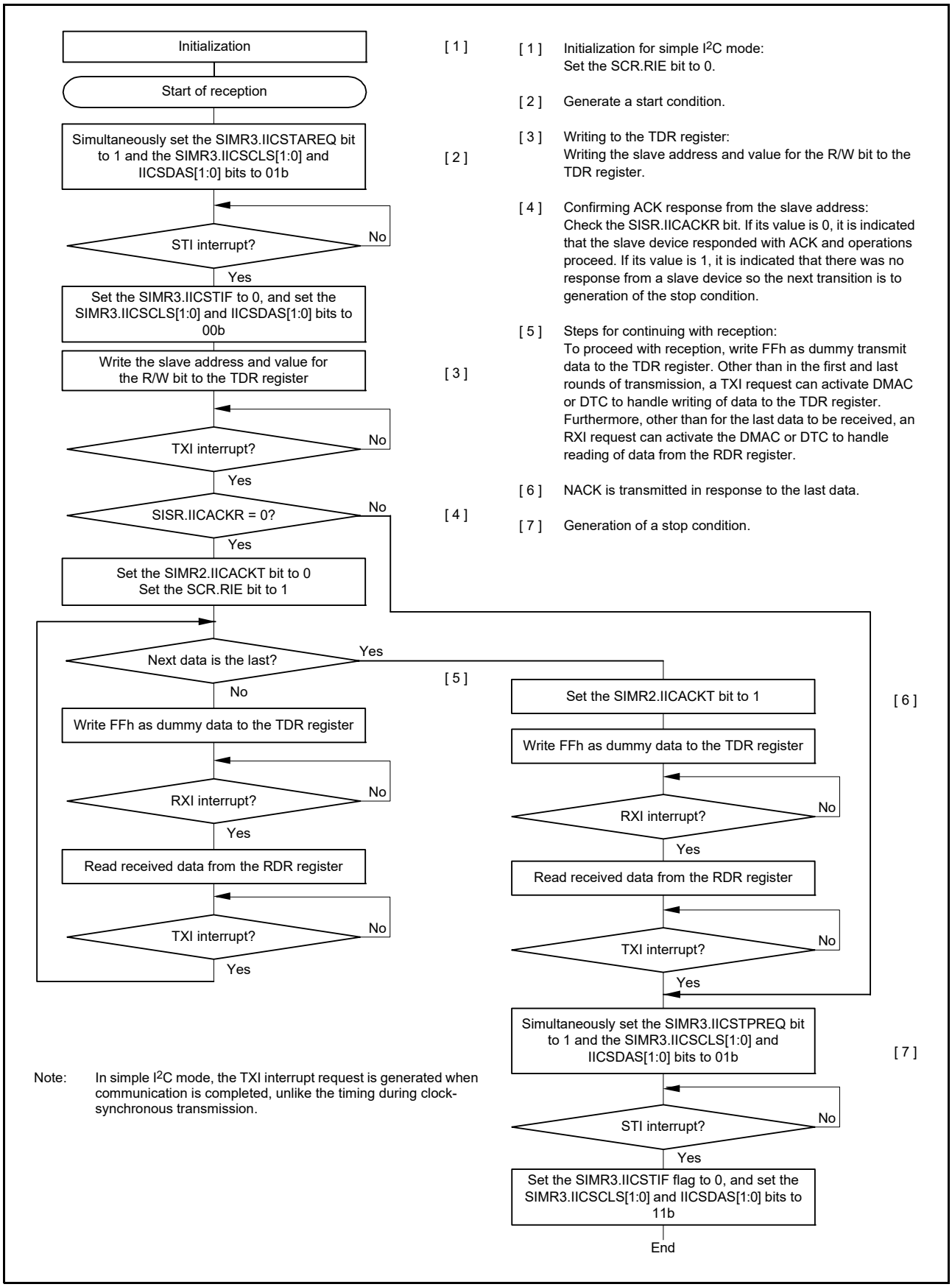


Figure 32.59 Example of Operations for Master Reception in Simple I2C-bus Mode (with 7-Bit Slave Addresses, Transmission Interrupts, and Reception Interrupts in Use)



[1] Initialization for simple I²C mode:
Set the SCR.RIE bit to 0.

[2] Generate a start condition.

[3] Writing to the TDR register:
Writing the slave address and value for the R/W bit to the TDR register.

[4] Confirming ACK response from the slave address:
Check the SISR.IICACKR bit. If its value is 0, it is indicated that the slave device responded with ACK and operations proceed. If its value is 1, it is indicated that there was no response from a slave device so the next transition is to generation of the stop condition.

[5] Steps for continuing with reception:
To proceed with reception, write FFh as dummy transmit data to the TDR register. Other than in the first and last rounds of transmission, a TXI request can activate DMAC or DTC to handle writing of data to the TDR register. Furthermore, other than for the last data to be received, an RXI request can activate the DMAC or DTC to handle reading of data from the RDR register.

[6] NACK is transmitted in response to the last data.

[7] Generation of a stop condition.

Note: In simple I²C mode, the TXI interrupt request is generated when communication is completed, unlike the timing during clock-synchronous transmission.

Figure 32.60 Example of the Procedure for Master Reception Operations in Simple I²C Mode (with Transmission Interrupts and Reception Interrupts in Use)

32.7.7 Recovery from Bus Hang-up

If the bus is stuck by an abnormal state in SCI because of the communication error, reset the SCI according to the following steps and release the bus.

(1) Set the SCR.TE and RE bit to 0 at the same time to reset SCI.
(2) Set the SIMR3 register to F0h to release the bus.
(3) If the SSR.RDRF flag is 1, dummy-read the RDR register to clear the flag.
(4) Set the SCR.TE and RE bit to 1 at the same time.

32.8 Operation in Simple SPI Mode

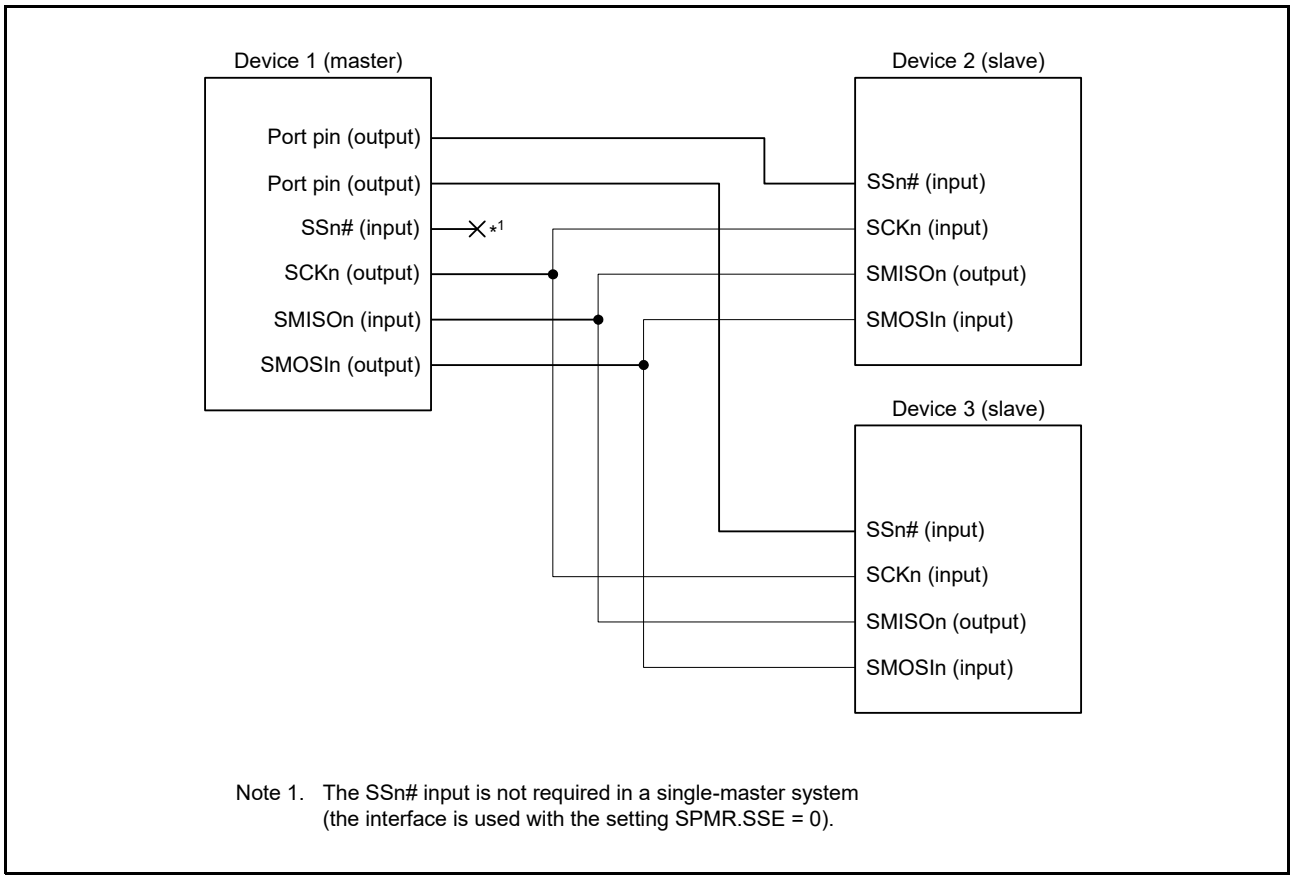
As an extended function, the SCI supports a simple SPI mode that handles transfer among one or multiple master devices and multiple slave devices.

Making the settings for clock synchronous mode (SCMR.SMIF = 0, SIMR1.IICM = 0, SMR.CM = 1) plus setting the SPMR.SSE bit to 1 places the SCI in simple SPI mode. However, the SS pin function on the master side is unnecessary for connection of the device used as the master in simple SPI mode when the configuration only has a single master, so set the SPMR.SSE bit to 0 in such cases.

Figure 32.61 shows an example of connections for simple SPI mode. Control a general port pin to produce the SS output signal from the master.

In simple SPI mode, data are transferred in synchronization with clock pulses in the same way as in clock synchronous mode. One character of data for transfer consists of 8 bits of data, and parity bits cannot be appended to this. The data can be inverted by setting the SCMR.SINV bit to 1.

Since the receiver and transmitter are independent of each other within the SCI module, full-duplex communications are possible, with a common clock signal. Furthermore, since both the transmitter and receiver have a double-buffered structure, writing of further transmit data while transmission is in progress and reading of previously received data while reception is in progress are both possible. Continuous transfer is thus possible.



Note 1. The SSn# input is not required in a single-master system (the interface is used with the setting SPMR.SSE = 0).

Figure 32.61 Example of Connections via a Simple SPI Mode (In Single Master Mode, SPMR.SSE Bit = 0)

32.8.1 States of Pins in Master and Slave Modes

The direction (input or output) of pins for the simple SPI mode interface differs according to whether the device is a master (SCR.CKE[1:0] = 00b or 01b and SPMR.MSS = 0) or slave (SCR.CKE[1:0] = 10b or 11b and SPMR.MSS = 1). Table 32.35 lists the states of pins according to the mode and the level on the SSn# pin.

Table 32.35 States of Pins by Mode and Input Level on the SSn# Pin

Mode	Input on SSn# Pin	State of SMOSIn Pin	State of SMISOn Pin	State of SCKn Pin
Master mode*1	High level (transfer can proceed)	Output for data transmission*2	Input for received data	Clock output*3
	Low level (transfer cannot proceed)	High-impedance	Input for received data (but disabled)	High-impedance
Slave mode	High level (transfer cannot proceed)	Input for received data (but disabled)	High-impedance	Clock input (but disabled)
	Low level (transfer can proceed)	Input for received data	Output for data transmission	Clock input

Note 1. When there is only a single master (SPMR.SSE = 0), transfer is possible regardless of the input level on the SSn# pin (this is equivalent to input of a high level on the SSn# pin). Since the SSn# pin function is not required, the pin is available for other purposes.
Note 2. The SMOSIn pin output is in the high-impedance state when serial transmission is disabled (SCR.TE bit = 0).
Note 3. The SCKn pin output is in the high-impedance state when serial transmission is disabled (SCR.TE and RE bits = 00b) in a multi-master configuration (SPMR.SSE = 1).

32.8.2 SS Function in Master Mode

Setting the SCR.CKE[1:0] bits to 00b and the SPMR.MSS bit to 0 selects master operation. The SSn# pin is not used in single-master configurations (SPMR.SSE = 0), so transmission or reception can proceed regardless of the value of the SSn# pin.

When the level on the SSn# pin is high in a multi-master configuration (SPMR.SSE = 1), a master device outputs clock signals from the SCKn pin before starting transmission or reception to indicate that there are no other masters or another master is performing reception or transmission. When the level on the SSn# pin is low in a multi-master configuration (SPMR.SSE = 1), there are other masters, and this indicates that transmission or reception is in progress. At this time the SMOSIn output and SCKn pins will be placed in the high-impedance state and starting transmission or reception will not be possible. Furthermore, the value of the SPMR.MFF bit will be 1, indicating a mode fault error. In a multi-master configuration, start error processing by reading SPMR.MFF flag. Also, even if a mode fault error occurs while transmission or reception is in progress, transmission or reception will not be stopped, but the SMOSIn and SCKn pin output will be placed in the high-impedance state after the completion of the transfer.

Control a general port pin to produce the SS output signal from the master.

32.8.3 SS Function in Slave Mode

Setting the SCR.CKE[1:0] bits to 10b and the SPMR.MSS bit to 1 selects slave operation. When the level on the SSn# pin is high, the SMISOn output pin will be in the high-impedance state and clock input through the SCKn pin will be ignored. When the level on the SSn# pin is low, clock input through the SCKn pin will be effective and transmission or reception can proceed.

If the input on the SSn# pin changes from low to high level during transmission or reception, the SMISOn output pin will be placed in the high-impedance state. Meanwhile, the internal processing for transmission or reception will continue at the rate of the clock input through the SCKn pin until processing for the character currently being transmitted or received is completed, after which it stops. At that time, an interrupt (the appropriate one from among TXI, RXI, and TEI) will be generated.

32.8.4 Relationship between Clock and Transmit/Receive Data

The CKPOL and CKPH bits in the SPMR register can be used to set up the clock for use in transmission and reception in four different ways. The relation between the clock signal and the transmission and reception of data is shown in Figure 32.62. The relation is the same for both master and slave operation.

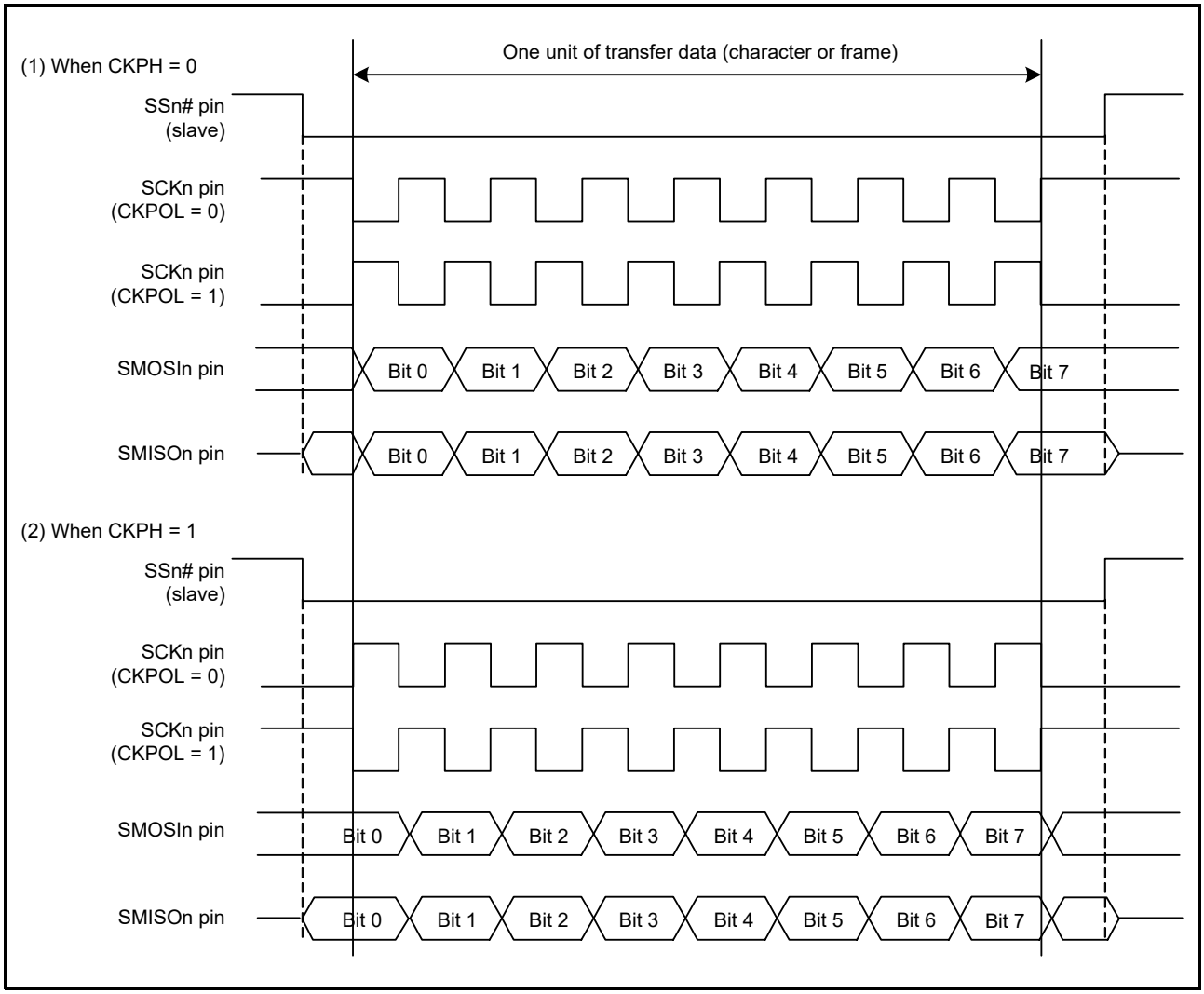


Figure 32.62 Relation between Clock Signal and Transmit/Receive Data in Simple SPI Mode

32.8.5 SCI Initialization (Simple SPI Mode)

The procedure is the same as for initialization in clock synchronous mode Figure 32.28, Sample SCI Initialization Flowchart. The CKPOL and CKPH bits in the SPMR register must be set to ensure that the kind of clock signal they select is suitable for both master and slave devices.
For initialization, changes to the operating mode, changes to the transfer format, and so on, initialize the SCR register before proceeding with changes.
As well as setting the RE bit to 0, note that the SSR.ORER, FER, and PER flags, as well as the RDR register, are not initialized.

32.8.6 Transmission and Reception of Serial Data (Simple SPI Mode)

In master operation, ensure that the SSn# pin of the slave device on the other side of the transfer is at the low level before starting the transfer and at the high level on completion of the transfer. Otherwise, the procedures are the same as in clock synchronous mode.

32.9 Bit Rate Modulation Function

The bit rate modulation function corrects the bit rate by thinning out the specified amount of clocks from those input to the baud rate generator.
When the SEMR.BRME bit is 1, the baud rate generator validates and counts the average interval of the number of clocks set in the MDDR register out of the total 256 clocks input.
Figure 32.63 assumes the SCI is in asynchronous mode, bits SMR.CKS[1:0] are 00b, the BRR register is 00h, and the MDDR register is 160. In this example, the cycle of the base clock is evenly corrected to 256/160, and the bit rate is corrected to 160/256. Note that there is an imbalance in thinning out the internal clock, and expansion and contraction occur in the pulse width of the base clock.

Note: Do not use this function in the highest speed settings (SMR.CKS[1:0] = 00b, SCR.CKE[1] = 0, and BRR = 0) in clock synchronous mode and simple SPI mode.

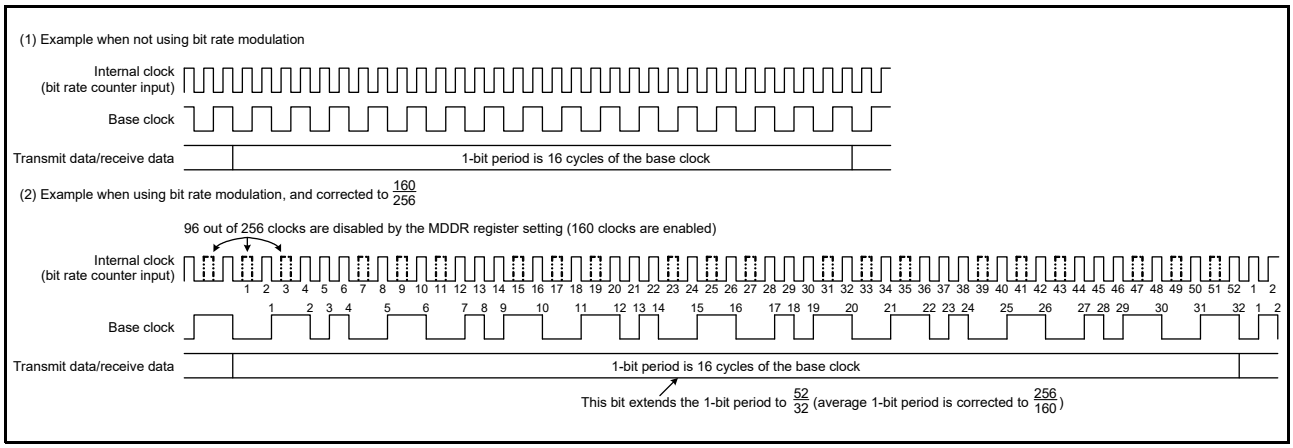


Figure 32.63 Example of the Base Clock When the Bit Rate Modulation Function is Used

The input of a clock signal with a shorter period to the baud rate generator reduces difference in the generated base clock period and, since the division ratio of the baud rate generator also becomes larger, reduces difference in the length of the 1-bit period.

32.10 Extended Serial Mode Control Section: Description of Operation

32.10.1 Serial Transfer Protocol

The extended serial mode control section of the SCI12 can realize the serial transfer protocol composed of Start Frames and Information Frames that is shown in Figure 32.64.
A Start Frame is composed of a Break Field, Control Field 0, and Control Field 1. An Information Frame is composed of a number of Data Fields, a CRC16 Upper Field, and a CRC16 Lower Field.

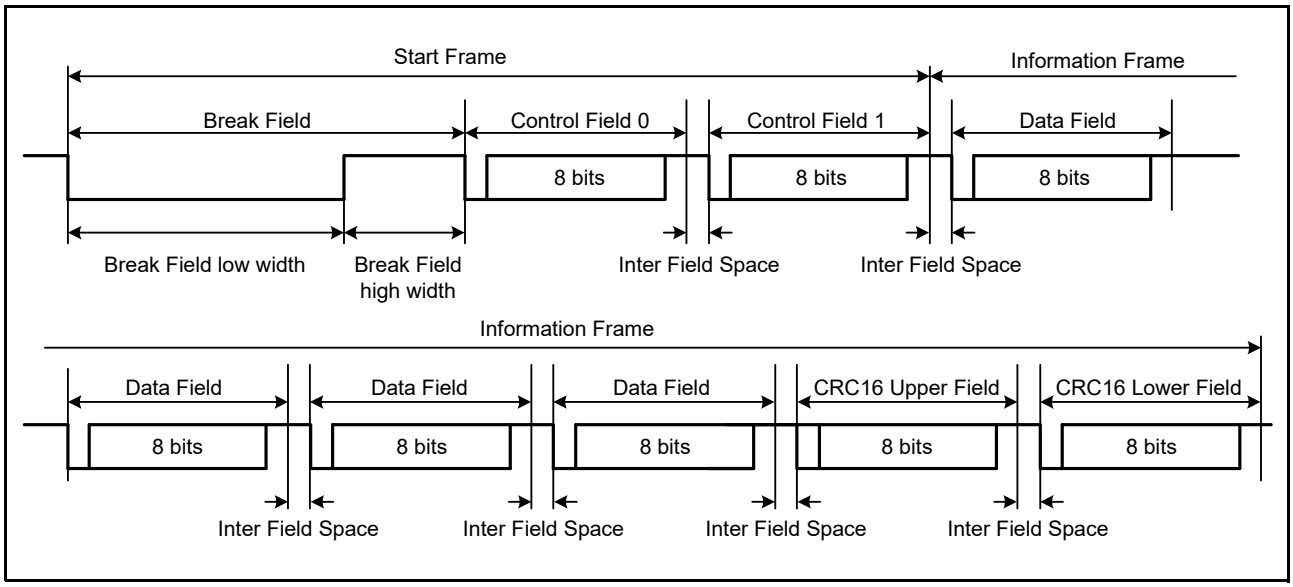


Figure 32.64 Protocol for Serial Transfer by the Extended Serial Mode Control Section

32.10.2 Transmitting a Start Frame

Figure 32.65 shows an example of operations to transmit a Start Frame, which is composed of the Break Field low width, Control Field 0, and Control Field 1. Figure 32.66 and Figure 32.67 are flowcharts for the transmission of a Start Frame.
Operations when the extended serial mode control section is to be used to transmit a Start Frame are as listed below. Be sure to use the SCI12 in asynchronous mode.

(1) With Break Field low width output mode as the operating mode for the timer, writing 1 to the TCR.TCST bit starts counting by the timer, and the low level will be output from the TXDX12 pin over the period corresponding to registers TCNT and TPRE settings.
(2) The output on the TXDX12 pin is inverted when the timer counter underflows, and the STR.BFDF flag is set to 1. An SCIX0 interrupt is also generated if the value of the ICR.BFDIE bit is 1.
(3) Write 0 to the TCR.TCST bit to stop counting by the timer, and send the data for Control Field 0. After the Break Field low width output, stop counting before the next underflow occurs.
(4) When the data for Control Field 0 have been transmitted, the data for Control Field 1 is transmitted.
(5) When the data for Control Field 1 have been transmitted, an Information Frame is transmitted.

Omit the Break Field and Control Field 0 to suit the structure of the Start Frame.

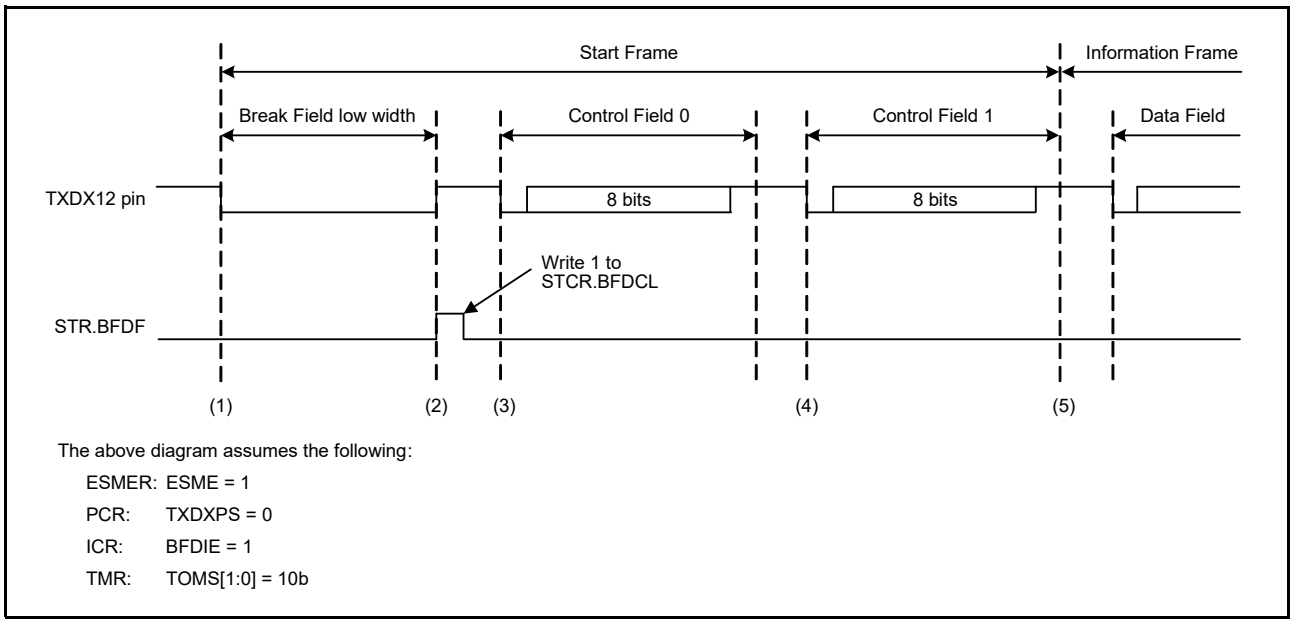


Figure 32.65 Example of Operations When Transmitting a Start Frame

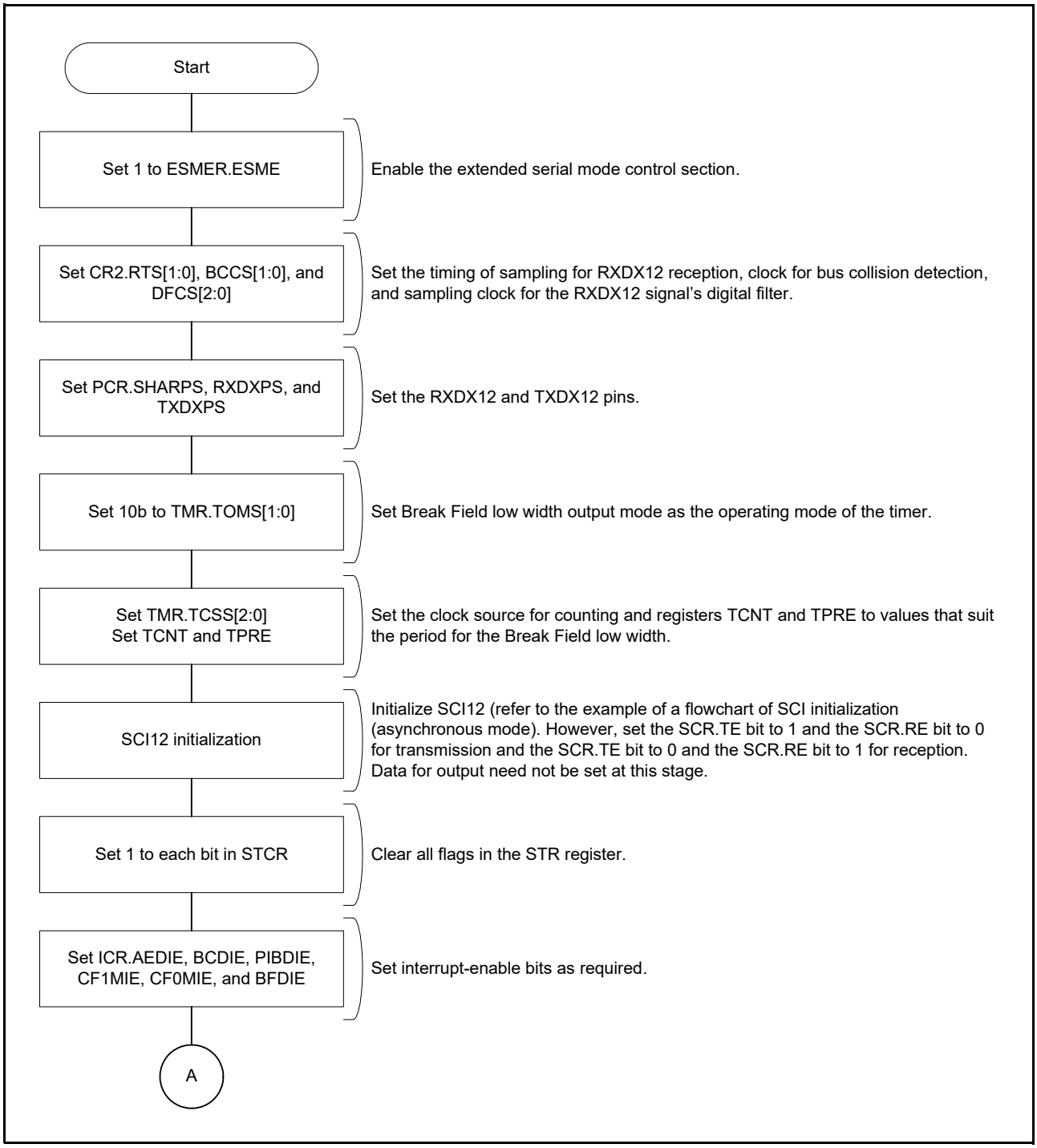


Figure 32.66 Example of Start Frame Transmission (1/2)

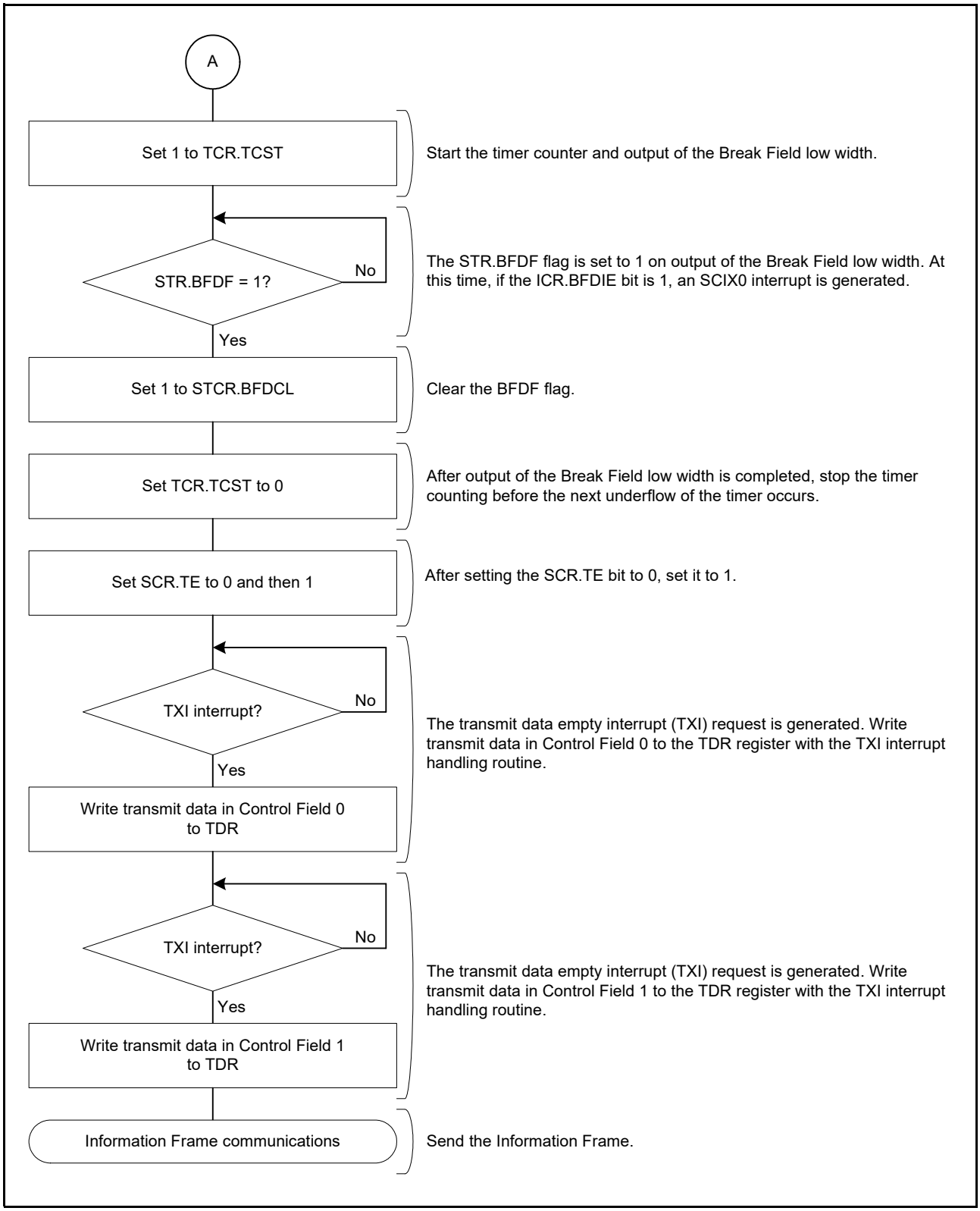


Figure 32.67 Example of Start Frame Transmission (2/2)

32.10.3 Receiving a Start Frame

The extended serial mode control section is capable of receiving Start Frames with the structures listed in Table 32.36.

Table 32.36 Structures of Start Frames

Bit Setting		Structures of Start Frames
BFE	CF0RE	
0	0	Control Field 1 (8 bits) → Information Frame
0	1	Control Field 0 (8 bits) → Control Field 1 (8 bits) → Information Frame
1	0	Break Field (low width) → Control Field 1 (8 bits) → Information Frame
1	1	Break Field (low width) → Control Field 0 (8 bits) → Control Field 1 (8 bits) → Information Frame

Figure 32.68 shows an example of operations to receive a Start Frame, which is composed of the Break Field low width, Control Field 0, and Control Field 1. Figure 32.69 and Figure 32.70 are flowcharts for the reception of a Start Frame, and Figure 32.71 is a state transition diagram when receiving a Start Frame.
Operations when the extended serial mode control section is to be used to receive a Start Frame are as listed below. Be sure to use the SCI12 in asynchronous mode.

(1) With Break Field low width detection mode as the operating mode for the timer, writing 1 to the CR3.SDST bit enables detection of the Break Field low width.
(2) Low-level input on the RXDX12 pin continuing over a period longer than that corresponding to the settings of registers TCNT and TPRE is detected as the Break Field low width. At this time, the STR.BFDF flag is set to 1. An SCIX0 interrupt is also generated if the value of the ICR.BFDIE bit is 1.
(3) When the input from the RXDX12 pin goes high after the Break Field low width, the CR0.RXDSF flag becomes 0 and reception of Control Field 0 starts.
(4) If the data received in Control Field 0 match the data set in the CF0DR register, the STR.CF0MF flag is set to 1. An SCIX1 interrupt is also generated if the value of the ICR.CF0MIE bit is 1. Reception of Control Field 1 starts after that. If the data received in Control Field 0 do not match the data set in the CF0DR register, a transition to the state prior to Break Field low width detection proceeds.
(5) If the data received in Control Field 1 match the data set in registers PCF1DR and SCF1DR, the STR.CF1MF flag is set to 1. An SCIX1 interrupt is also generated if the value of the ICR.CF1MIE bit is 1. Transfer of the Information Frame starts after that. If the data received in Control Field 1 do not match the data set in either or both of registers PCF1DR and SCF1DR, a transition to the state prior to Break Field low width detection proceeds.

Omit the Break Field and Control Field 0 to suit the structure of the Start Frame.

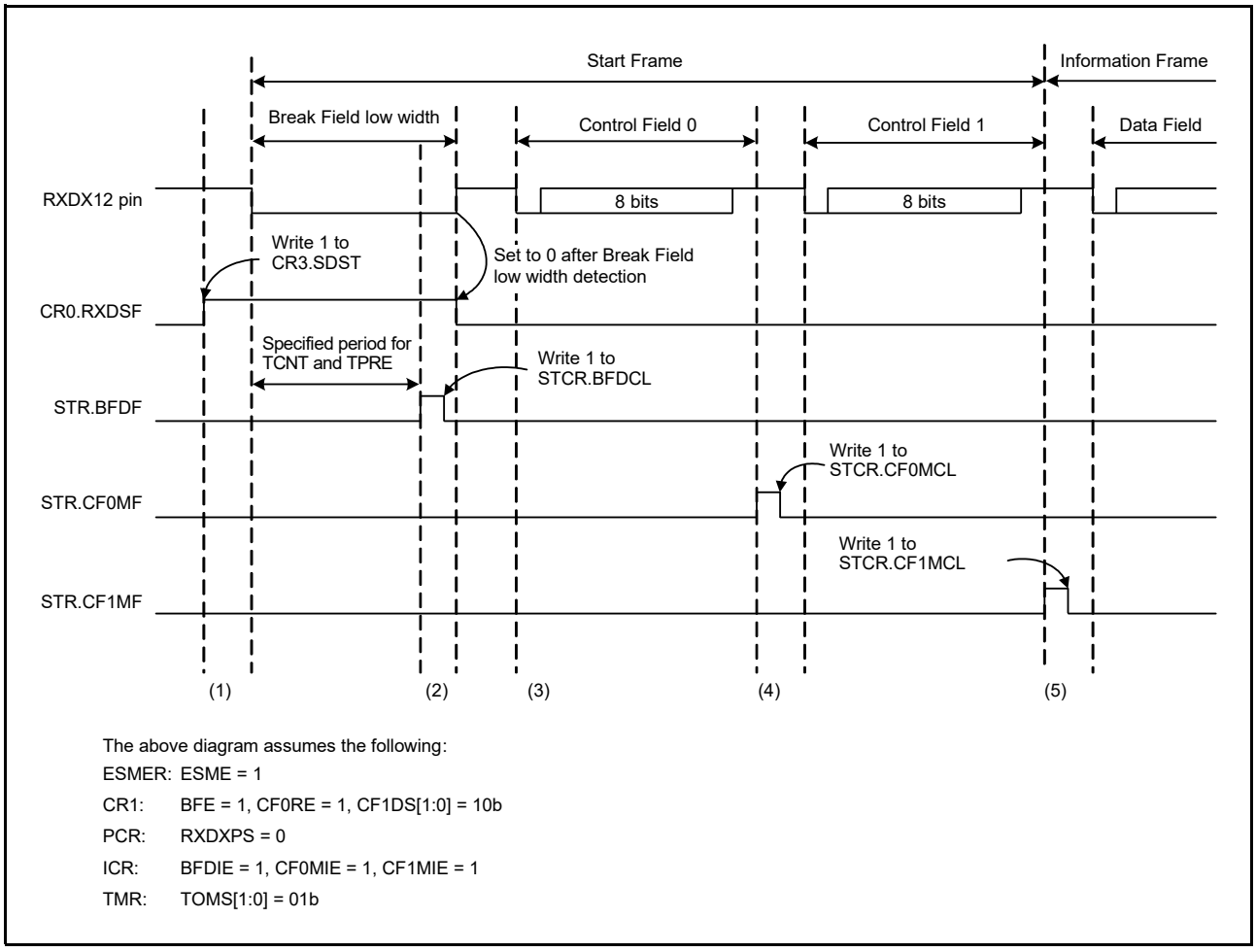


Figure 32.68 Example of Operations at the Time of Start Frame Reception

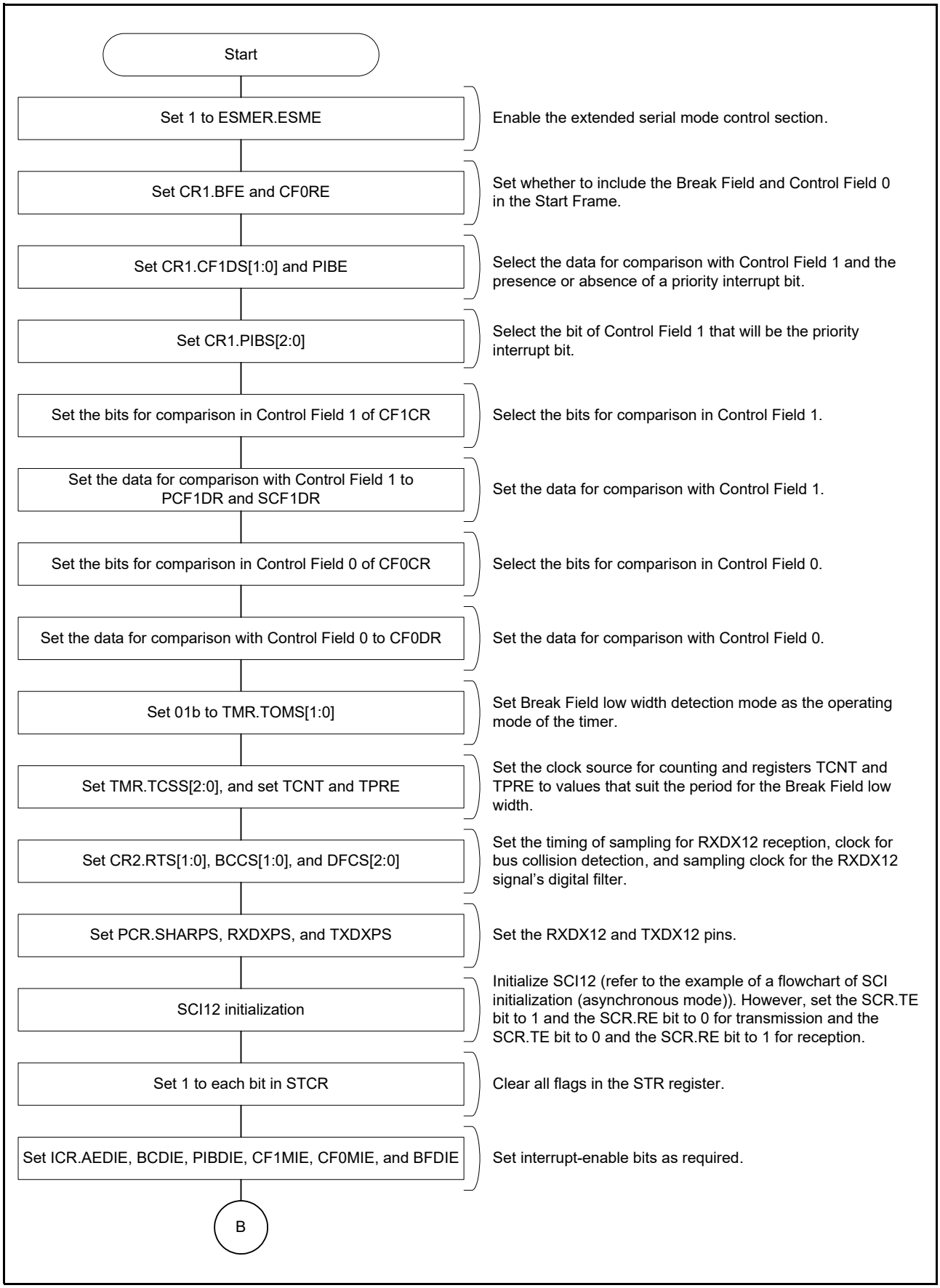


Figure 32.69 Sample Flowchart for Reception of a Start Frame (1)

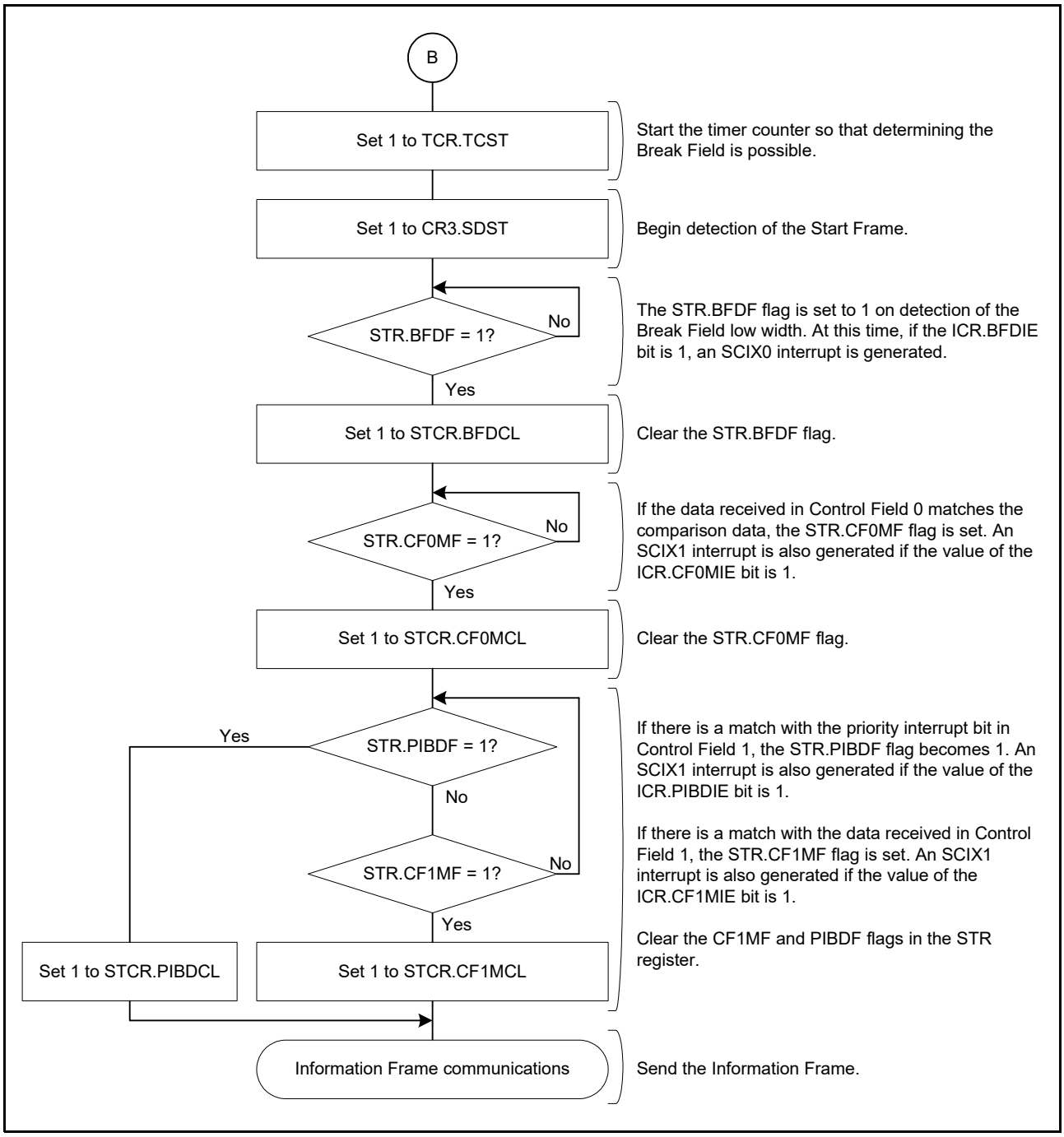


Figure 32.70 Sample Flowchart for Reception of a Start Frame (2)

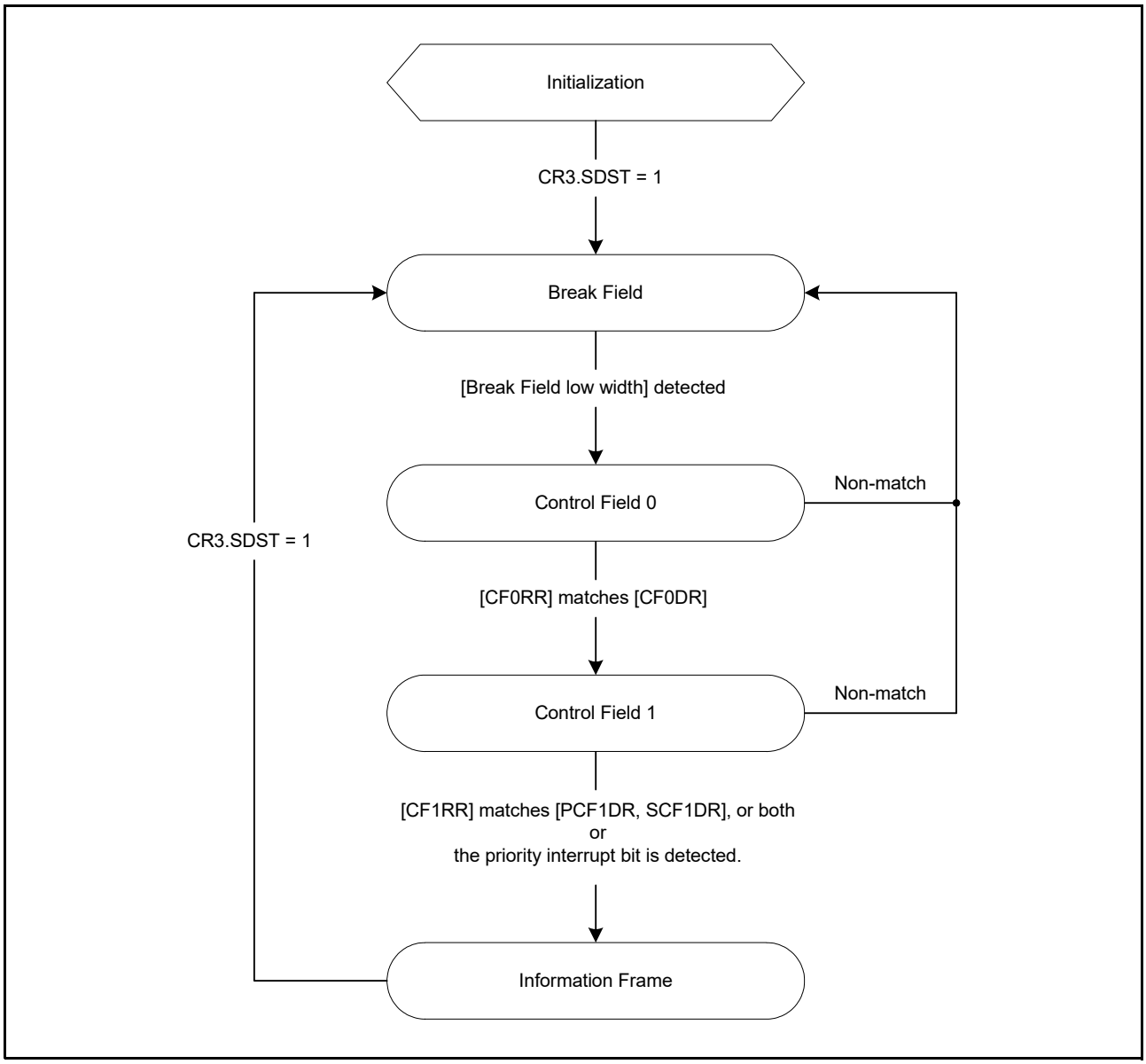


Figure 32.71 State Transitions When Receiving a Start Frame

32.10.3.1 Priority Interrupt Bit

Figure 32.72 shows an example of operation in Start Frame reception where a priority interrupt bit is in use. Setting the CR1.PIBE bit to 1 enables the use of a priority interrupt bit.
Operations of the extended serial mode control section in start Frame reception where a priority interrupt bit is in use are as described below.

Steps (1) to (4) are the same as in Figure 32.68, for Start Frame reception.

(5) If the value of the bit selected by the CR1.PIBS[2:0] bits matches the corresponding bit in the PCF1DR register, the STR.PIBDF flag is set to 1. An SCIX1 interrupt is also generated if the value of the ICR.PIBDIE bit is 1. Transfer of the Information Frame starts after that. If the data received in Control Field 1 do not match the data set in either or both of registers PCF1DR and SCF1DR and the priority interrupt bit is not detected, a transition to the state prior to Break Field low width detection proceeds.

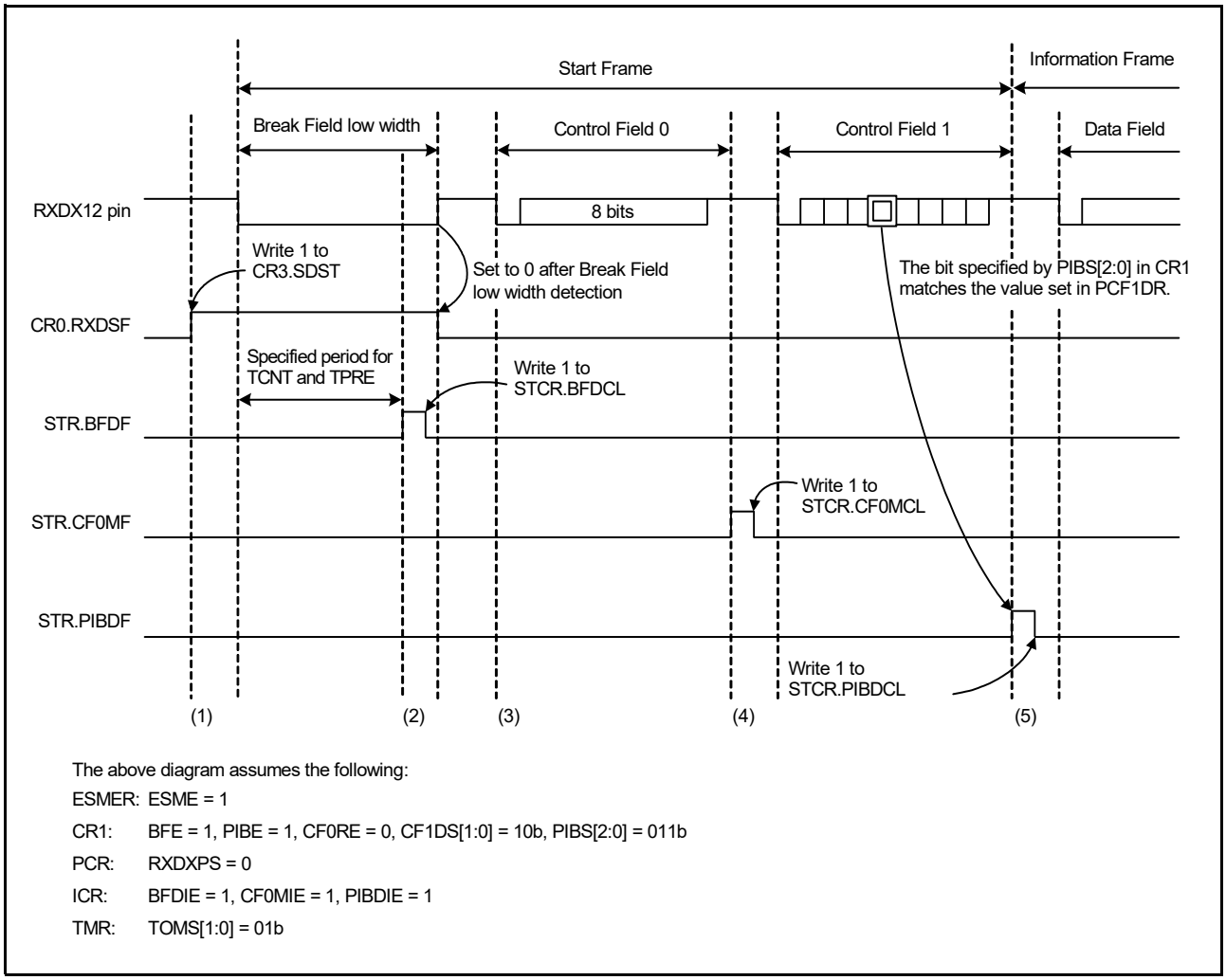


Figure 32.72 Example of Operations When Receiving a Start Frame While the CR1.PIBE Bit is 1

32.10.4 Detection of Bus Collisions

Detection of bus collisions operate for cases where output of the Break Field low width and transmission of data are in progress when the ESMER.ESME bit and the SCR.TE bit are set to 1.

Figure 32.73 shows an example of operations with bus collision detection. Signals output through TXDX12 and input through RXDX12 are sampled with the bus collision detection clock set with the CR2.BCCS[1:0] bits as the sampling clock, and the STR.BCDF flag is set to 1 if the signals fail to match three times in a row. An SCIX2 interrupt is also generated if the value of the ICR.BCDIE bit is 1.

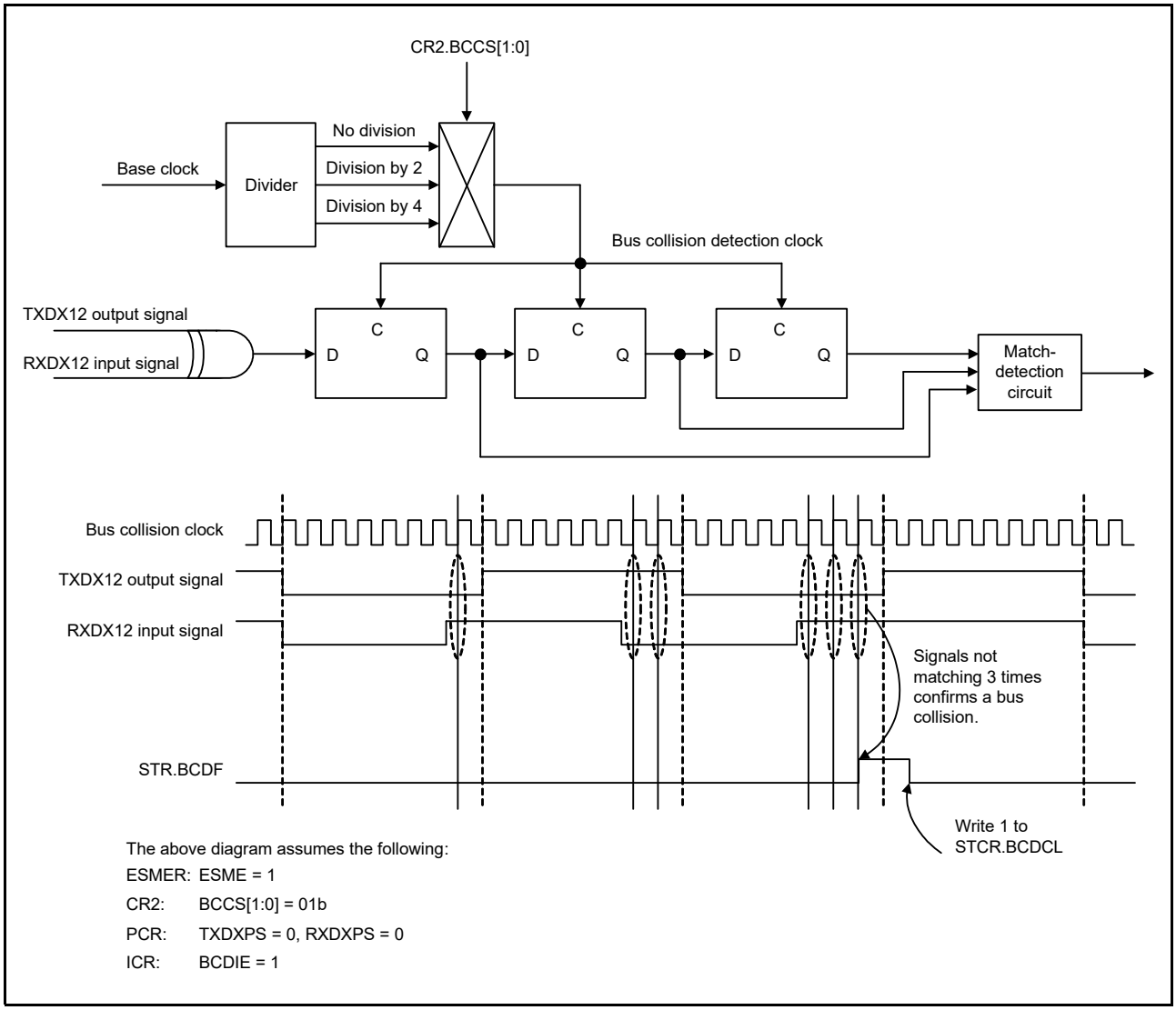


Figure 32.73 Example of Operations with Bus Collision Detection

32.10.5 Digital Filter for Input on the RXDX12 Pin

Signals input through the RXDX12 pin can be passed through a digital filter before they are conveyed to the internal circuits. The digital filter consists of three flip-flop circuit stages connected in series and a match-detecting circuit. The CR2.DFCS[2:0] bits select the sampling clock for the RXDX12 pin input signals. If the outputs of all three latches match, the given level is conveyed to subsequent circuits. If the levels do not match, the previous value is retained. In other words, levels are confirmed as being the signal if they are retained for at least three cycles of the sampling clock but judged to be noise rather than changes in the signal level if they change within three cycles of the sampling clock. Figure 32.74 shows an example of operations with the digital filter.

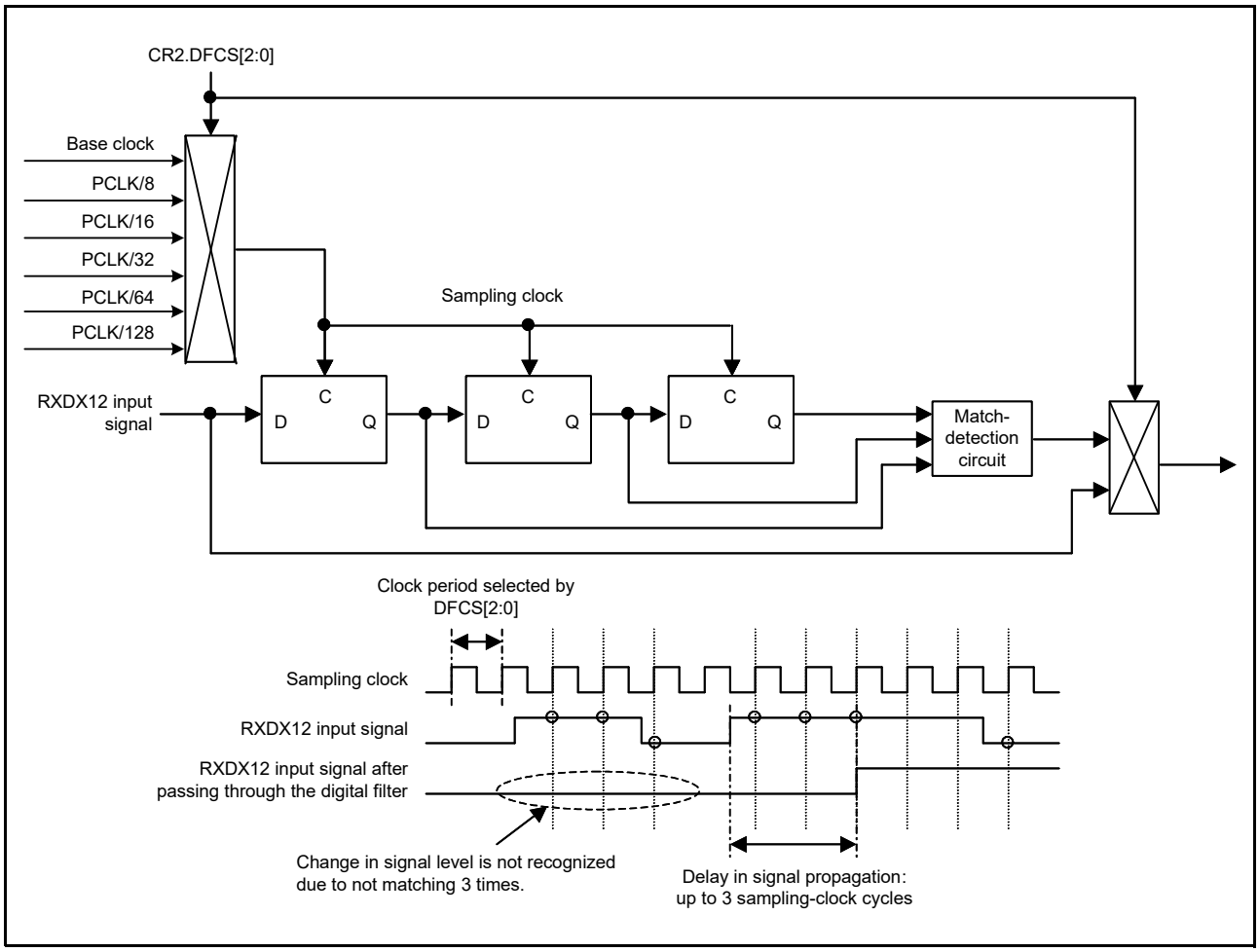
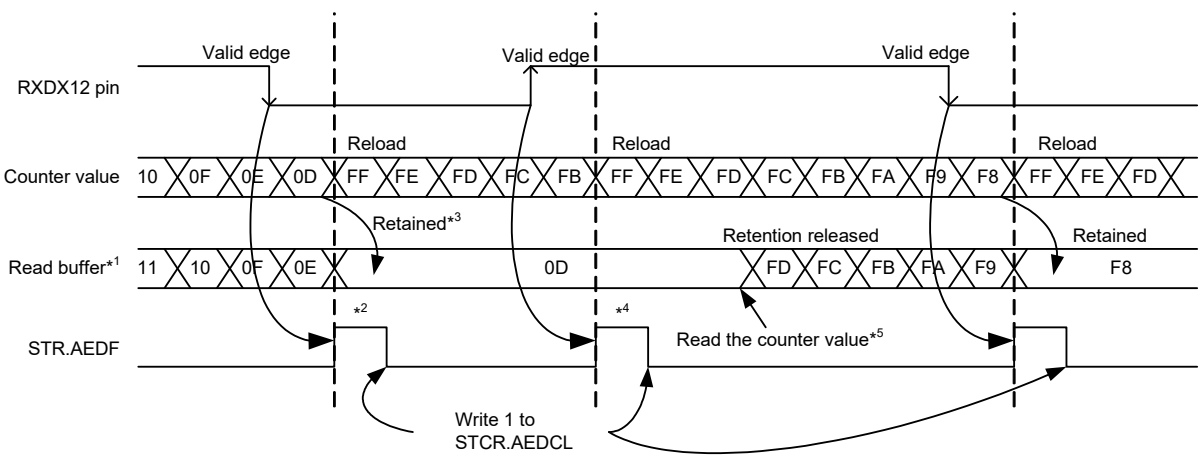


Figure 32.74 Example of Operations with the Digital Filter

32.10.6 Bit Rate Measurement

The bit rate measurement function measures the intervals between rising and falling edges and between falling and rising edges of the signal input from the RXDX12 pin. Figure 32.75 shows an example of operations for bit rate measurement.

(1) Writing 1 to the CR0.BRME bit enables bit rate measurement. Only set the BRME bit to 1 when you wish to proceed with bit rate measurement. Furthermore, bit rate measurement will not proceed during a Break Field, even if the BRME bit is set to 1.
(2) After detection of the Break Field low width, bit rate measurement starts when the level input on the RXDX12 pin becomes high.
(3) Once bit rate measurement has started, counter values from the timer are retained in the read buffers on the input of valid edges from the RXDX12 pin (rising and falling edges) and the counter is reloaded. An SCIX3 interrupt is also generated if the value of the ICR.AEDIE bit is 1. Retention by registers TCNT and TPRE is released by reading these registers.
(4) The bit rate as calculated from the values counted during intervals between valid edges can be used for adjusting the rate by changing the settings of the BRR register. To disable the bit rate measurement after a match with Control Field 1, write 0 to the CR0.BRME bit.



Note 1. The TCNT or TPRE register indicates the contents of the read buffers.
Note 2. The STR.AEDF flag becomes 1 on detection of a valid edge.
Note 3. When an effective edge is detected, the current value of the counter is stored in the read buffer.
Note 4. The value retained in the read buffers due to one valid edge is not updated to the new value even if a further valid edge is detected.
Note 5. Retention of a value in a read buffer due to a valid edge is released by reading the TCNT or TPRE register.

Figure 32.75 Example of Operations for Bit Rate Measurement

32.10.7 Selectable Timing for Sampling Data Received through RXDX12

The extended serial mode control section provides a way of adjusting the timing for the sampling of data received through the RXDX12 pin by setting the CR2.RTS[1:0] bits to select the rising edges of 8th, 10th, 12th, or 14th cycle of the base clock. If the value of the SEMR.ABCS bit is 1, the bits select the rising edges of 4th, 5th, 6th, or 7th cycle of the base clock. Figure 32.76 shows timing for the sampling of data received through RXDX12.

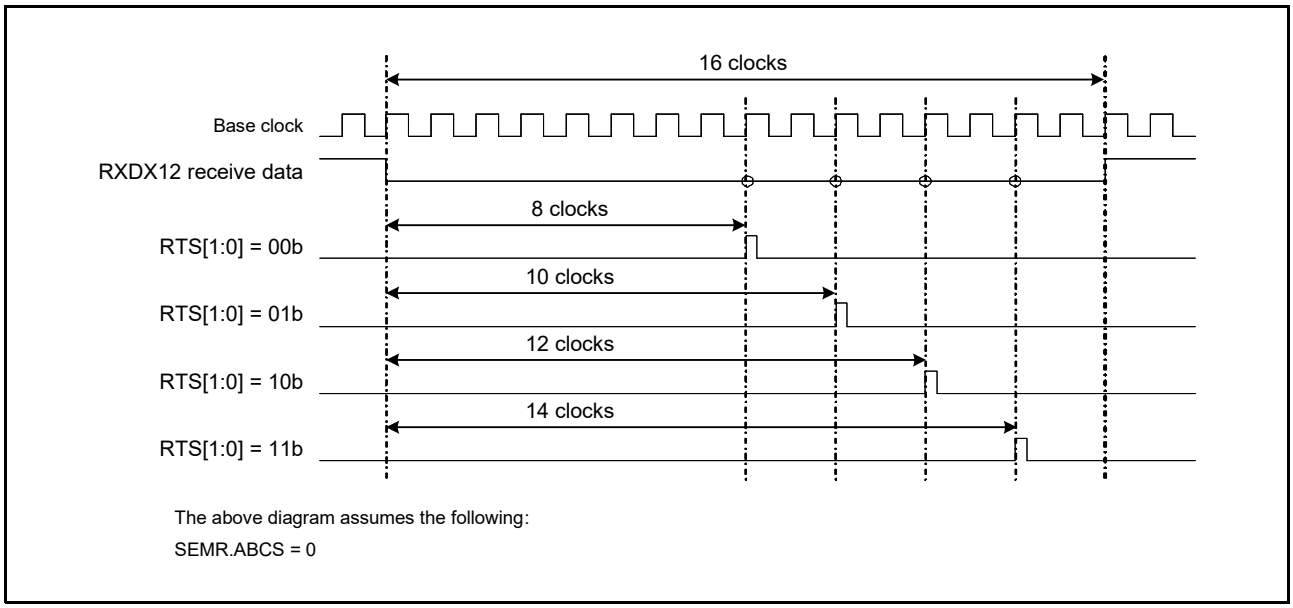


Figure 32.76 Timing for Sampling of Data Received through RXDX12

32.10.8 Timer

The timer has the following operating modes.

(1) Break Field Low Width Output Mode

This mode is for output through the TXDX12 pin of the low level over the Break Field low width at the transmission of a Start Frame. Setting the TMR.TOMS[1:0] bits to 10b switches operation to Break Field low width output mode. The TMR.TCSS[2:0] bits select the clock source for the counter. When the TCR.TCST bit is set to 1, the output on the TXDX12 pin goes to the low level and counting starts. When the timer underflows, the output on the TXDX12 pin goes to the high level and the STR.BFDF flag is set to 1. An SCIX0 interrupt is also generated if the value of the ICR.BFDIE bit is 1. When 0 is written to the TCR.TCST bit, counting stops after reloading of registers TPRE and TCNT. After output of the Break Field low width is completed, stop the timer before it underflows again. Figure 32.77 shows an example of operations in Break Field low width output mode.

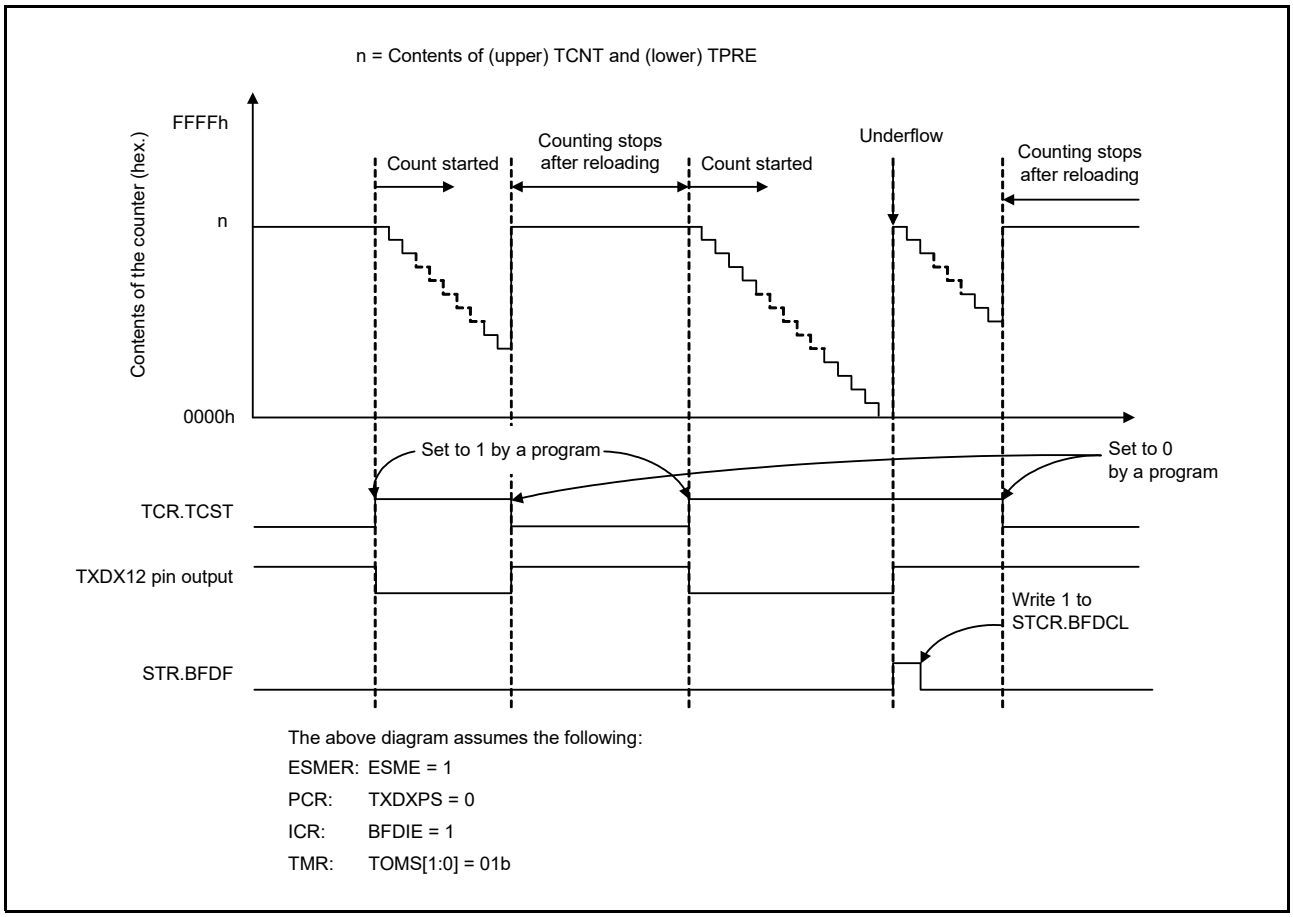


Figure 32.77 Example of Operations in Break Field Low Width Output Mode

(2) Break Field Low Width Determination Mode

This mode is for determining the Break Field low width in the input signal on the RXDX12 pin at the reception of a Start Frame. Setting the TMR.TOMS[1:0] bits to 01b switches operation to Break Field low width determination mode. The TMR.TCSS[2:0] bits select the clock source for the counter. When the TCR.TCST bit is set to 1, the interface enters the Break Field low width determinable state. Determination starts when a low level is input from the RXDX12 pin. When a high level is then input on the RXDX12 pin, registers TPRE and TCNT are reloaded and the interface enters the Break Field low width determinable state. When the timer underflows during Break Field low width determination, the STR.BFDF flag is set to 1. An SCIX0 interrupt is also generated if the value of the ICR.BFDIE bit is 1. If an underflow of the timer during data transfer cause a problem in the form of interrupt generation, stop the timer after Break Field low width determination. Figure 32.78 shows an example of operations in Break Field low width output mode.

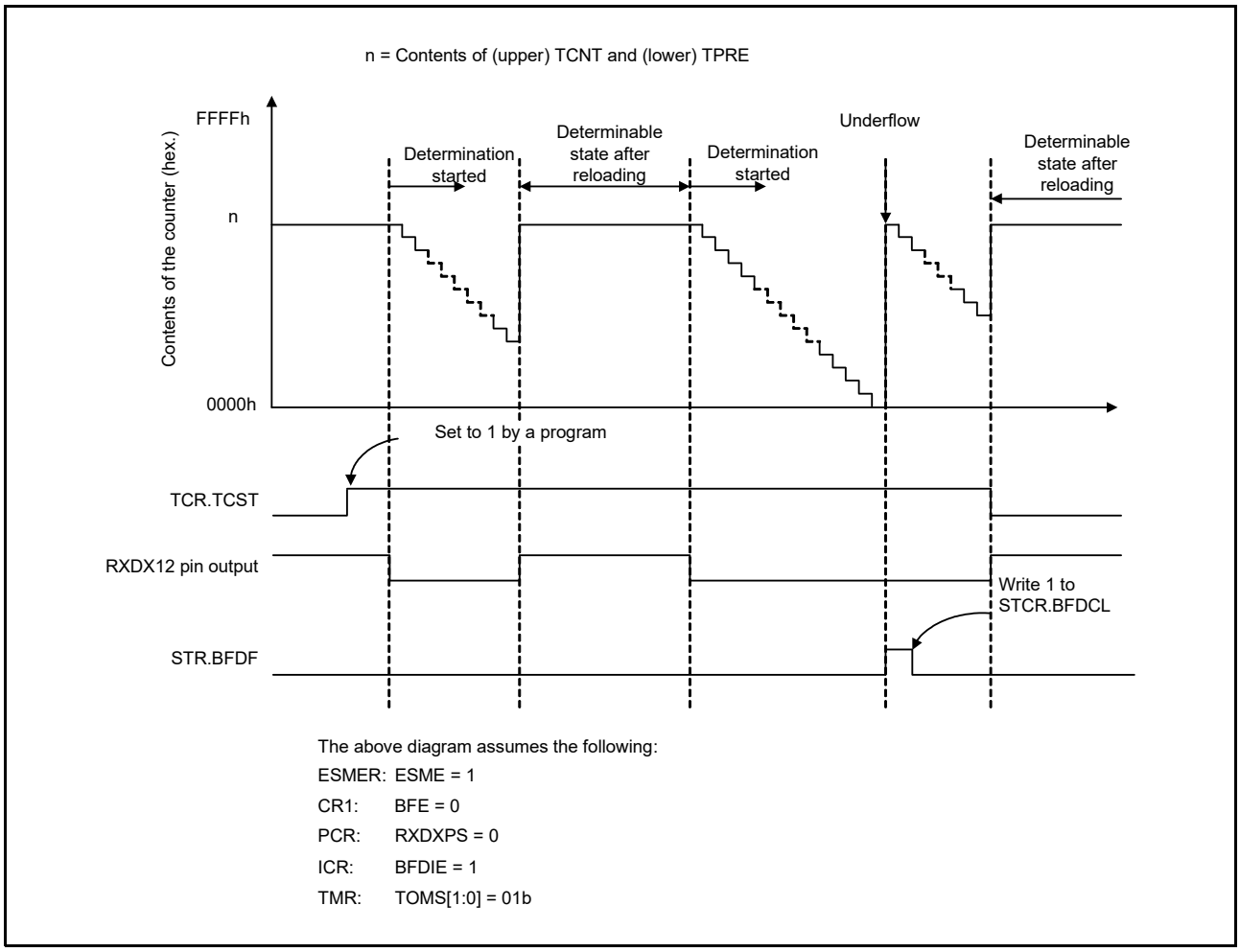


Figure 32.78 Example of Operations in Break Field Low Width Determination Mode

(3) Timer Mode

This mode is for counting cycles of the internal clock as the clock source. Setting the TMR.TOMS[1:0] bits to 00b switches operation to timer mode. The TMR.TCSS[2:0] bits select the clock source for the counter. Counting starts when 1 is written to the TCR.TCST bit and stops when 0 is written to the TCST bit. Registers TPRE and TCNT both count down. The TPRE register counts cycles of the clock source for counting, and underflows of the TPRE register provide the clock source for counting by the TCNT register. When the timer underflows, the STR.BFDF flag is set to 1. An SCIX0 interrupt is also generated if the value of the ICR.BFDIE bit is 1.

32.11 Noise Cancellation Function

Figure 32.79 shows the configuration of the noise filter used for noise cancellation. The noise filter consists of two stages of flip-flop circuits and a match-detection circuit. When the level on the pin matches in three consecutive samples taken at the set sampling interval, the matching level continues to be conveyed internally until the level on the pin again matches in three consecutive samples.

In asynchronous mode, the noise cancellation function can be applied on the RXDn input signal. The period of the base clock (1/16th of a bit-period when SEMR.ABCSE = 0 and SEMR.ABCS = 0, 1/8th of a bit-period when SEMR.ABCSE = 0 and SEMR.ABCS = 1, and 1/6th of a bit-period when SEMR.ABCSE = 1) is the sampling interval.

In simple I^2C mode, the noise cancellation function can be applied on the SSDAn and SSCLn input signals. The sampling clock is the clock signal produced by frequency-dividing the signal from the clock source for the internal baud-rate generator by one, two, four, or eight as selected by the setting of the SNFR.NFCS[2:0] bits.

If the base clock is stopped with the noise filter enabled and then the clock input is started again, the noise filter operation resumes from where the clock was stopped. If SCR.TE and SCR.RE are set to 0 during base clock input, all of the noise filter flip-flop values are initialized to 1. Accordingly, if the input data is 1 when reception operation resumes, it is determined that a level match is detected and is conveyed to the internal signal. When the level being input corresponds to 0, the initial output of the noise filter is retained until the level matches in three consecutive samples.

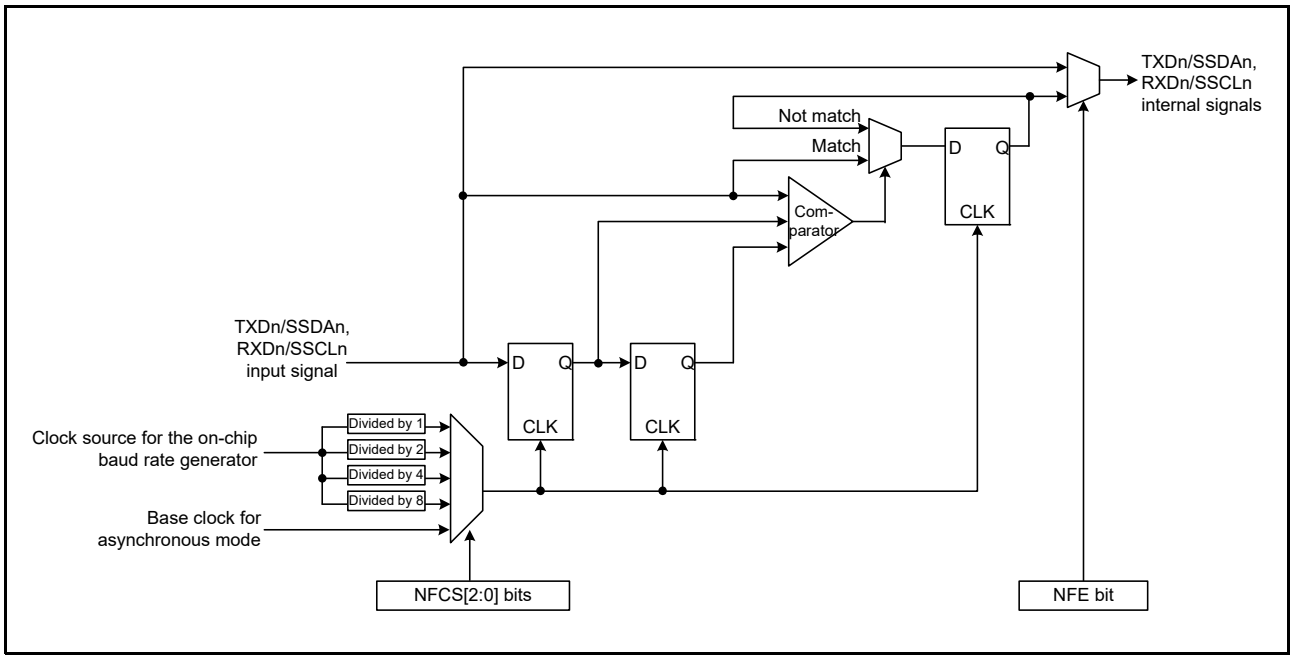


Figure 32.79 Block Diagram of Digital Noise Filter

32.12 Interrupt Sources

32.12.1 Buffer Operations for TXI and RXI Interrupts

If the conditions for a TXI and RXI interrupt are satisfied while the interrupt status flag in the interrupt controller is 1, the SCI does not output the interrupt request but retains it internally (with a capacity for retention of one request per source). When the value of the interrupt status flag in the interrupt controller becomes 0, the interrupt request retained within the SCI is output. The internally retained interrupt request is automatically discarded once the actual interrupt is output. Clearing of the corresponding interrupt enable bit (the TIE or RIE bit in the SCR register) can also be used to discard an internally retained interrupt request.

32.12.2 Interrupts in Asynchronous Mode, Clock Synchronous Mode, and Simple SPI Mode

Table 32.37 lists interrupt sources in asynchronous mode, clock synchronous mode, and simple SPI mode. Individual interrupt sources can be enabled or disabled with the enable bits in the SCR register.

If the SCR.TIE bit is 1, a TXI interrupt request is generated when transmit data is transferred from the TDR or TDRL register*1 to the TSR register. A TXI interrupt request can also be generated by setting the SCR.TE bit to 1 after setting the SCR.TIE bit to 1 or by using a single instruction to set the SCR.TE and SCR.TIE bit to 1 at the same time. A TXI interrupt request can activate the DTC or DMAC to handle data transfer.

A TXI interrupt request is not generated by setting the SCR.TE bit to 1 while the setting of the SCR.TIE bit is 0 or by setting the SCR.TIE bit to 1 while the setting of the SCR.TE bit is 1.*2

Note that setting the SCR.TE bit to 0 while the SCR.TIE bit is 1 leads to the generation of a TXI interrupt.

When new data is not written by the time of transmission of the last bit of the current transmit data and the setting of the SCR.TEIE bit is 1, the SSR.TEND flag becomes 1 and a TEI interrupt request is generated. Furthermore, when the setting of the SCR.TE bit is 1, the SSR.TEND flag retains the value 1 until further transmit data are written to the TDR or TDRL register*1, and setting the SCR.TEIE bit to 1 leads to the generation of a TEI interrupt request.

Writing data to the TDR or TDRL register*1 leads to clearing of the SSR.TEND flag and, after a certain time, discarding of the TEI interrupt request.

If the SCR.RIE bit is 1, an RXI interrupt request is generated when received data is stored in the RDR register. An RXI interrupt request can activate the DTC or DMAC to handle data transfer.

Setting of any from among the ORER, FER, and PER flags in the SSR register to 1 while the SCR.RIE bit is 1 leads to the generation of an ERI interrupt request. An RXI interrupt request is not generated at this time. Clearing all three flags (ORER, FER, and PER) leads to discarding of the ERI interrupt request.

Note 1. In the case where asynchronous mode and 9-bit data length are selected

Note 2. To temporarily disable TXI interrupts at the time of transmission of the last of the data and so on when you wish a new round of transmission to start after handling of the transmission-completed interrupt, control disabling and enabling of the interrupt by using the interrupt request enable bit in the interrupt controller rather than using the SCR.TIE bit. This can prevent the suppression of TXI interrupt requests in the transfer of new data.

Table 32.37 Interrupt Sources

Name	Interrupt Source	Interrupt Flag	DTC Activation	DMAC Activation
ERI	Receive error	ORER, FER, PER, DFER*1, or DPER*1	Not possible	Not possible
RXI	Receive data full	RDRF	Possible	Possible
	Data match*1	DCMF*1		
TXI	Transmit data empty	TDRE	Possible	Possible
TEI	Transmit end	TEND	Not possible	Not possible

Note 1. Available in SCI1, SCI5, and SCI6 only.

32.12.3 Interrupts in Smart Card Interface Mode

Table 32.38 lists interrupt sources in smart card interface mode. A transmit end interrupt (TEI) request cannot be used in this mode.

Table 32.38 SCI Interrupt Sources

Name	Interrupt Source	Interrupt Flag	DTC Activation	DMAC Activation
ERI	Receive error or error signal detection	ORER, PER, or ERS	Not possible	Not possible
RXI	Receive data full	—	Possible	Possible
TXI	Transmit data empty	TEND	Possible	Possible

Data transmission/reception using the DTC or DMAC is also possible in smart card interface mode, similar to in the normal SCI mode. In transmission, when the SSR.TEND flag is set to 1, a TXI interrupt request is generated. This TXI interrupt request activates the DTC or DMAC allowing transfer of transmit data if the TXI request is specified beforehand as a source of DTC or DMAC activation. The TEND flag is automatically set to 0 when the DTC or DMAC transfers the data.

If an error occurs, the SCI automatically retransmits the same data. During the retransmission, the TEND flag is kept to 0 and the DTC or DMAC is not activated. Therefore, the SCI and DTC or DMAC automatically transmit the specified number of bytes, including retransmission in the case of error occurrence. However, the SSR.ERS flag is not automatically cleared to 0 at error occurrence. Therefore, the ERS flag must be cleared by previously setting the SCR.RIE bit to 1 to enable an ERI interrupt request to be generated at error occurrence.

When transmitting/receiving data using the DTC or DMAC, be sure to make settings to enable the DTC or DMAC before making SCI settings. For DTC or DMAC settings, refer to section 18, Data Transfer Controller (DTCb) and section 17, DMA Controller (DMACAa).

In reception, an RXI interrupt request is generated when receive data is set to the RDR register. This RXI interrupt request activates the DTC or DMAC allowing transfer of receive data if the RXI request is specified beforehand as a source of DTC or DMAC activation. If an error occurs, the error flag is set. Therefore, the DTC or DMAC is not activated and an ERI interrupt request is issued to the CPU instead; the error flag must be cleared.

32.12.4 Interrupts in Simple I2C Mode

The interrupt sources in simple I2C mode are listed in Table 32.39. The STI interrupt is allocated to the transmit end interrupt (TEI) request. The receive error interrupt (ERI) request cannot be used.

The DTC or DMAC can also be used to handle transfer in simple I2C mode.

When the value of the SIMR2.IICINTM bit is 1, an RXI request will be generated on the falling edge of the SSCLn signal for the eighth bit. If the RXI has been set up as an activating request for the DTC or DMAC beforehand, the RXI request will activate the DTC or DMAC to handle transfer of the received data. Furthermore, a TXI request is generated on the falling edge of the SSCLn signal for the ninth bit (acknowledge bit). If the TXI has been set up as an activating request for the DTC or DMAC beforehand, the TXI request will activate the DTC or DMAC to handle transfer of the transmit data.

When the value of the SIMR2.IICINTM bit is 0, an RXI request (ACK detection) if the input on the SSDAn pin is at the low level or a TXI request (NACK detection) if the input on the SSDAn pin is at the high level will be generated on the rising edge of the SSCLn signal for the ninth bit (acknowledge bit). If the RXI has been set up as an activating request for the DTC or DMAC beforehand, the RXI request will activate the DTC or DMAC to handle transfer of the received data. Also, if the DTC or DMAC is used for data transfer in reception or transmission, be sure to set up and enable the DTC or DMAC before setting up the SCI.

When the IICSTAREQ, IICRSTAREQ, and IICSTPREQ bits in the SIMR3 register are used to generate a start condition, restart condition, or stop condition, the STI request is issued when generation is complete.

Table 32.39 SCI Interrupt Sources

Name	Interrupt Source		Interrupt Flag	DTC Activation	DMAC Activation
	IICINTM bit = 0	IICINTM bit = 1			
RXI	ACK detection	Reception	—	Possible	Possible
TXI	NACK detection	Transmission	—	Possible*1	Possible*1
STI	Completion of generation of a start, restart, or stop condition		IICSTIF	Not possible	Not possible

Note 1. Activation of the DTC or DMAC is only possible when the SIMR2.IICINTM bit is 1 (use reception and transmission interrupts).

32.12.5 Interrupt Requests from the Extended Serial Mode Control Section

The extended serial mode control section has a total of six types of interrupt request for generating the SCIX0 interrupt (Break Field low width detected), SCIX1 interrupt (Control Field 0 match, Control Field 1 match, priority interrupt bit detected), SCIX2 interrupt (bus collision detected), and SCIX3 interrupt (valid edge detected). When any of the interrupt factors is generated, the corresponding status flag is set to 1. Details of all of the interrupt requests are listed in Table 32.40.

Table 32.40 Interrupt Sources of the Extended Serial Mode Control Section

Interrupt Request	Status Flag	Interrupt Factors
SCIX0 interrupt (Break Field low width detected)	BFDF	• Detection of a Break Field low width longer than the interval corresponding to the timer setting • Completion of the output of a Break Field low width over the interval corresponding to the timer setting • Underflow of the timer
SCIX1 interrupt (Control Field 0 match)	CF0MF	The data received in Control Field 0 matching the value set in the CF0DR register
SCIX1 interrupt (Control Field 1 match)	CF1MF	The data received in Control Field 1 matching the value set in the PCF1DR or SCF1DR register
SCIX1 interrupt (priority interrupt bit detected)	PIBDF	The value of the bit specified as the priority interrupt bit matching the value set in the PCF1DR register
SCIX2 interrupt (bus collision detected)	BCDF	The output level on the TXDX12 pin and the input level on the RXDX12 pin not matching on three consecutive cycles of the bus collision detection clock
SCIX3 interrupt (valid edge detected)	AEDF	Detection of a valid edge during bit rate measurement

32.13 Event Linking

By employing interrupt request signals as event signals, SCI5 is able to provide linked operation through the event link controller (ELC) for modules selected in advance.
Event signals can be output regardless of the values of the corresponding interrupt request enable bits.

(1) Error (receive error, error signal detected) event output

- Indicates abnormal termination due to a parity error during reception in asynchronous mode.
- Indicates abnormal termination due to a framing error during reception in asynchronous mode.
- Indicates abnormal termination due to an overrun error during reception.
- Indicates detection of the error signal during transmission in smart card interface mode.

(2) Receive data full event output

- Indicates that received data have been set in the receive data register (RDR or RDRL).
- Indicates that ACK has been detected if the SIMR2.IICINTM bit is 0 in simple I^2C mode.
- Indicates that the 8th-bit SSCL5 falling edge has been detected if the SIMR2.IICINTM bit is 1 in simple I^2C mode.
- When the SIMR2.IICINTM bit is 1 during master transmission in simple I^2C mode, set the event link controller (ELC) so that receive data full events are not used.

(3) Transmit data empty event output

- Indicates that the SCR.TE bit has been changed from 0 to 1.
- Indicates that transmit data have been transferred from the transmit data register (TDR or TDRL) to the transmit shift register (TSR).
- Indicates that transmission has been completed in smart card interface mode.
- Indicates that NACK has been detected if the SIMR2.IICINTM bit is 0 in simple I^2C mode.
- Indicates that the ninth-bit SSCL5 falling edge has been detected if the SIMR2.IICINTM bit is 1 in simple I^2C mode.

(4) Transmit end event output

- Indicates the completion of transmission.
- Indicates that the starting condition, resumption condition, or termination condition has been generated in simple I^2C mode.

32.14 Usage Notes

32.14.1 Setting the Module Stop Function

Module stop control register B (MSTPCRB) is used to stop and start SCI operations. With the value after a reset, SCI operations are stopped. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

32.14.2 Break Detection and Processing

When a framing error is detected, a break can be detected by reading the RXDn pin value directly or reading the value of the SPTR.RXDMON flag (only for SCI1, SCI5, and SCI6). In a break, the input from the RXDn pin becomes all 0s, and so the SSR.FER flag is set to 1 (framing error has occurred), and the SSR.PER flag may also be set to 1 (parity error has occurred). When the SEMR.RXDESEL bit is 0, the SCI continues the receive operation even after a break is received. Therefore, note that even if the FER flag is set to 0 (no framing error occurred), it will be set to 1 again. When the SEMR.RXDESEL bit is 1, the SCI sets the SSR.FER flag to 1 and stops receiving operation until a start bit of the next data frame is detected. If the SSR.FER flag is set to 0 at this time, the SSR.FER flag retains 0 during the break. When the RXDn pin becomes high and the break ends, detecting the beginning of the start bit at the first falling edge of the RXDn pin allows the SCI to start the receiving operation.

32.14.3 Mark State and Sending Breaks

When the SCR.TE bit is 0 (serial transmission is disabled), the TXDn pin becomes high-impedance. To forcibly set the TXDn pin to mark or space state while the TE bit is 0, set the I/O port associated registers and switch the TXDn pin to general output port.

For holding the communication line in the mark (“1”) state until the TE bit is set to 1 (serial transmission is enabled), set the corresponding bit in the PODR register to 1 for high output from general output port. To start communications, set the TE bit to 1 and then the corresponding bit in the PMR register to 1.

To send a break (the space state for longer than a certain period of time) while data transmission, set the corresponding bit in the PODR register to 0 (low output), and set the corresponding bit in the PMR register to 0 (general I/O port). Then set the TE bit to 0 if necessary. When the TE bit is set to 0, the transmitter is initialized regardless of the current transmit status.

The SPTR register, if it is included, can set the TXDn pin in mark/space state without switching the pin function to general output port. Set the SPTR.SPB2IO bit to 1 (output) and the SPB2DT bit to 1 (mark) or 0 (space), and then set the TE bit to 0.

32.14.4 Receive Error Flags and Transmit Operations (Clock Synchronous Mode and Simple SPI Mode)

Transmission cannot be started when a receive error flag (SSR.ORER) is set to 1, even if data is written to the TDR register. Be sure to set the receive error flags to 0 before starting transmission. Note also that the receive error flags cannot be set to 0 even if the SCR.RE bit is set to 0 (serial reception is disabled).

32.14.5 Writing Data to the TDR Register

Data can be written to registers TDR, TDRH, and TDRL. However, if new data is written to registers TDR, TDRH, and TDRL when transmit data is remaining in registers TDR, TDRH, and TDRL, the previous data in registers TDR, TDRH, and TDRL is lost because it has not been transferred to the TSR register yet. Be sure to write transmit data to registers TDR, TDRH, and TDRL in the TXI interrupt request handling routine.

32.14.6 Restrictions on Clock Synchronous Transmission (Clock Synchronous Mode and Simple SPI Mode)

When the external clock source is used as a synchronization clock, the following restrictions apply.

(1) Start of transmission

Update the TDR register by the CPU, DMAC, or DTC and wait for at least five PCLK cycles before allowing the transmit clock to be input (refer to Figure 32.80).

(2) Continuous transmission

(a) Write the next transmit data to the TDR or TDRL register before the falling edge of the transmit clock (bit 7) (refer to Figure 32.80).

(b) When updating the TDR register after bit 7 has started to transmit, update the TDR register while the synchronization clock is in the low-level period, and set the high-level width of the transmit clock (bit 7) to four PCLK cycles or longer (refer to Figure 32.80).

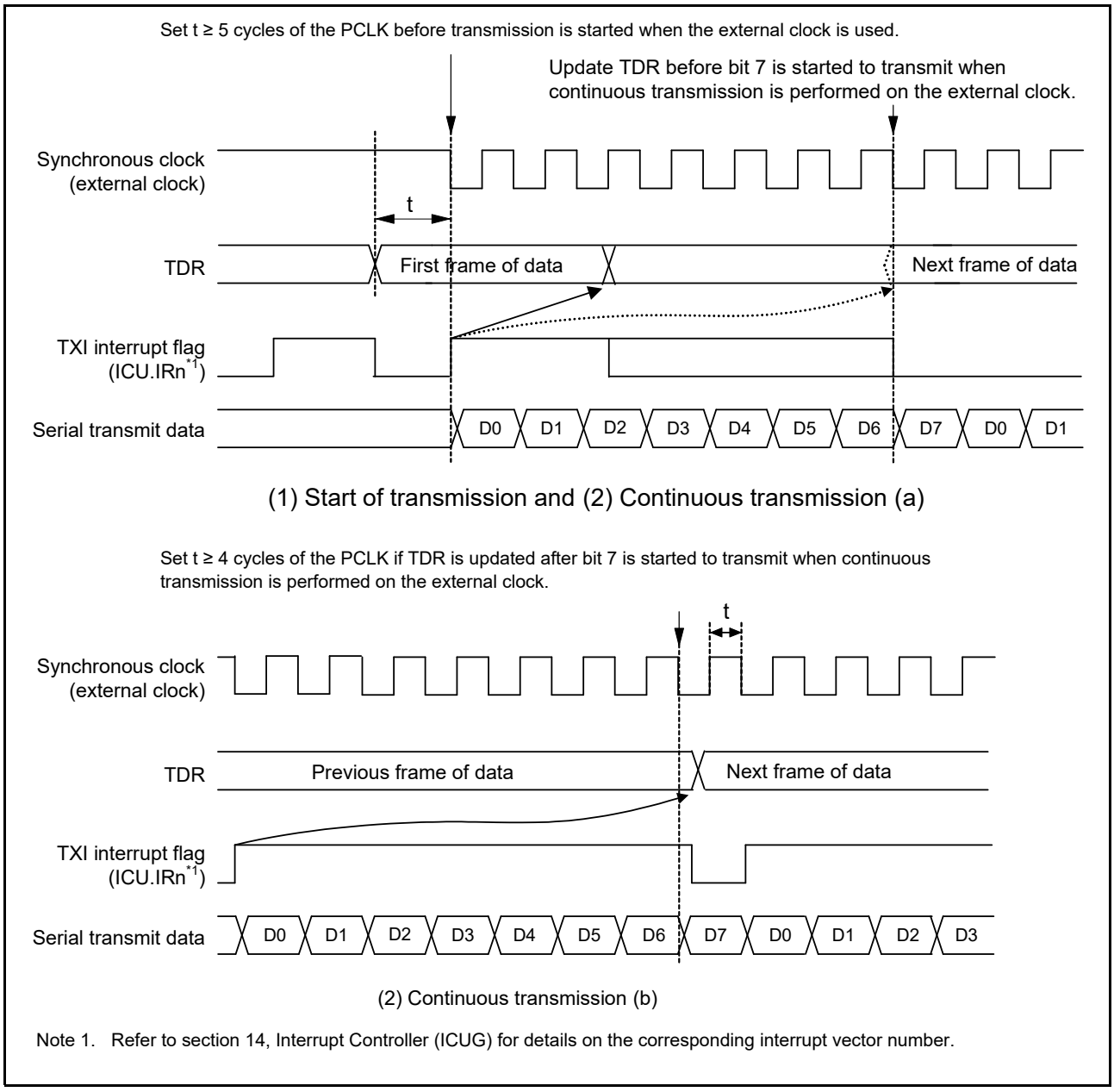


Figure 32.80 Restrictions on Use of External Clock in Clock Synchronous Transmission

32.14.7 Restrictions on Using DMAC or DTC

When using the DMAC or DTC to read the RDR, RDRH, and RDRL registers, be sure to set the receive data full interrupt (RXI) as the activation source of the relevant SCI.

32.14.8 Notes on Starting Transfer

At the point where transfer starts when the interrupt status flag (IRn.IR flag) in the interrupt controller is 1, follow the procedure below to clear interrupt requests before permitting operations (by setting the SCR.TE or SCR.RE bit to 1). For details on the interrupt status flag, refer to section 14, Interrupt Controller (ICUG).

- Confirm that transfer has stopped (the setting of the SCR.TE or SCR.RE bits is 0).
- Set the corresponding interrupt enable bit (SCR.TIE or SCR.RIE) to 0.
- Read the corresponding interrupt enable bit (SCR.TIE or SCR.RIE bit) to check that it has become 0.
- Set the interrupt status flag (IRn.IR flag) in the interrupt controller to 0.

32.14.9 SCI Operations during Low Power Consumption State

(1) Transmission

When making settings for the module stopped state or in transitions to software standby, stop operations (by setting the TIE, TE, and TEIE bits in the SCR register to 0) after switching the TXDn pin to the general I/O port pin function or fix the output level of the TXDn pin by the SPTR register (only for SCI1, SCI5, and SCI6). Setting the TE bit to 0 resets the TSR register and the SSR.TEND flag. Depending on the port settings or SPTR register setting (only for SCI1, SCI5, and SCI6), output pins may output the level before a transition to the low power consumption state is made after release from the module stopped state or software standby mode. When transitions to these states are made during transmission, the data being transmitted become indeterminate.

To transmit data in the same transmit mode after cancellation of the low power consumption state, set the TE bit to 1, read the SSR register, and write data to the TDR register sequentially to start data transmission. To transmit data with a different transmit mode, initialize the SCI first.

Figure 32.81 shows a sample flowchart for transition to software standby mode during transmission. Figure 32.82 and Figure 32.83 show the port pin states during transition to software standby mode.

Before specifying the module stop state or making a transition to software standby mode from the transmit mode using DTC/DMA transfer, stop the transmit operations (TE = 0). To start transmission after cancellation using the DTC/DMAC, set the TE and TIE bits to 1. The TXI interrupt flag is set to 1 and transmission starts using the DTC/DMAC.

(2) Reception

Before specifying the module stop state or making a transition to software standby mode, stop the receive operations (SCR.RE = 0). If transition is made during data reception, the data being received will be invalid.

To receive data in the same receive mode after cancellation of the low power consumption state, set the RE bit to 1, and then start reception. To receive data in a different receive mode, initialize the SCI first.

Figure 32.84 shows a sample flowchart for transition to software standby mode during reception.

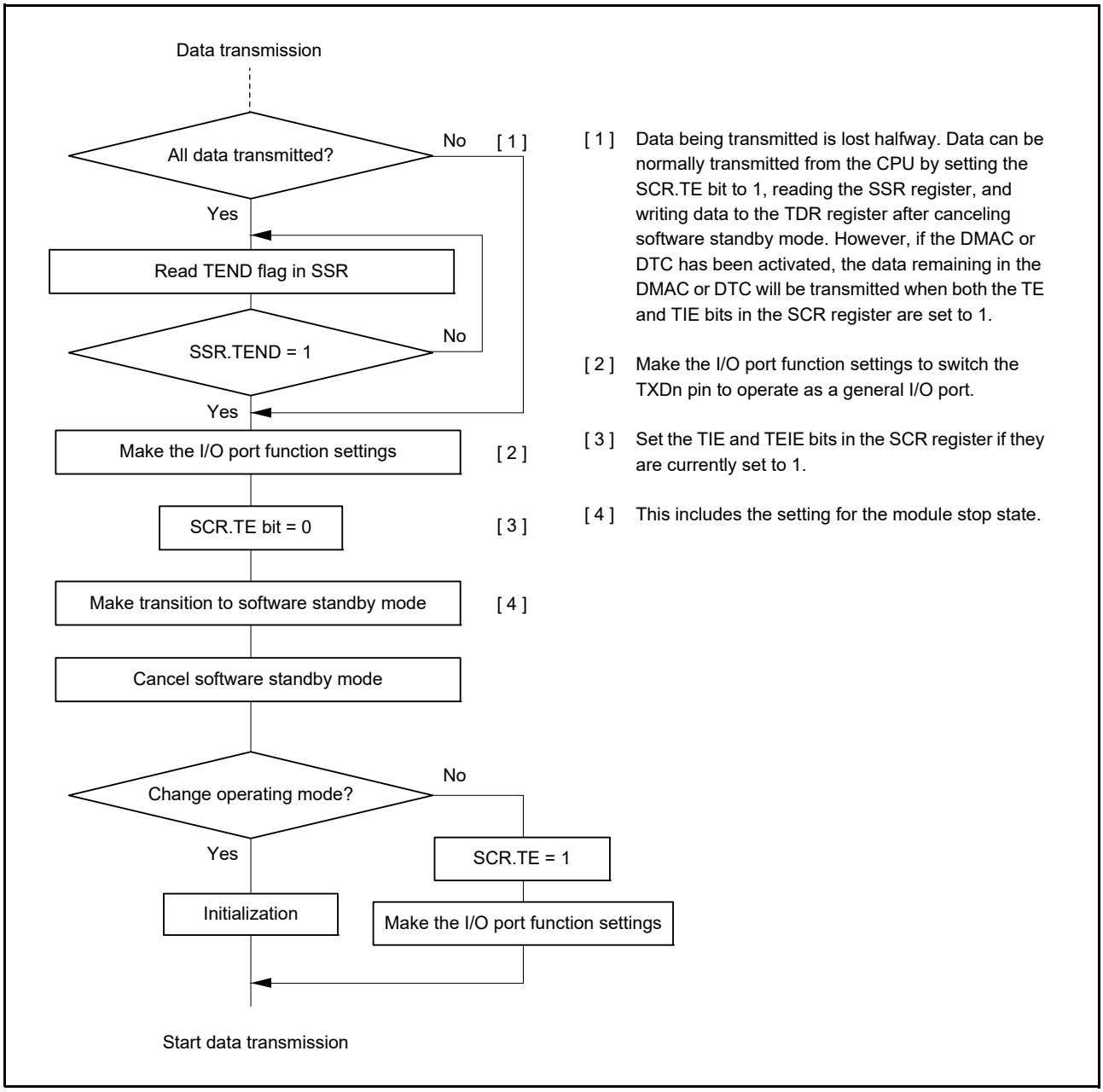


Figure 32.81 Example of Flowchart for Transition to Software Standby Mode during Transmission

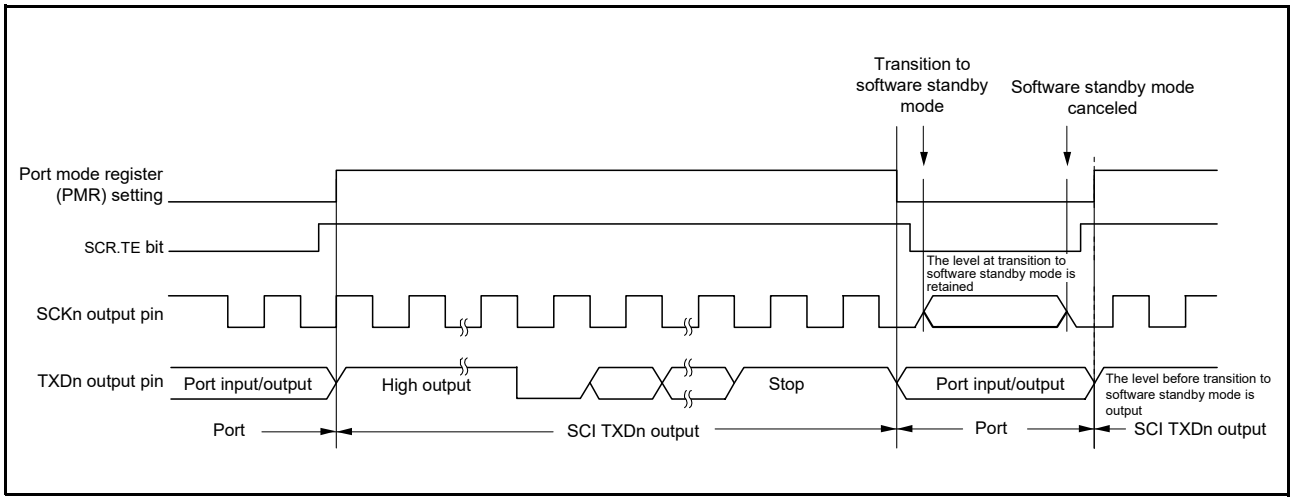


Figure 32.82 Port Pin States during Transition to Software Standby Mode (Internal Clock, Asynchronous Transmission)

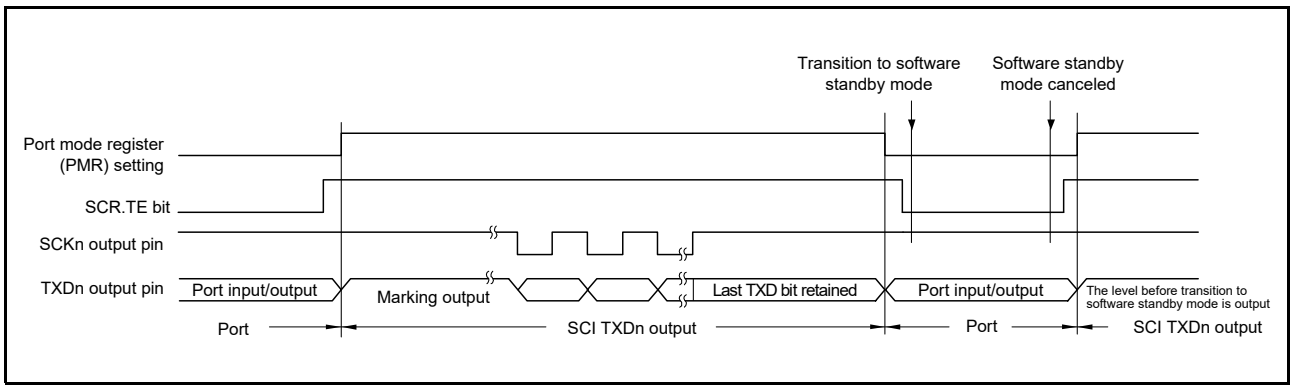


Figure 32.83 Port Pin States during Transition to Software Standby Mode (Internal Clock, Clock Synchronous Transmission)

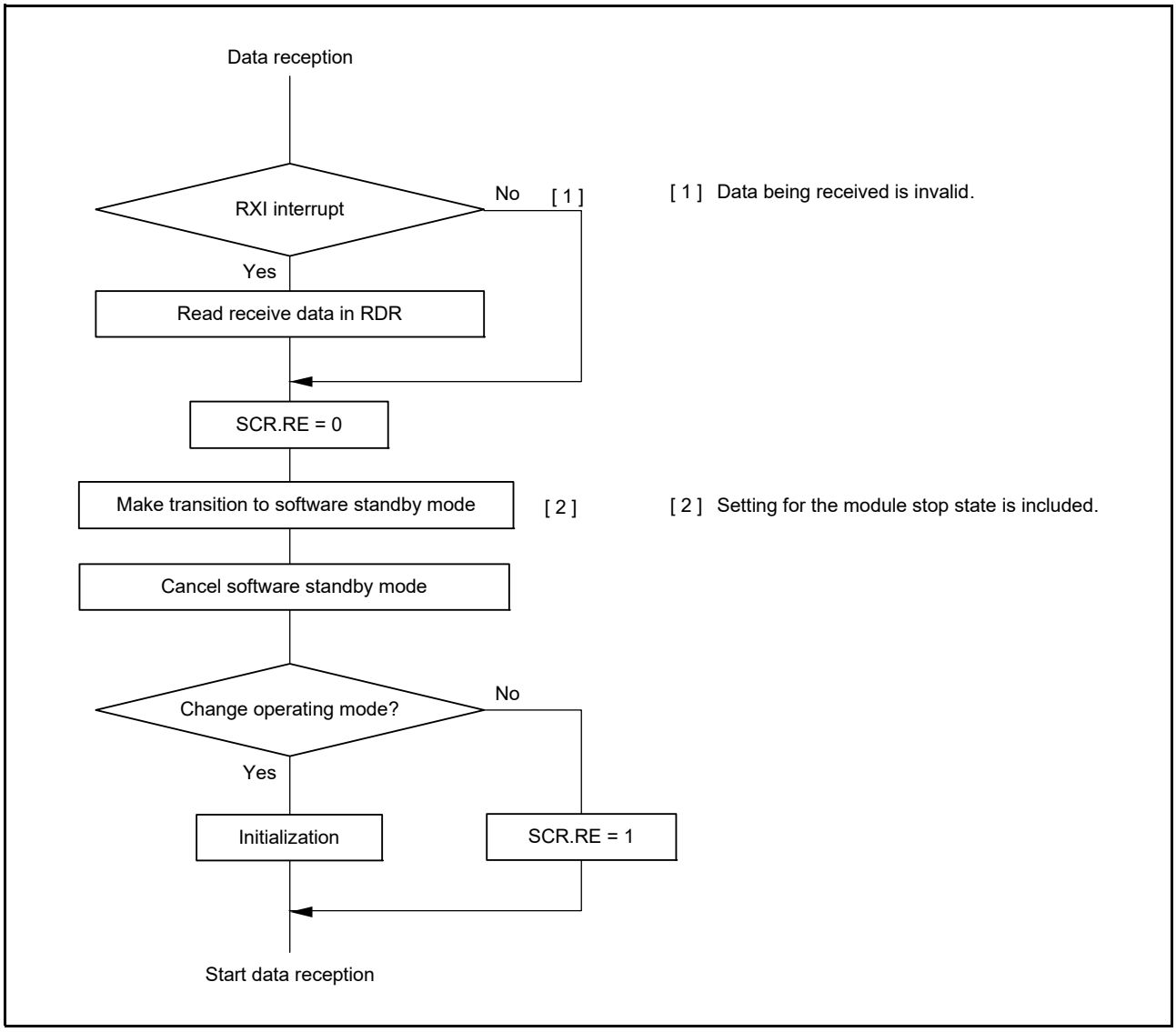


Figure 32.84 Example of Flowchart for Transition to Software Standby Mode during Reception

32.14.10 External Clock Input in Clock Synchronous Mode and Simple SPI Mode

In clock synchronous mode and simple SPI mode, the external clock SCKn must be input as follows:
High-pulse period, low-pulse period = 2 PCLK cycles or more, period = 6 PCLK cycles or more

32.14.11 Limitations on Simple SPI Mode

(1) Master Mode

- Use a resistor to pull up or pull down the clock line matching the initial settings for the transfer clock set by the SPMR.CKPH and CKPOL bits when the SPMR.SSE bit is 1.
  This prevents the clock line from being placed in the high-impedance state when the SCR.TE bit is set to 0 or unexpected edges from being generated on the clock line when the SCR.TE bit is changed from 0 to 1. When the SPMR.SSE bit is 0 in single master mode, pulling up or pulling down the clock line is not necessary because the clock line is not placed in the high-impedance state even when the SCR.TE bit is set to 0.
- In the case of the setting for clock delay (SPMR.CKPH bit is 1), the receive data full interrupt (RXI) is generated before the final clock edge on the SCKn pin as indicated in Figure 32.85. If the TE and RE bits in the SCR register become 0 at this time before the final edge of the clock signal on the SCKn pin, the SCKn pin is placed in the high-impedance state, so the width of the last clock pulse of the transfer clock is shortened. Furthermore, an RXI interrupt may lead to the input signal on the SSn# pin of a connected slave going to the high level before the final edge of the clock signal on the SCKn pin, leading to incorrect operation of the slave.
- In a multi-master configuration, take care because the SCKn pin output becomes high-impedance while the input on the SSn# pin is at the low level if a mode fault error occurs as the current character is being transferred, stopping supply of the clock signal to the connected slave. Remake the settings for the connected slave to avoid misaligned bits when transfer is restarted.

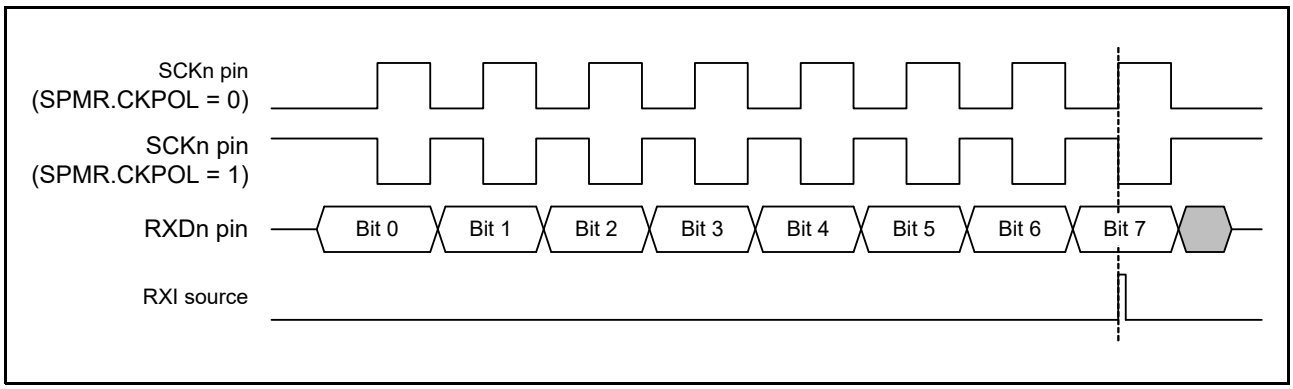


Figure 32.85 Timing of the RXI Interrupt in Simple SPI Mode (with Clock Delay)

(2) Slave Mode

- Secure at least five cycles of the PCLK from writing transmit data in the TDR register to start of the external clock input. Also secure at least five cycles of the PCLK from input of low level on the SSn# pin to start of the external clock input.
- Provide an external clock signal from the master the same as the transmit/receive data length.
- Control the input on the SSn# pin before the start and after the end of data transfer.
- When the level being input on the SSn# pin is to be changed from low to high while the current character is being transferred, set the TE and RE bits in the SCR register to 0 and, after remaking the settings, restart transfer of the first byte.

32.14.12 Limitation 1 on Usage of the Extended Serial Mode Control Section

When the PCR.SHARPS bit is set to 1, output on the TXDX12/RXDX12 pin is only possible when the following conditions apply.

- The timer is in Break Field low width output mode and the value of the TCR.TCST bit is 1 (when the TCST bit is set to 1, the high level continues to be output for up to one cycle of the clock source for counting by the timer counter before output of the low level)
- The value of the SCR.TE bit is 1.

32.14.13 Limitation 2 on Usage of the Extended Serial Mode Control Section

The TXI, RXI, ERI, and TEI interrupt requests are generated even if the extended serial mode is enabled. However, the RXI interrupt should not be enabled during reception of a Start Frame because the extended serial mode control section uses the receive data full signal.
To use the RXI interrupts during a reception of the Information Frame, use it in accordance with one of the following procedures. When a receive error is detected, clear the receive error flag and initialize the extended serial mode control section.

(1) Set the SCR.RIE bit to 0 to disable the output of interrupt requests. Check the error flags in the SSR register on completion of the reception of a Start Frame, because an ERI interrupt is not generated if a receive error occurs. After reception of the Start Frame is completed, set the SCR.RIE bit to 1 by the time the first byte of the Information Frame is received.
(2) Set the SCR.RIE bit to 1 to disable RXI interrupts and enable ERI interrupts for ICU.
Clear the IRn.IR flag to enable the acceptance of RXI interrupts by ICU by the time the first byte of the Information Frame is received after the completion of Start Frame reception.

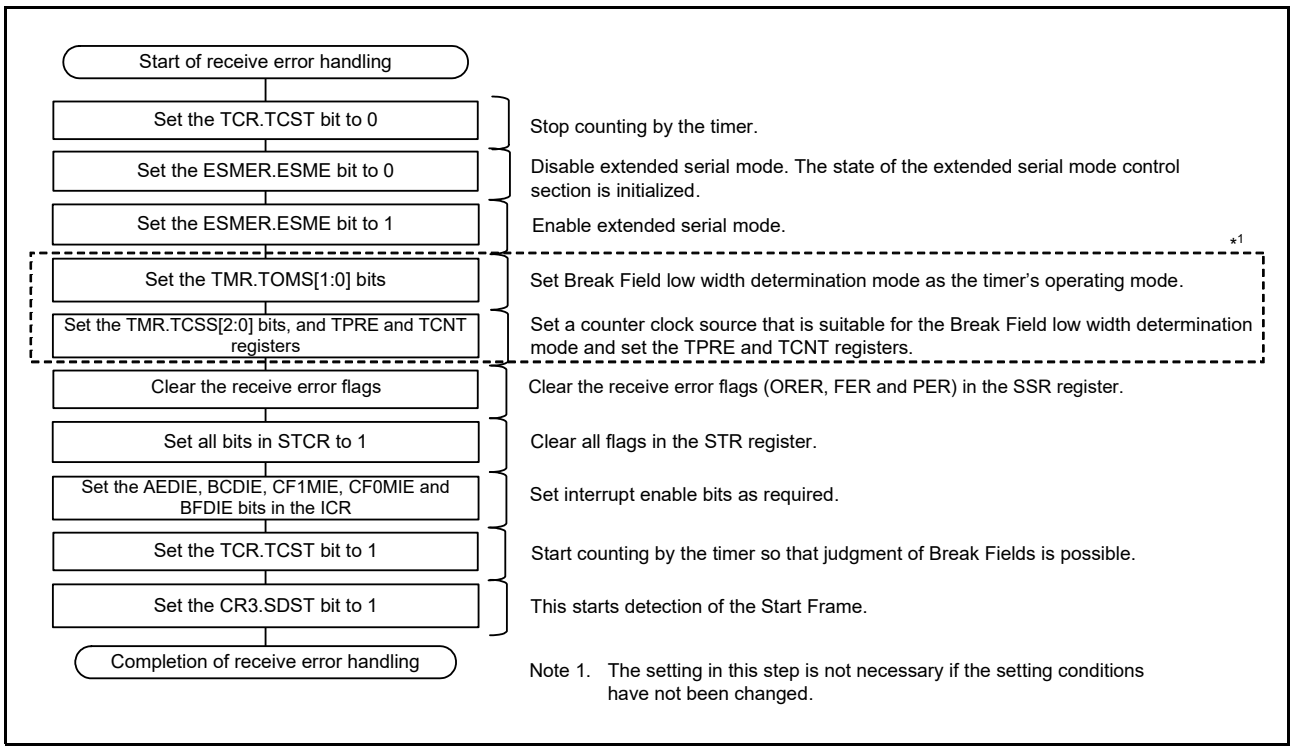


Figure 32.86 Example of Flowchart for Receive Error Handling (during Reception of the Start Frame)

32.14.14 Note on Transmit Enable Bit (TE Bit)

When setting the pin function to “TXDn” while the SCR.TE bit is 0 (serial transmission is disabled) or setting the TE bit to 0 while the pin function is “TXDn”, output of the TXDn pin becomes high-impedance.
Prevent the TXDn line from becoming high-impedance by any of the following ways:

(1) Connect a pull-up or pull-down resistor to the TXDn line.
(2) Set the TE bit to 1*1 before changing the pin function to “TXDn”. Before setting the TE bit to 0, change the pin function to “general-purpose I/O port” and drive the pin high or low.
(3) Set the SPTR.SPB2IO bit to 1 first, and change the pin function to “TXDn”. Leave the value of the SPB2IO bit as 1 after that (for SCI1, SCI5, and SCI6).

Note 1. An interrupt is generated when the TE bit is set to 1 while the TXI interrupt is enabled. If this creates a problem, change the pin function to “TXDn” first, and then set the corresponding ICU.IERm.IENj bit to 1.

32.14.15 Note on Stopping Reception When Using the RTS Function in Asynchronous Mode

One clock cycle of PCLK is required for the time from setting the SCR.RE bit to 0 to stopping the RTS signal generator in asynchronous mode.
When reading the RDR (or RDRL) register after setting the SCR.RE bit to 0, confirm that the RE bit has been set to 0 before reading the RDR (or RDRL) register to prevent these two processes from being performed consecutively.

33. Serial Communications Interface (RSCI)

In this section, “PCLK” for RSCI8 and RSCI9 is used to refer to PCLKB and “PCLK” for RSCI11 is used to refer to PCLKA.

33.1 Overview

RSCI can handle both asynchronous and clock synchronous serial communications. RSCI has 32-stage FIFO buffers in transmission/reception blocks, and it can select the FIFO composition, and it can transmit/receive efficiently, and it can also communicate continuously.

Asynchronous serial data communications can be carried out with standard asynchronous communications chips such as a Universal Asynchronous Receiver/Transmitter (UART) or Asynchronous Communications Interface Adapter (ACIA). As an extended function in asynchronous communications mode, the RSCI also supports smart card (IC card) interfaces conforming to ISO/IEC 7816-3 (standard for identification cards).

The RSCI is also supports serial communication using manchester code (manchester mode), simple SPI interfaces, simple I^2C-bus interfaces (single master), and extended serial communication.

In addition, asynchronous mode has a support function for generating AMI waveform of negative logic coding with 50% duty cycle used in home bus system (HBS) communications.

Table 33.1 lists the RSCI specifications and Table 33.2 lists the functions of each channel.

Table 33.1 RSCI Specifications (1/3)

Item	Description
Serial communication modes	• Asynchronous • Manchester • Clock synchronous • Smart card interface • Simple I^2C-bus • Simple SPI bus • Extended serial
Transfer speed	Bit rate specifiable with the on-chip baud rate generator.
Full-duplex communications	Transmitter: Continuous transmission possible using double-buffer structure. Receiver: Continuous reception possible using double-buffer structure.
Half-duplex communications	Half-duplex communication is possible by using only TXDn pins
Data transfer	Selectable as LSB first or MSB first transfer
I/O signal level inverting function	Input signal and output signal can be inverted independently.
RXD input signal select function	If the RXD signal is attenuated by transmission line effects, it can be improved by using a comparator instead of a receiver.
Interrupt sources	Transmit end, transmit data empty, receive data full, receive error, receive data ready, and receive data match Break Field detection/transmission, Bus collision detection, Active edge detection Completion of generation of a start condition, restart condition, or stop condition
RS-485 driver control function	Output DE signal to enable external transceiver transmit mode
Loopback function	Self-diagnosis of communication function is possible by connecting TXD and RXD inside the RSCI
Low power consumption function	Module stop state can be set for each channel.

Table 33.1 RSCI Specifications (2/3)

Item		Description
Asynchronous mode	Data length	7, 8, or 9 bits
	Transmission stop bit	1 or 2 bits
	Parity	Even parity, odd parity, or no parity
	Receive error detection	Parity, overrun, and framing errors
	Hardware flow control	CTSn# and RTSn# pins can be used in controlling transmission/reception.
	Transmission/Reception	Selectable either 1 stage register or 32 stage FIFO
	Data match detection	The interrupt request can issue by detecting the match between receive data and comparison data.
	Start-bit detection	Low level or falling edge is selectable.
	Receive data sampling timing adjustment	Sampling point of receive data can be changed to front or rear of center of bit.
	Transmit signal transition timing adjustment	Falling or rising edge of the transmit data can be delayed.
	Break detection	When a framing error occurs, a break can be detected by reading the SSR register.
	Clock source	An internal or external clock can be selected.
	Double-speed mode	Baud rate generator double-speed mode is selectable.
	Multi-processor communications function	Serial communication among multiple processors
	Noise cancellation	The signal paths from input on the RXDn pins incorporate digital noise filters.
	HBS support mode	Transmission/reception using inverted RZI (Return to Zero, Inverted) code is possible.
Manchester mode	Data length	7, 8, or 9 bits
	Transmission stop bit	1 or 2 bits
	Receive error detection	Parity, overrun, framing, manchester code errors, preface, start bit, and receive Sync
	Hardware flow control	CTSn# and RTSn# pins can be used in controlling transmission.
	Clock source	Only internal clock can be used. (The setting of external clock is prohibited because it is not the object of operation guarantee.)
	Double-speed mode	Baud rate generator double-speed mode is selectable.
	Multi-processor communications function	Serial communication among multiple processors
	Noise cancellation	The signal paths from input on the RXDn pins incorporate digital noise filters.
	Manchester encoding/ decoding function	Function to perform manchester encoding/decoding of transmission/reception data and communicate using manchester code
	Preface setting/detection function	Function to detect the beginning of a frame from the preface pattern. Preface pattern can be selected from 4 types. The length can also be changed from 0 to 15 bits.
	Start Bit setting/detection function	The Start Bit length can be set to 1 bit or 3 bits. In the case of 3-bit length, it is possible to judge the type of subsequent data with two types of patterns.
	Reception retiming function	Function to perform timing correction for each bit center edge by using manchester code having edge at bit center
Smart card interface mode	Error processing	An error signal can be automatically transmitted when detecting a parity error during reception
		Data can be automatically retransmitted when receiving an error signal during transmission
	Data type	Both direct convention and inverse convention are supported.

Table 33.1 RSCI Specifications (3/3)

Item		Description
Extended serial mode	Start Frame Transmission	Break Field transmission possible, Break Field transmission complete interrupt output possible Bus collision detection possible, bus collision detection interrupt output possible
	Start Frame Reception	Break Field detectable, Break Field detected interrupt output possible Control Field 0/1 data comparison function Control Field 1 can set two types of comparison data of primary and secondary Priority interrupt bit can be set in Control Field 1 Bit rate measurement function
Simple I2C mode	Transfer format	I2C-bus format
	Operating mode	Master (single-master operation only)
	Transfer rate	Up to 400 kbps
	Noise cancellation	The signal paths from input on the SSCLn and SSDAn pins incorporate digital noise filters, and the interval for noise cancellation is adjustable.
Clock synchronous mode	Data length	8 bits
	Adjustment of receive sampling timing	Adjustable receive sampling timing after the default timing in master mode only when using internal clock
	Receive error detection	Overrun error
	Clock source	An internal clock (master mode) or external clock (slave mode) can be selected.
	Double-speed mode	Baud rate generator double-speed mode is selectable
	Hardware flow control	CTSn# and RTSn# pins can be used in controlling transmission/reception.
	Transmission/Reception	Selectable either 1 stage register or 32 stage FIFO
Simple SPI bus	Data length	8 bits
	Detection of errors	Overrun error
	Clock source	An internal clock (master mode) or external clock (slave mode) can be selected.
	Double-speed mode	Baud rate generator double-speed mode is selectable
	Adjustment of receive sampling timing	Adjustable receive sampling timing after the default timing in master mode only when using internal clock
	SS input pin function	When the SSn # pin is high level, the output pin can be set to high impedance
	Adjustment of receive sampling timing	Four kinds of settings for clock phase and clock polarity are selectable.
	Transmission/Reception	Selectable either 1 stage register or 32 stage FIFO
Bit rate modulation function		Correction of outputs from the on-chip baud rate generator can reduce errors.
Event link function		Error (receive error or error signal detection) event output
		Receive data full event output
		Transmit data empty event output
		Transmit end event output

Table 33.2 Functions of Each Channel

Item	RSCI8	RSCI9	RSCI11
Asynchronous mode	Available	Available	Available
Manchester mode	Not available	Available	Available
Smart card interface mode	Available	Available	Available
Extended serial mode	Not available	Available	Available
Simple I^2C mode	Available	Available	Available
Clock synchronous mode	Available	Available	Available
Simple SPI mode	Available	Available	Available
FIFO buffer	Not available	Not available	Available
Event link function	Not available	Not available	Available
Peripheral module clock	PCLKB	PCLKB	PCLKA

Figure 33.1 shows RSCI Block Diagram (n = 010, 011) with all functions.

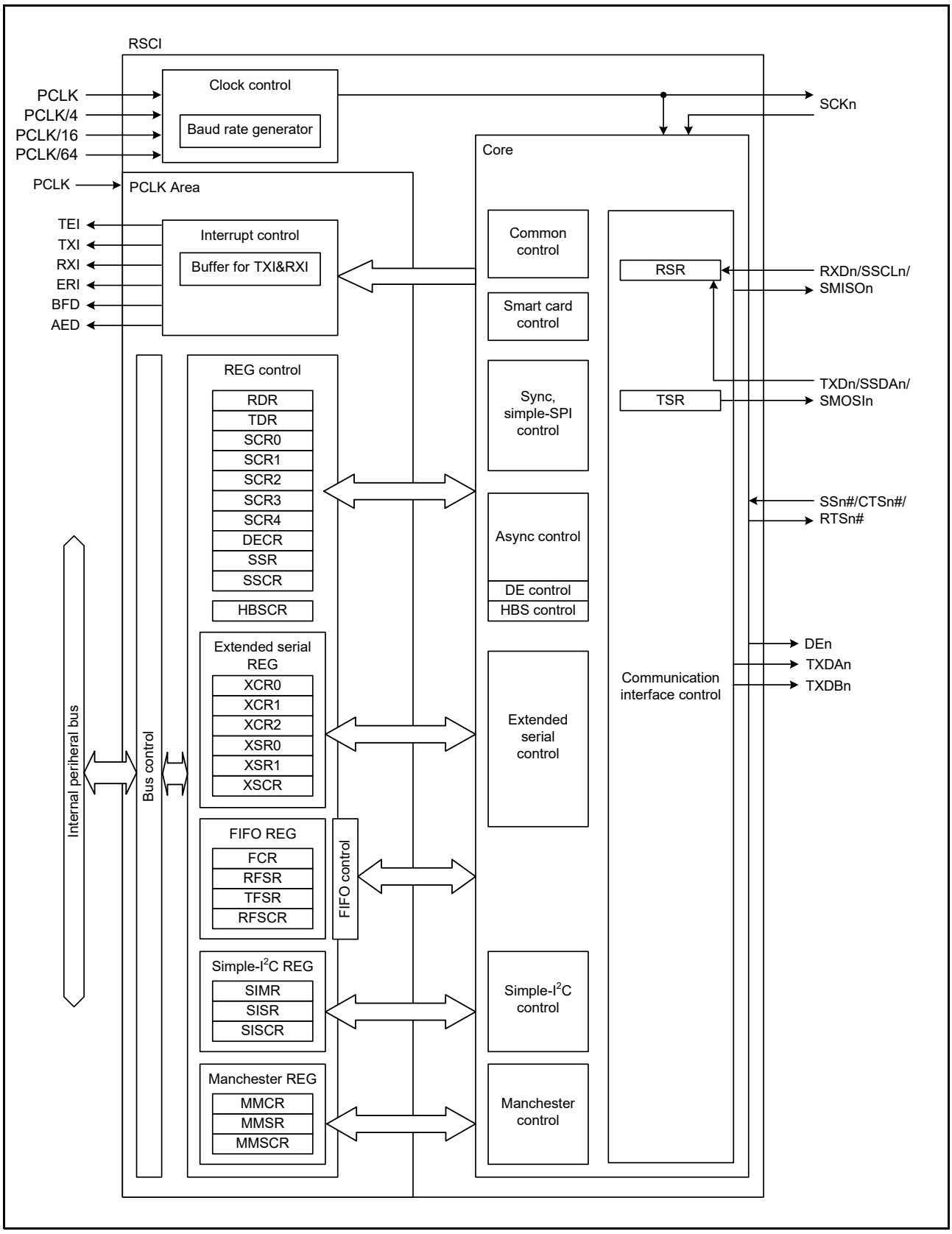


Figure 33.1 RSCI Block Diagram (n = 010, 011)

Table 33.3 to Table 33.6 list RSCI's input/output pins.

Table 33.3 RSCI Input/Output Pin (Asynchronous Mode/Clock Synchronous Mode/Manchester Mode/ Extended Serial Mode)

Channel	Pin Name	I/O	Function
RSCI8	SCK008	I/O	RSCI8 clock input/output
	RXD008	Input	RSCI8 receive data input
	TXD008	Output	RSCI8 transmit data output
	RTS008#	Output	RSCI8 request-to-send signal output
	CTS008#	Input	RSCI8 transmission start control input
	DE008	Output	RSCI8 RS-485 driver control output
RSCI9	SCK009	I/O	RSCI9 clock input/output
	RXD009	Input	RSCI9 receive data input
	TXD009	Output	RSCI9 transmit data output
	RTS009#	Output	RSCI9 request-to-send signal output
	CTS009#	Input	RSCI9 transmission start control input
	DE009	Output	RSCI9 RS-485 driver control output
RSCI11	SCK011	I/O	RSCI11 clock input/output
	RXD011	Input	RSCI11 receive data input
	TXD011	Output	RSCI11 transmit data output
	RTS011#	Output	RSCI11 request-to-send signal output
	CTS011#	Input	RSCI11 transmission start control input
	DE011	Output	RSCI11 RS-485 driver control output

Table 33.4 RSCI Input/Output Pin (Simple I^2C Mode)

Channel	Pin Name	I/O	Function
RSCI8	SSCL008	I/O	RSCI8 I^2C clock input/output
	SSDA008	I/O	RSCI8 I^2C data input/output
RSCI9	SSCL009	I/O	RSCI9 I^2C clock input/output
	SSDA009	I/O	RSCI9 I^2C data input/output
RSCI11	SSCL011	I/O	RSCI11 I^2C clock input/output
	SSDA011	I/O	RSCI11 I^2C data input/output

Table 33.5 RSCI Input/Output Pin (Simple SPI Mode)

Channel	Pin Name	I/O	Function
RSCI8	SCK008	I/O	RSCI8 clock input/output
	SMISO008	I/O	RSCI8 slave transmit data input/output
	SMOSI008	I/O	RSCI8 master transmit data input/output
	SS008#	Input	RSCI8 slave select input
RSCI9	SCK009	I/O	RSCI9 clock input/output
	SMISO009	I/O	RSCI9 slave transmit data input/output
	SMOSI009	I/O	RSCI9 master transmit data input/output
	SS009#	Input	RSCI9 slave select input
RSCI11	SCK011	I/O	RSCI11 clock input/output
	SMISO011	I/O	RSCI11 slave transmit data input/output
	SMOSI011	I/O	RSCI11 master transmit data input/output
	SS011#	Input	RSCI11 slave select input

Table 33.6 RSCI Input/Output Pin (HBS Support Mode)

Channel	Pin Name	I/O	Function
RSCI8	RXD008	Input	RSCI8 receive data input
	TXD008	Output	RSCI8 transmit data output
	TXDA008/TXDB008	Output	RSCI8 transmit data output (in alternate output)
RSCI9	RXD009	Input	RSCI9 receive data input
	TXD009	Output	RSCI9 transmit data output
	TXDA009/TXDB009	Output	RSCI9 transmit data output (in alternate output)
RSCI11	RXD011	Input	RSCI11 receive data input
	TXD011	Output	RSCI11 transmit data output
	TXDA011/TXDB011	Output	RSCI11 transmit data output (in alternate output)

33.2 Register Descriptions

This chapter describes the RSCI registers, their functional specifications, and their operating specifications.

33.2.1 Receive Shift Register (RSR)

RSR is a shift register which is used to receive serial data input from the RXDn pin and converts it into parallel data. When one frame of data has been received, it is automatically transferred to the RDR register.
The RSR register cannot be directly accessed by the CPU.

33.2.2 Receive Data Register (RDR)

Address(es): RSCI8.RDR 000A 1400h, RSCI9.RDR 000A 1480h, RSCI11.RDR 000E 2080h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	AFER	APER	—	—	ORER	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	FER	PER	DR	MPB	RDAT[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b8 to b0	RDAT[8:0]	Receive Data	RDAT[8:0] bits are a 9-bit field for storing received data. Received data is stored in [6:0] when 7-bit data is selected, in [7:0] when 8-bit data is selected, and in [8:0] when 9-bit data is selected. And 0 is stored in the unused bit.	R
b9	MPB	Multi-Processor Bit Monitor Flag	0: Data transmission cycles 1: ID transmission cycles	R
b10	DR	Receive Data Ready Flag	RFSR.DR flag can be read.	R
b11	PER	Parity Error Flag	(Valid only in asynchronous mode) 0: There is no parity error in the data read from the receive FIFO 1: There is parity error in the data read from the receive FIFO	R
b12	FER	Framing Error Flag	(Valid only in asynchronous mode) 0: There is no framing error in the data read from the receive FIFO 1: There is framing error in the data read from the receive FIFO	R
b23 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R
b24	ORER	Overrun Error Flag	SSR.ORER flag can be read.	R
b26, b25	—	Reserved	These bits are read as 0. The write value should be 0.	R
b27	APER	Aggregate Parity Error Flag	SSR.APER flag can be read.	R
b28	AFER	Aggregate Framing Error Flag	SSR.AFER flag can be read.	R
b31 to b29	—	Reserved	These bits are read as 0. The write value should be 0.	R

In FIFO mode (SCR3.FM bit = 1), this register is 32-stage FIFO buffer configuration.

RDAT[8:0] Bits (Receive Data)

After one frame of data is received, the received data is transferred from the RSR register to this registers, thus allowing the RSR register to receive the next data.
The RSR and RDR registers have a double-buffered construction to enable continuous reception.
In non-FIFO mode, read the RDR register only once when a receive data full interrupt (RXI) request is issued. Without reading received data from RDR register, if the next one frame is received, an overrun error occurs.
In FIFO mode, continuous reception is executed until 32 stages are stored. If data is read when there is no received data in the receive FIFO (RDR register), the value is undefined. When the receive FIFO (RDR register) are full of received data, subsequent serial receive data is lost.
The CPU cannot write to RDR register.
0 is stored in the bit position which isn't received (RDAT[8] or RDAT[7]) at the time of 7bit or 8bit communication of asynchronous and manchester mode.

MPB Flag (Multi-Processor Bit Monitor Flag)

In asynchronous mode and manchester mode, during multi-processor communication (SCR3.MP bit = 1), the value of the multi-processor bit corresponding to the received data (RDAT[8:0] bits) can be read.

PER Flag (Parity Error Flag)

Indicates whether the data read from the receive FIFO has a parity error.
The FER and PER flags store error information of received data only in FIFO mode. In non-FIFO mode, 0 is stored.

FER Flag (Framing Error Flag)

Indicates whether the data read from receive FIFO has a framing error.
The FER and PER flags store error information of received data only in FIFO mode. In non-FIFO mode, 0 is stored.

33.2.3 Transmit Data Register (TDR)

Address(es): RSCI8.TDR 000A 1404h, RSCI9.TDR 000A 1484h, RSCI11.TDR 000E 2084h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	SYNC	—	—	MPBT	TDAT[8:0]								
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b8 to b0	TDAT[8:0]	Transmit Data	TDAT[8:0] bits are a 9-bit field for setting transmit data. Transmit data is set in [6:0] when 7-bit data is selected, in [7:0] when 8-bit data is selected, and in [8:0] when 9-bit data is selected. When writing the TDR register in 8-bit units, write first to the TDR.LH and then TDR.LL.	R/W
b9	MPBT	Transmit Multi-Processor	Value of the multi-processor bit in the transmission frame. This bit is use in asynchronous and manchester mode. When writing to this bit when not used, write the initial value. 0: Data transmission cycles 1: ID transmission cycles	R/W
b11, b10	—	Reserved	These bits are read as 1. The write value should be 1.	R
b12	SYNC	Sync Pulse Select	It is valid when MMCR.SBLEN bit = 1 and MMCR.SYNCE bit = 1 in manchester mode. When writing to this bit when not used, write the initial value. 0: The Start Bit is transmitted as DATA Sync. 1: The Start Bit is transmitted as COMMAND Sync.	R/W
b31 to b13	—	Reserved	These bits are read as 1. The write value should be 1.	R

In FIFO mode (SCR3.FM bit = 1), this register is 32-stage FIFO buffer configuration.

TDAT[8:0] Bits (Transmit Data)

The TDAT[8:0] bits are a 9-bit field for storing transmit data.

When empty space is detected in the TSR register, the transmit data stored in the TDR register is transferred to TSR register, and transmitting is started.

The TSR and TDR registers have a double-buffered construction to realize continuous reception. When the next data to be transmitted is stored in TDR register after one frame of data has been transmitted, the transmitting operation is continued by transfer to the TSR register.

In FIFO mode, continuous serial transmission is executed until there is no transmit data left in the transmit FIFO (TDR register). When transmit FIFO is full of transmit data (32 frames), no more data can be written. If writing of new data is attempted, the data is ignored.

In non-FIFO mode, the TDR register is always readable and writable by the CPU. When a transmit data empty interrupt (TXI) request is issued and SCR0.TE bit is 1, write transmit data to the TDR register only once.

When writing the TDR register in 8-bit units, write first to the TDR.LH and then the TDR.LL.

MPBT Bit (Transmit Multi-Processor)

Selects the multi-processor bit of transmit frame.

SYNC Bit (Sync Pulse Select)

This bit is valid when the MMCR.SYNCE and MMCR.SBLEN bits are set to 1 in Manchester mode (SCR3.MOD [2: 0] bit = 101b).

The Sync type of start bit area in the transmission frame can be set to Data Sync or Command Sync.

33.2.4 Transmit Shift Register (TSR)

TSR register is a shift register that transmits serial data. TSR register cannot be directly accessed by the CPU.

To perform serial data transmission, the SCI first automatically transfers transmit data from TDR register to TSR register, and then sends the data to the TXDn pin.

33.2.5 Control Register 0 (SCR0)

Address(es): RSCI8.SCR0 000A 1408h, RSCI9.SCR0 000A 1488h, RSCI11.SCR0 000E 2088h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	SSE	—	—	TEIE	TIE	—	—	—	RIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	IDSEL	DCME	MPIE	—	—	—	TE	—	—	—	RE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RE	Receive Enable	0: Serial reception is disabled 1: Serial reception is enabled	R/W *1, *3
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R
b4	TE	Transmit Enable	0: Serial transmission is disabled 1: Serial transmission is enabled	R/W *1
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	MPIE	Multi-Processor Interrupt Enable	(Valid in asynchronous mode and manchester mode when SCR3.MP is 1.) This bit should set 0 in smart card interface mode. 0: Non-multi-processor reception 1: Multi-processor reception When the data with the multi-processor bit set to 0 is received, the data is not read, and setting the status flags to 1 is disabled. When the data with the multiprocessor bit set to 1 is received, the MPIE bit is automatically cleared to 0, and non-multi-processor reception is resumed. If you want to continue receiving operation using the multiprocessor function, set this bit to 1 sufficiently earlier than receiving the STOP bit of the next received frame.	R/W *2
b9	DCME	Data Compare Match Enable	(Valid only in asynchronous mode) 0: Data match detection function is disabled 1: Data match detection function is enabled	R/W *2
b10	IDSEL	ID Frame Select	(Valid only in asynchronous mode with multi-processor) 0: It's always compared data in spite of the value of the multi-processor bit. 1: It's compared data when the multi-processor bit is 1 (ID frame) only.	R/W *4
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	RIE	Receive Interrupt Enable	0: RXI and ERI interrupt requests are disabled 1: RXI and ERI interrupt requests are enabled	R/W
b19 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R
b20	TIE	Transmit Interrupt Enable	0: TXI interrupt request is disabled 1: TXI interrupt request is enabled	R/W
b21	TEIE	Transmit End Interrupt Enable	This bit should set 0 in smart card interface mode. 0: TEI interrupt request is disabled 1: TEI interrupt request is enabled	R/W
b23, b22	—	Reserved	These bits are read as 0. The write value should be 0.	R
b24	SSE	SSn# Pin Function Enable	(Valid in simple SPI mode.) In slave mode (SCR3.CKE[1:0] bits = 1xb), set this bit to 1. 0: SSn# pin function is disabled. 1: SSn# pin function is enabled.	R/W *4
b31 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. In clock synchronous mode (SCR3.MOD[2:0] bits = 010b), simple SPI mode (SCR3.MOD[2:0] bits = 011b), and simple I^2C

mode (SCR3.MOD[2:0] bits = 100b), 1 can be written only when TE bit = 0 and RE bit = 0. After setting TE bit or RE bit to 1, only 0 can be written in TE bit and RE bit. In other mode, writing is enabled under any condition.

Note 2. This bit is a bit that is cleared by hardware. Note that writing to a bit other than this bit with a bit manipulation instruction may cause this bit to be unintentionally set to 1 by a read-modify-write operation.

Note 3. In clock synchronous mode and simple SPI mode, receive only setting with internal clock (master mode) is prohibited (TE bit = 0 and RE bit = 1 setting prohibited).

Note 4. Writable only when SCR0.TE bit = 0 and SCR0.RE bit = 0.

RE Bit (Receive Enable)

Enables or disables serial receive operation.

When this bit is set to 1, serial reception starts when RSCI detects the start bit in synchronous mode, the falling edge of the RXD input in Manchester mode, the synchronous clock input in clock synchronous mode, or the start bit in smart card interface mode.

Note that the SCR0 and SCR3 registers should be set prior to setting the RE bit to 1 in order to designate the reception format.

Except smart card interface mode, even if reception is halted by setting the RE bit to 0, the SSR.RDRF, AFER, APER, ORER, MMSR.MCER, SBER, SYER, and PFER flags in non-FIFO mode, and RFSR.DR flag in FIFO mode are not affected and the previous values is retained. In smart card interface mode, even if reception is halted by setting the RE bit to 0, the SSR.AFER, APER, and ORER flags are not affected and the previous value is retained.

TE Bit (Transmit Enable)

Enables or disables serial transmission.

When this bit is set to 1, serial transmission becomes possible. Transmission is started by writing transmit data to TDR register. Note that SCR3 should be set prior to setting the TE bit to 1 in order to designate the transmission format.

MPIE Bit (Multi-Processor Interrupt Enable)

When this bit is set to 1 and the data with the multi-processor bit set to 0 is received, the data is not read and setting the status flags (SSR.RDRF, ORER, AFER, RFSR.DR, MMSR.MCER, SYER, PFER, SBER) are not set.

When the data with the multi-processor bit set to 1 is received, the MPIE is automatically cleared to 0, and normal reception is resumed. For details, refer to section 33.4, Multi-Processor Communication Function. If you want to continue receiving operation using the multiprocessor function, set this bit to 1 sufficiently earlier than receiving the STOP bit of the next received frame.

When the receive data includes the multi-processor bit set to 0, the receive data is not transferred from the RSR register to the RDR register, a receive error is not detected, and setting the flags ORER and AFER, MCER, SYER, PFER, and SBER to 1 is disabled.

When the receive data includes the multi-processor bit set to 1, the MPB flag is set to 1, the MPIE bit is automatically set to 0, the RXI and ERI interrupt requests are enabled (if SCR0.RIE bit is set to 1), and setting the flags ORER, AFER, MCER, SYER, PFER, and SBER to 1 is enabled.

MPIE should be set to 0 if multi-processor communications function is not to be used.

DCME Bit (Data Compare Match Enable)

It can select whether the data match detection function uses or not.

When DCME bit is 1, if RSCI detects the match to the comparison data (SCR4.CMPD[8:0] bits) with receive data, DCME bit is cleared automatically, and after that, RSCI operation mode will be receive mode without data match detection function.

Refer to section 33.3.6, Data Match Detection.

The write value should be 0 other than asynchronous mode.

IDSEL Bit (ID Frame Select)

This bit specifies the condition of the received data to be compared. The bit is valid only when the DCME bit is 1.
When setting this bit to 1, only data in received frames (ID frames) in which the multi-processor bit has the value 1 are compared.
When this bit is set to 0, all received data is compared.

RIE Bit (Receive Interrupt Enable)

Enables or disables RXI and ERI interrupt requests.
RXI and ERI interrupt request is disabled by setting the RIE bit to 0.
An ERI interrupt request can be canceled by reading 1 from the SSR.ORER, AFER, or APER flag and then setting the flag to 0, or setting the RIE bit to 0.
In the case of manchester mode, the MMSR.MCER, SYER, PFER, and SBER flags are also the cause of ERI interrupt request, so the same processing is necessary. For details of these flags, see section 33.2.12, Manchester Mode Control Register (MMCR) and section 33.2.21, Manchester Mode Status Register (MMSR).

TIE Bit (Transmit Interrupt Enable)

Enables or disables TXI interrupt request.
An TXI interrupt request is disabled by setting the TIE bit to 0. At the beginning of transmission, set 1 to SCR0.TE bit and SCR0.TIE bit simultaneously. Then the TXI interrupt request is generated.

TEIE Bit (Transmit End Interrupt Enable)

T Enables or disables a TEI interrupt request.
A TEI interrupt request is disabled by setting the TEIE bit to 0.
In simple I^2C mode, the TEI is allocated to the interrupt on completion of issuing a start, restart, or stop condition (STI). In this case, the TEIE bit can be used to enable or disable the STI.

SSE Bit (SSn# Pin Function Enable)

Set this bit to 1 if the SSn# pin is to be used in control of transmission and reception (in simple SPI mode). Set this bit to 0 in any other mode. Do not set both the SSE and CTSE bits to enabled (even if this setting is made, operation is the same as that when these bits are set to 0).
In the slave mode (SCR3.CKE[1:0] bits = 10b or 11b), SSE should be set 1.
In the master mode (SCR3.CKE[1:0] bits = 00b or 01b) and single-master, the SSn# pin on the master side is not required to control reception and transmission, so SSE should be set 0.

33.2.6 Control Register 1 (SCR1)

Address(es): RSCI8.SCR1 000A 140Ch, RSCI9.SCR1 000A 148Ch, RSCI11.SCR1 000E 208Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	NFEN	—	NFCS[2:0]			—	—	—	HDSEL	—	—	—	LOOP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	RINV	TINV	—	—	PM	PE	—	—	SPB2IO	SPB2DT	—	—	CRSEP	CTSE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CTSE	CTS Enable	0: CTS function is disabled (RTS output function is enabled). 1: CTS function is enabled.	R/W *1
b1	CRSEP	CTS/RTS Separation*2	0: Use either CTS or RTS function 1: Use both CTS and RTS functions at the same time	R/W *1
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b4	SPB2DT	Serial Port Break Data	The output level of TXDn (TXDAn/TXDBn*5) pin is selected when SCR0.TE bit = 0 and SPB2IO bit = 1.*3 When TINV is 0, 0: Low level is output in TXDn (TXDAn/TXDBn*5) pin. 1: High level is output in TXDn (TXDAn/TXDBn*5) pin. When TINV is 1, 0: High level is output in TXDn (TXDAn/TXDBn*5) pin. 1: Low level is output in TXDn (TXDAn/TXDBn*5) pin.	R/W
b5	SPB2IO	Serial Port Break I/O	It's selected whether the value of SPB2DT is output to TXDn (TXDAn/TXDBn*5) pin when SCR0.TE = 0.*3 0: The value of SPB2DT bit isn't output in TXDn (TXDAn/TXDBn*5) pin. 1: The value of SPB2DT bit is output in TXDn (TXDAn/TXDBn*5) pin.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	PE	Parity Enable	(Valid only in asynchronous mode and manchester mode. In smart card interface mode, set 1 to this bit.) When transmitting 0: Parity bit addition is not performed 1: The parity bit is added When receiving 0: Parity bit checking is not performed 1: The parity bit is checked	R/W *1
b9	PM	Parity Mode	(Valid only when the PE bit is 1) 0: Selects even parity 1: Selects odd parity	R/W *1
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R
b12	TINV	Transmitter Output Invert*4	0: Transmit data is not inverted and output to TXDn (TXDAn/TXDBn*5) pin. 1: Transmit data is inverted and output to TXDn (TXDAn/TXDBn*5) pin.	R/W *1
b13	RINV	Receiver Input Invert*4	0: Received data from RXDn is not inverted and input. 1: Received data from RXDn is inverted and input.	R/W *1
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	LOOP	Loopback Mode Setting	It can be used when internal clock operation in asynchronous mode, internal mode operation in manchester mode, internal clock operation in clock synchronous mode. 0: Normal mode 1: Loopback mode	R/W *1

Bit	Symbol	Bit Name	Description	R/W
b19 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R
b20	HDSEL	Half-Duplex Communication Select	In the smart card interface mode, the simple I2C mode, or in the simple SPI mode, this bit should be set 0. 0: TXDn pin, RXDn pin independent 1: TXDn/RXDn pin combination use (Half-duplex communication using TXDn pin)	R/W *1
b23 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R
b26 to b24	NFCS[2:0]	Noise Filter Clock Select	(Valid in asynchronous mode and manchester mode, extended serial mode, and simple I2C mode.) Select for the noise filter's clock source. b26 b24 0 0 0: The base clock signal divided by 1. 0 0 1: The on-chip baud rate generator source clock*6 divided by 1. 0 1 0: The on-chip baud rate generator source clock*6 divided by 2. 0 1 1: The on-chip baud rate generator source clock*6 divided by 4. 1 0 0: The on-chip baud rate generator source clock*6 divided by 8. Settings other than above are prohibited. In simple I2C mode, 000b setting is prohibited. “The on-chip baud rate generator source clock” means the clock selected by SCR2.CKS[1:0] bits.	R/W *1
b27	—	Reserved	This bit is read as 0. The write value should be 0.	R
b28	NFEN	Digital Noise Filter Enable	(Valid in asynchronous mode, manchester mode, extended serial mode and simple I2C) In asynchronous mode, manchester mode and extended serial mode 0: Noise cancellation function for the RXDn input signal is disabled. 1: Noise cancellation function for the RXDn input signal is enabled. In simple I2C mode 0: Noise cancellation function for the SSCLn and SSDAn input signals is disabled. 1: Noise cancellation function for the SSCLn and SSDAn input signals is enabled.	R/W *1
b31 to b29	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Writable only when SCR0.TE bit = 0 and SCR0.RE bit = 0.
Note 2. This bit is available in asynchronous mode and manchester mode. Set this bit to 0 in other mode.
Note 3. Please use this bit in asynchronous mode and manchester mode only. Movement by other mode isn’t guaranteed.
Note 4. RINV/TINV should be set to 0 in smart card interface mode and simple I2C mode.
Note 5. When the alternate output is enabled in HBS support mode.
Note 6. The clock is selected by SCR2.CKS[1:0] bits.

CTSE Bit (CTS Enable)

Set this bit to 1 if the SSn# pin is to be used for inputting of the CTS control signal to control of transmission and reception. The RTS signal is output when this bit is set to 0. Set this bit to 0 in smart card interface mode, simple SPI mode, simple I2C mode, and extended serial mode. Do not set both the CTSE and SSE bits to enabled (even if this setting is made, operation is the same as that when these bits are set to 0).

CRSEP Bit (CTS/RTS Separation)

This bit selects usage of the CTSn#, RTSn#, and CTSn#/RTSn# pins when the CTSE bit is 1.
When either CTS or RTS function is to be used, set this bit to 0.
When both CTS and RTS functions are to be used, set this bit to 1.
When the CTSE bit is set to 0, set this bit to 0.
Refer to Table 33.7 for the relationship between the CRSEP and CTSE bit settings and the pin functions.

Table 33.7 Relationship between the CRSEP and CTSE Bit Settings and Pin Functions

CTSE Bit	CRSEP Bit	CTSn#/RTSn# Multiplexed Pin	CTSn# Dedicated Pin	RTSn# Dedicated Pin
0	0	RTSn# signal output	Disabled	RTSn# signal output
1	0	CTSn# signal input	CTSn# signal input	Disabled
1	1	RTSn# signal output	CTSn# signal input	RTSn# signal output

SPB2DT Bit (Serial Port Break Data), SPB2IO Bit (Serial Port Break I/O)

The TXDn (TXDAn/TXDBn) pins status decided by combination of SCR0.TE bit, SCR1.SPB2IO bit and SCR1.SPB2DT bit is indicated in Table 33.8.

Table 33.8 Controlling the TXDn (TXDAn/TXDBn) Pins

SCR0.TE Bit Setting	SCR1.SPB2IO Bit Setting	SCR1.SPB2DT Bit Setting	TINV Bit Setting	TXDn Pin Status
0 (Transmission disabled)	0 (Input)	0 or 1	0 or 1	Hi-Z
	1 (Output)	0	0	Low is output
			1	High is output
		1	0	High is output
			1	Low is output
1 (Transmission enabled)	0 or 1	0 or 1	0 or 1	Transmit data output pin

PE Bit (Parity Enable)

When PE bit to 1, the parity bit is added to transmit data before transmission, and the parity bit is checked in reception. In the multiprocessor format, the parity bit is not added or checked regardless of this bit setting.

PM Bit (Parity Mode)

Selects the parity mode for transmission and reception (even or odd). In multi-processor mode, this bit is invalid.
For details on the usage of this bit in smart card interface mode, refer to section 33.7.2, Data Format (Except in Block Transfer Mode).

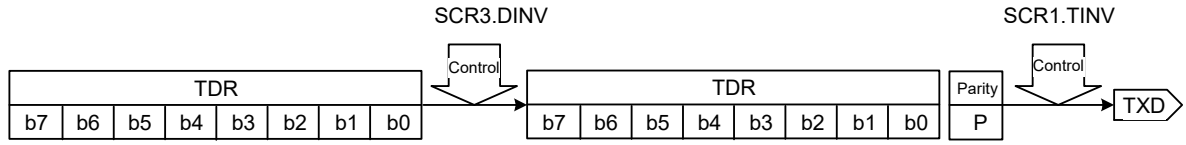
TINV Bit (Transmitter Output Invert), RINV Bit (Receiver Input Invert)

The data of RDR register is controlled by RINV bit and SCR3.DINV bit. And the data from TXDn pin is controlled by TINV bit and SCR3.DINV bit. The control by RINV/TINV bits are done to communication pins (RXDn/TXDn), so they can control not only data bits but also other bits (start bit, stop bit, parity bit). Please refer to Figure 33.2 in detail. When the TXDAn/TXDBn pins are used, the data is also inverted according to the TINV value.
During half-duplex communication and slave operation in simple SPI mode, use the TXDn pin for reception, so set the inversion control of the received data with the TINV bit.

Note: Sentences and a timing chart of the RSCI operation explanation are mentioned by TINV bit = 0 and RINV bit = 0 when TINV's value and RINV's value are not specified.

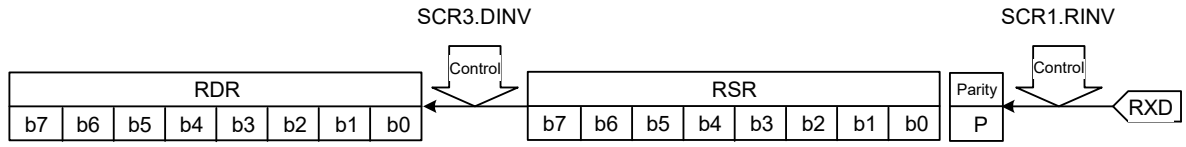
The receive/transmit data control (Data size = 8bits, Even parity, MSB first)

The transmit data is controlled by SCR1.TINV and SCR3.DINV.



SCR3. DINV	SCR1. TINV	TDR	TSR	Prity (even)	TXDn waveform
0	0	BEh	BEh	0	
0	1	BEh	BEh	0	
1	0	BEh	41h	0	
1	1	BEh	41h	0	

The received data is controlled by SCR1.RINV and SCR3.DINV.



SCR3. DINV	SCR1. RINV	RDR	RSR	Prity (even)	RXDn waveform
0	0	BEh	BEh	0	
1	0	41h	BEh	0	
0	1	BEh	BEh	0	
1	1	41h	BEh	0	

Figure 33.2 Example of the Receive/Transmit Data Control

LOOP Bit (Loopback Mode Setting)

When this bit is 1, RSCI blocks the input path from RXD and connects the output path to TXD to the reception data register.
Transmit data can be inverted and received by combining it with TINV bit.
Clock synchronous mode Set to 0 at slave operation and asynchronous mode use of external clock, and extended serial mode.

HDSEL Bit (Half-Duplex Communication Select)

Setting this bit to 1 enables half-duplex communication using the TXDn pin. However, it cannot be used in simple SPI mode, simple I^2C mode and smart card interface mode.
If this bit is set to 1 and SCR0.TE bit = 1, SCR0.RE bit = 0, the TXDn pin becomes communication output. If this bit is set to 1 and SCR0.TE bit = 0, SCR0.RE bit = 1, the TXDn pin becomes the communication input. For details, see section 33.16, Half-Duplex Communication Function.

NFCS[2:0] Bits (Noise Filter Clock Select)

These bits select the sampling clock for the digital noise filter.
To use the noise filter in asynchronous mode, manchester mode and extended serial mode set these bits from 000b to 100b. In simple I^2C mode, set the bits to a value in the range from 001b to 100b.

NFEN Bit (Digital Noise Filter Enable)

This bit enables or disables the digital noise filter function. When the function is enabled, noise cancellation is applied to the RXDn input signal in asynchronous mode, manchester mode, extended serial mode, and noise cancellation is applied to the SSDAn and SSCLn input signals in simple I^2C mode. In any mode other than above, set the NFEN bit to 0 to disable the digital noise filter function. When the function is disabled, input signals are transferred as is, as internal signals.

33.2.7 Control Register 2 (SCR2)

Address(es): RSCI8.SCR2 000A 1410h, RSCI9.SCR2 000A 1490h, RSCI11.SCR2 000E 2090h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MDDR[7:0]								—	—	CKS[1:0]		—	—	—	BRME
Value after reset:	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BRR[7:0]								—	ABCSE	ABCS	BGDM	—	BCP[2:0]		
Value after reset:	1	1	1	1	1	1	1	1	0	0	0	0	0	1	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	BCP[2:0]	Base Clock Pulse	Selects the number of base clock cycles in smart card interface mode. b2 b0 0 0: 93 clock cycles (S = 93)*2 0 0 1: 128 clock cycles (S = 128)*2 0 1 0: 186 clock cycles (S = 186)*2 0 1 1: 512 clock cycles (S = 512)*2 1 0 0: 32 clock cycles (S = 32)*2 (Initial value) 1 0 1: 64 clock cycles (S = 64)*2 1 1 0: 372 clock cycles (S = 372)*2 1 1 1: 256 clock cycles (S = 256)*2	R/W *1
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R
b4	BGDM	Baud Rate Generator Double-Speed Mode Select	Valid in asynchronous/manchester/clock synchronous/simple SPI mode and SCR3.CKE[1] bit = 0. 0: Baud rate generator outputs the clock with single frequency. 1: Baud rate generator outputs the clock with doubled frequency.	R/W *1
b5	ABCS	Asynchronous Mode Base Clock Select	(Valid only in asynchronous mode, manchester mode and extended serial mode) 0: Selects 16 base clock cycles for 1-bit period. 1: Selects 8 base clock cycles for 1-bit period.	R/W *1
b6	ABCSE	Asynchronous Mode Base Clock Select Extended	(Valid only in asynchronous mode and SCR3.CKE[1] bit = 0) 0: Clock cycles for 1-bit period is decided with combination between SCR2.BGDM bit and SCR2.ABCS bit. 1: Baud rate is 6 base clock cycles for 1-bit period and the clock of a double frequency is output from the baud rate generator.	R/W *1
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R
b15 to b8	BRR[7:0]	Bit Rate Setting	An 8-bit field that adjusts the bit rate.	R/W *1
b16	BRME	Bit Rate Modulation Enable	0: Bit rate modulation function is disabled. 1: Bit rate modulation function is enabled.	R/W *1
b19 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R
b21, b20	CKS[1:0]	Clock Select	b21 b20 0 0: PCLK (n = 0)*3 0 1: PCLK/4 (n = 1)*3 1 0: PCLK/16 (n = 2)*3 1 1: PCLK/64 (n = 3)*3	R/W *1
b23, b22	—	Reserved	These bits are read as 0. The write value should be 0.	R
b31 to b24	MDDR[7:0]	Modulation Duty Setting	MDDR[7:0] bits corrects the bit rate adjusted by the BRR[7:0] bits.	R/W *1

Note 1. Writable only when SCR0.TE bit = 0 and SCR0.RE bit = 0.
Note 2. S is the value of S in BRR[7:0] bits explanation.
Note 3. n is the decimal notation of the value of n in BRR[7:0] bits explanation.

BCP[2:0] Bits (Base Clock Pulse)

These bits select the number of base clock cycles in a 1-bit data transfer time in smart card interface mode.
For details, refer to section 33.7.4, Receive Data Sampling Timing and Reception Margin.

BGDM Bit (Baud Rate Generator Double-Speed Mode Select)

This bit is valid when the on-chip baud rate generator is selected as the clock source (SCR3.CKE[1] bit = 0) in asynchronous mode, manchester mode, clock synchronous mode, simple SPI mode. When external clock is selected (SCR3.CKE[1] bit = 1), set it to 0. For the clock output from the baud rate generator, either single or doubled frequency can be selected. The base clock is generated by the clock output from the baud rate generator. When the BGDM bit is set to 1, the base clock cycle is halved and the bit rate is doubled.
Set this bit to 0 in modes other than asynchronous mode or manchester mode or clock synchronous mode or simple SPI.

ABCS Bit (Asynchronous Mode Base Clock Select)

Selects the clock cycles for 1-bit period.
Set it to 0 in modes other than asynchronous mode, manchester mode and extended serial mode.

ABCSE Bit (Asynchronous Mode Base Clock Select Extended)

The pulse number for a base clock at 1-bit period is 6 and the clock of a double frequency is output from baud rate generator. Only when the bit rate is set to 6 dividing frequency of the bus clock, please use this bit and set SCR2.CKS[1:0] bits = 00b and BRR[7:0] bits = 0.
Set it to 0 in modes other than asynchronous mode. Even in asynchronous mode, set it to 0 when using external clock.

Table 33.9 Base Clock Cycle Number per 1-Bit

ABCSE Bit	ABCS Bit	BGDM Bit	The Base Clock Cycles/ 1-Bit	The Output Frequency of the Baud Rate Generator
0	0	0	16	×1
0	0	1	16	×2
0	1	0	8	×1
1	1	1	8	×2
1	—	—	6	×2

—: Don’t care

BRR[7:0] Bits (Bit Rate Setting)

BRR[7:0] bits are an 8-bit field that adjusts the bit rate.
RSCI has independent baud rate generator control, different bit rates can be set for each. Table 33.10 shows the relationship between the setting (N) in the BRR[7:0] bits and the bit rate (B) for asynchronous mode, multiprocessor transfer, manchester mode, clock synchronous mode, smart card interface mode, simple SPI mode, and simple I^2C mode.

Table 33.10 Relationship between N Setting in BRR[7:0] Bits and Bit Rate B

Mode	SCR2 Settings			BRR[7:0] Bits Setting	Error (%)
	BGDM bit	ABCS bit	ABCSE bit		
Asynchronous, multi-processor communication, manchester, extended serial*3	0	0	0	$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 64 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
	1	0	0	$N = \frac{PCLK \times 10^6}{32 \times 2^{2n-1} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 32 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
	0	1	0		
	1	1	0	$N = \frac{PCLK \times 10^6}{16 \times 2^{2n-1} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 16 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
	0 or 1	0 or 1	1*2	$N = \frac{PCLK \times 10^6}{12 \times 2^{2n-1} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 12 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
Clock synchronous, simple SPI	0	0 (Initial value)	0 (Initial value)	$N = \frac{PCLK \times 10^6}{8 \times 2^{2n-1} \times B} - 1$	
	1	0 (Initial value)	0 (Initial value)	$N = \frac{PCLK \times 10^6}{4 \times 2^{2n-1} \times B} - 1$	
Smart card interface				$N = \frac{PCLK \times 10^6}{S \times 2^{2n+1} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times S \times 2^{2n+1} \times (N+1)} - 1 \right\} \times 100$
Simple I^2C*1				$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	

B: Bit rate (bps)
N: BRR[7:0] bits setting ($0 \le N \le 255$)
PCLK: Operating frequency (MHz)
n and S: Determined by the settings of the SCR2 registers as listed in the table below.Please be careful about "$2^{(2n+1)}$" is used in the expression for smart card interface, "$2^{(2n-1)}$" is used in other mode.

Note 1. Adjust the bit rate so that the widths at high and low level of the SCL output in simple I^2C mode satisfy the I^2C standard.
Note 2. In manchester mode, only ABCSE bit = 0 can be selected.
Note 3. In extended serial mode, BGDM bit = 0 and ABCSE bit = 0 can be selected.

Table 33.11 Calculating Widths at High and Low Level for SCL

Mode	SCL	Formula (Result in Seconds)
I^2C	High period (minimum value)	$(N+1) \times 4 \times 2^{2n-1} \times 7 \times \frac{1}{PCLK \times 10^6}$
	Low period (minimum value)	$(N+1) \times 4 \times 2^{2n-1} \times 8 \times \frac{1}{PCLK \times 10^6}$

Table 33.12 Clock Source Settings

SCR2 Setting		
CKS[1:0] Bits	Clock Source	n
0 0	PCLK	0
0 1	PCLK/4	1
1 0	PCLK/16	2
1 1	PCLK/64	3

Table 33.13 Base Clock Settings in Smart Card Interface Mode

SCR2 Setting		
BCP[2:0] Bits	**Base Clock Cycles for 1-bit Period**	**S**
0 0 0	93 clock cycles	93
0 0 1	128 clock cycles	128
0 1 0	186 clock cycles	186
0 1 1	512 clock cycles	512
1 0 0	32 clock cycles	32
1 0 1	64 clock cycles	64
1 1 0	372 clock cycles	372
1 1 1	256 clock cycles	256

Table 33.14 and Table 33.15 list examples of N settings in BRR[7:0] in asynchronous mode and manchester mode. Table 33.16 lists the maximum bit rate settable for each operating frequency. Examples of BRR[7:0] bits (N) settings in clock synchronous mode and simple SPI mode are listed in Table 33.18. Examples of BRR[7:0] bits (N) settings in smart card interface mode are listed in Table 33.20. Examples of BRR[7:0] bits (N) settings in simple I^2C mode are listed in Table 33.22. In smart card interface mode, the number of base clock cycles S in a 1-bit data transfer time can be selected. For details, refer to section 33.7.4, Receive Data Sampling Timing and Reception Margin. Table 33.17 and Table 33.19 list the maximum bit rates with external clock input.

When either the asynchronous mode base clock select bit (ABCS) or the baud rate generator double-speed mode select bit (BGDM) is set to 1 in asynchronous mode and manchester mode, the bit rate becomes twice that listed in Table 33.14 and Table 33.15. When both of those registers are set to 1, the bit rate becomes four times the listed value.

Table 33.14 Examples of BRR[7:0] Bits Settings for Various Bit Rates (Asynchronous Mode and Manchester Mode) (1)

Bit Rate (bps)	Operating Frequency PCLK (MHz)														
	8			9.8304			10			12			12.288		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	141	0.03	2	174	–0.26	2	177	–0.25	2	212	0.03	2	217	0.08
150	2	103	0.16	2	127	0.00	2	129	0.16	2	155	0.16	2	159	0.00
300	1	207	0.16	1	255	0.00	2	64	0.16	2	77	0.16	2	79	0.00
600	1	103	0.16	1	127	0.00	1	129	0.16	1	155	0.16	1	159	0.00
1200	0	207	0.16	0	255	0.00	1	64	0.16	1	77	0.16	1	79	0.00
2400	0	103	0.16	0	127	0.00	0	129	0.16	0	155	0.16	0	159	0.00
4800	0	51	0.16	0	63	0.00	0	64	0.16	0	77	0.16	0	79	0.00
9600	0	25	0.16	0	31	0.00	0	32	–1.36	0	38	0.16	0	39	0.00
19200	0	12	0.16	0	15	0.00	0	15	1.73	0	19	–2.34	0	19	0.00
31250	0	7	0.00	0	9	–1.70	0	9	0.00	0	11	0.00	0	11	2.40
38400	—	—	—	0	7	0.00	0	7	1.73	0	9	–2.34	0	9	0.00

Bit Rate (bps)	Operating Frequency PCLK (MHz)														
	14			16			17.2032			18			19.6608		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	248	–0.17	3	70	0.03	3	75	0.48	3	79	–0.12	3	86	0.31
150	2	181	0.16	2	207	0.16	2	223	0.00	2	233	0.16	2	255	0.00
300	2	90	0.16	2	103	0.16	2	111	0.00	2	116	0.16	2	127	0.00
600	1	181	0.16	1	207	0.16	1	223	0.00	1	233	0.16	1	255	0.00
1200	1	90	0.16	1	103	0.16	1	111	0.00	1	116	0.16	1	127	0.00
2400	0	181	0.16	0	207	0.16	0	223	0.00	0	233	0.16	0	255	0.00
4800	0	90	0.16	0	103	0.16	0	111	0.00	0	116	0.16	0	127	0.00
9600	0	45	–0.93	0	51	0.16	0	55	0.00	0	58	–0.69	0	63	0.00
19200	0	22	–0.93	0	25	0.16	0	27	0.00	0	28	1.02	0	31	0.00
31250	0	13	0.00	0	15	0.00	0	16	1.20	0	17	0.00	0	19	–1.70
38400	—	—	—	0	12	0.16	0	13	0.00	0	14	–2.34	0	15	0.00

Note: This is an example when the SCR2.ABCS bit = 0, SCR2.BGDM bit = 0 and SCR2.ABCSE bit = 0.
When either ABCS bit or BGDM bit is set to 1, the bit rate doubles.
When both ABCS bit = 1 and BGDM bit = 1, the bit rate increases four times.

Table 33.15 Examples of BRR[7:0] Bits Settings for Various Bit Rates (Asynchronous Mode and Manchester Mode) (2)

Bit Rate (bps)	Operating Frequency PCLK (MHz)														
	20			25			30			33			40		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	88	–0.25	3	110	–0.02	3	132	0.13	3	145	0.33	3	177	–0.25
150	3	64	0.16	3	80	0.47	3	97	–0.35	3	106	0.39	3	129	0.16
300	2	129	0.16	2	162	–0.15	2	194	0.16	2	214	–0.07	3	64	0.16
600	2	64	0.16	2	80	0.47	2	97	–0.35	2	106	0.39	2	129	0.16
1200	1	129	0.16	1	162	–0.15	1	194	0.16	1	214	–0.07	2	64	0.16
2400	1	64	0.16	1	80	0.47	1	97	–0.35	1	106	0.39	1	129	0.16
4800	0	129	0.16	0	162	–0.15	0	194	0.16	0	214	–0.07	1	64	0.16
9600	0	64	0.16	0	80	0.47	0	97	–0.35	0	106	0.39	0	129	0.16
19200	0	32	–1.36	0	40	–0.76	0	48	–0.35	0	53	–0.54	0	64	0.16
31250	0	19	0.00	0	24	0.00	0	29	0.00	0	32	0.00	0	39	0.00
38400	0	15	1.73	0	19	1.73	0	23	1.73	0	26	–0.54	0	32	–1.36

Bit Rate (bps)	Operating Frequency PCLK (MHz)											
	50			60			100			120		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	221	–0.02	—	—	—	—	—	—	—	—	—
150	3	162	–0.15	3	194	0.16	—	—	—	—	—	—
300	3	80	0.47	3	97	–0.35	3	162	–0.15	3	194	0.16
600	2	162	–0.15	2	194	0.16	3	80	0.47	3	97	–0.35
1200	2	80	0.47	2	97	–0.35	2	162	–0.15	2	194	0.16
2400	1	162	–0.15	1	194	0.16	2	80	0.47	2	97	–0.35
4800	1	80	0.47	1	97	–0.35	1	162	–0.15	1	194	0.16
9600	0	162	–0.15	0	194	0.16	1	80	0.47	1	97	–0.35
19200	0	80	0.47	0	97	–0.35	0	162	–0.15	0	194	0.16
31250	0	49	0.00	0	59	0.00	1	24	0.00	0	119	0.00
38400	0	40	–0.76	0	48	–0.35	0	80	0.47	0	97	–0.35

Note: This is an example when the SCR2.ABCS bit = 0, SCR2.BGDM bit = 0 and SCR2.ABCSE bit = 0.
When either ABCS bit or BGDM bit is set to 1, the bit rate doubles.
When both ABCS bit = 1 and BGDM bit = 1, the bit rate increases four times.

Table 33.16 Maximum Bit Rate for Each Operating Frequency (Asynchronous Mode and Manchester Mode) (1)

PCLK (MHz)	SCR2 Settings					Maximum Bit Rate (bps)	PCLK (MHz)	SCR2 Settings					Maximum Bit Rate (bps)
	BGDM Bit	ABCS Bit	ABCSE Bit	n	N			BGDM Bit	ABCS Bit	ABCSE Bit	n	N	
8	0	0	0	0	0	250000	19.6608	0	0	0	0	0	614400
		1	0	0	0	500000			1	0	0	0	1228800
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1000000			1	0	0	0	2457600
	0 or 1	0 or 1	1	0	0	1333333		0 or 1	0 or 1	1	0	0	3276800
9.8304	0	0	0	0	0	307200	20	0	0	0	0	0	625000
		1	0	0	0	614400			1	0	0	0	1250000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1228800			1	0	0	0	2500000
	0 or 1	0 or 1	1	0	0	1638400		0 or 1	0 or 1	1	0	0	3333333
10	0	0	0	0	0	312500	25	0	0	0	0	0	781250
		1	0	0	0	625000			1	0	0	0	1562500
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1250000			1	0	0	0	3125000
	0 or 1	0 or 1	1	0	0	1666667		0 or 1	0 or 1	1	0	0	4166667
12	0	0	0	0	0	375000	30	0	0	0	0	0	937500
		1	0	0	0	750000			1	0	0	0	1875000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1500000			1	0	0	0	3750000
	0 or 1	0 or 1	1	0	0	2000000		0 or 1	0 or 1	1	0	0	5000000
12.288	0	0	0	0	0	384000	33	0	0	0	0	0	1031250
		1	0	0	0	768000			1	0	0	0	2062500
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1536000			1	0	0	0	4125000
	0 or 1	0 or 1	1	0	0	2048000		0 or 1	0 or 1	1	0	0	5500000
14	0	0	0	0	0	437500	40	0	0	0	0	0	1250000
		1	0	0	0	875000			1	0	0	0	2500000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1750000			1	0	0	0	5000000
	0 or 1	0 or 1	1	0	0	2333333		0 or 1	0 or 1	1	0	0	6666667
16	0	0	0	0	0	500000	50	0	0	0	0	0	1562500
		1	0	0	0	1000000			1	0	0	0	3125000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	2000000			1	0	0	0	6250000
	0 or 1	0 or 1	1	0	0	2666667		0 or 1	0 or 1	1	0	0	8333333
17.2032	0	0	0	0	0	537600	60	0	0	0	0	0	1875000
		1	0	0	0	1075200			1	0	0	0	3750000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	2150400			1	0	0	0	7500000
	0 or 1	0 or 1	1	0	0	2867200		0 or 1	0 or 1	1	0	0	10000000
18	0	0	0	0	0	562500	120	0	0	0	0	0	3750000
		1	0	0	0	1125000			1	0	0	0	7500000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	2250000			1	0	0	0	15000000
	0 or 1	0 or 1	1	0	0	3000000		0 or 1	0 or 1	1	0	0	20000000

Table 33.17 Maximum Bit Rate with External Clock Input (Asynchronous Mode)

		Maximum Bit Rate (bps)	
PCLK (MHz)	External Input Clock (MHz)	SCR2.ABCS Bit = 0	SCR2.ABCS Bit = 1
8	2.0000	125000	250000
9.8304	2.4576	153600	307200
10	2.5000	156250	312500
12	3.0000	187500	375000
12.288	3.0720	192000	384000
14	3.5000	218750	437500
16	4.0000	250000	500000
17.2032	4.3008	268800	537600
18	4.5000	281250	562500
19.6608	4.9152	307200	614400
20	5.0000	312500	625000
25	6.2500	390625	781250
30	7.5000	468750	937500
33	8.2500	515625	1031250
40	10.0000	625000	1250000
50	12.5000	781250	1562500
60	15.0000	937500	1875000
120	30.0000	1875000	3750000

Table 33.18 Examples of BRR[7:0] Bits Settings for Various Bit Rates (Clock Synchronous Mode, Simple SPI Mode)

Bit Rate (bps)	Operating Frequency PCLK (MHz)														
	8			10			30			60			120		
	BGDM	n	N	BGDM	n	N	BGDM	n	N	BGDM	n	N	BGDM	n	N
250	0	3	124	0	3	177	—	—	—	—	—	—	—	—	—
500	0	2	249	0	3	77	0	3	233	—	—	—	—	—	—
1 k	0	2	124	0	3	38	0	3	116	0	3	233	—	—	—
2.5 k	0	2	49	0	1	249	0	3	46	0	3	93	0	3	187
5 k	0	2	24	0	1	124	0	2	93	0	3	46	0	3	93
10 k	0	1	49	0	0	249	0	2	46	0	2	93	0	3	46
25 k	0	2	4	0	1	24	0	1	74	0	1	149	0	2	74
50 k	0	1	9	0	0	49	0	0	149	0	1	74	0	1	149
100 k	0	1	4	0	0	24	0	0	74	0	0	149	0	1	74
250 k	0	1	1	0	0	9	0	0	29	0	1	14	0	1	29
500 k	0	1	0	0	0	4	0	0	14	0	0	29	0	1	14
1 M	0	0	1	1	0	4	1	0	14	0	0	14	0	0	29
2.5 M	—	—	—	0	0	0	0	0	2	0	0	5	0	1	2
5 M	—	—	—	1	0	0	1	0	2	0	0	2	0	0	5
7.5 M	—	—	—	—	—	—	0	0	0	0	0	1	0	1	0
60M	—	—	—	—	—	—	—	—	—	—	—	—	1	0	0

—: Can be set, but an error over 10% will occur.

Table 33.19 Maximum Bit Rate with External Clock Input (Clock Synchronous Mode, Simple SPI Mode)

PCLK (MHz)	External Input Clock (MHz)	Maximum Bit Rate (Mbps)
8	4	4
10	5	5
12	6	6
14	7	7
16	8	8
18	9	9
20	10	10
25	12.5	12.5
30	15	15
33	16.5	16.5
40	20	20
50	25	25
60	30	30
120	60	60

Table 33.20 BRR[7:0] Bits Settings for Various Bit Rates (Smart Card Interface Mode, n = 0, S = 372)

Bit Rate (bps)	PCLK (MHz)	n	N	Error (%)
9600	7.1424	0	0	0.00
	10.00	0	1	–30.00
	10.7136	0	1	–25.00
	13.00	0	1	–8.99
	14.2848	0	1	0.00
	16.00	0	1	12.01
	18.00	0	2	–15.99
	20.00	0	2	–6.66
	25.00	0	3	–12.49
	30.00	0	3	5.01
	33.00	0	4	–7.59
	40.00	0	5	–6.66
	50.00	0	6	0.01
	60.00	0	7	5.01
	120.00	0	16	–1.17

Table 33.21 Maximum Bit Rate for Each Operating Frequency (Smart Card Interface Mode, S = 32)

PCLK (MHz)	Maximum Bit Rate (bps)	n	N
10.00	156250	0	0
10.7136	167400	0	0
13.00	203125	0	0
16.00	250000	0	0
18.00	281250	0	0
20.00	312500	0	0
25.00	390625	0	0
30.00	468750	0	0
33.00	515625	0	0
40.00	625000	0	0
50.00	781250	0	0
60.00	937500	0	0
120.00	1875000	0	0

Table 33.22 BRR[7:0] Bits Settings for Various Bit Rates (Simple I²C Mode)

Bit Rate (bps)	Operating Frequency PCLK (MHz)														
	8			10			16			20			25		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
10 k	0	24	0.0	0	31	–2.3	1	12	–3.8	1	15	–2.3	1	19	–2.3
25 k	0	9	0.0	0	12	–3.8	1	4	0.0	1	6	–10.7	1	7	–2.3
50 k	0	4	0.0	0	6	–10.7	1	2	–16.7	1	3	–21.9	1	3	–2.3
100 k	0	2	–16.7	0	3	–21.9	0	4	0.0	0	6	–10.7	1	1	–2.3
250 k	0	0	0.0	0	1	–37.5	0	1	0.0	0	2	–16.7	0	3	–21.9
350 k										0	1	–10.7	0	2	–25.6
400 k										0	1	–21.9	0	1	–2.3

Bit Rate (bps)	Operating Frequency PCLK (MHz)														
	30			33			40			50			60		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
10 k	1	23	–2.3	1	25	–0.8	0	124	0.00	2	9	–2.3	1	46	–0.27
25 k	1	9	–6.3	1	10	–6.3	0	40	0.00	2	3	–2.3	0	74	0.00
50 k	1	4	–6.3	1	5	–14.1	0	24	0.00	2	1	–2.3	0	37	–1.32
100 k	1	2	–21.9	1	2	–14.1	0	12	–3.85	1	3	–2.3	0	18	–1.32
250 k	0	3	–6.3	0	4	–17.5	0	4	0.00	0	6	–10.7	0	7	–6.25
350 k	0	2	–10.7	0	2	–1.8	0	3	–10.71	0	4	–10.7	0	4	7.14
400 k	0	1	17.2	0	2	–14.1	0	2	4.17	0	3	–2.34	0	4	–6.25

Bit Rate (bps)	Operating Frequency PCLK (MHz)		
	120		
	n	N	Error (%)
10 k	1	93	–0.27
25 k	0	149	0.00
50 k	0	74	0.00
100 k	0	37	–1.31
250 k	0	14	0.00
350 k	0	10	–2.60
400 k	0	8	4.17

Table 33.23 Minimum Widths at High and Low Level for SCL at Various Bit Rates (Simple I²C Mode)

Bit Rate (bps)	Operating Frequency PCLK (MHz)											
	8			10			16			20		
	n	N	Min. Widths at High/Low Level for SCL (µs)	n	N	Min. Widths at High/Low Level for SCL (µs)	n	N	Min. Widths at High/Low Level for SCL (µs)	n	N	Min. Widths at High/Low Level for SCL (µs)
10 k	0	24	43.75/50.00	0	31	44.80/51.20	1	12	45.50/52.00	1	15	44.80/51.20
25 k	0	9	17.50/20.00	0	12	18.20/20.80	1	4	17.50/20.00	1	6	19.60/22.40
50 k	0	4	8.75/10.00	0	6	9.80/11.20	1	2	10.50/12.00	1	3	11.20/12.80
100 k	0	2	5.25/6.00	0	3	5.60/6.40	0	4	4.37/5.00	0	6	4.90/5.60
250 k	0	0	1.75/2.00	0	1	2.80/3.20	0	1	1.75/2.00	0	2	2.10/2.40
350 k										0	1	1.40/1.60
400 k										0	1	1.40/1.60

Bit Rate (bps)	Operating Frequency PCLK (MHz)											
	25			30			33			40		
	n	N	Min. Widths at High/Low Level for SCL (µs)	n	N	Min. Widths at High/Low Level for SCL (µs)	n	N	Min. Widths at High/Low Level for SCL (µs)	n	N	Min. Widths at High/Low Level for SCL (µs)
10 k	1	19	44.80/51.20	1	23	44.80/51.20	1	25	44.12/50.42	1	32	46.20/52.80
25 k	1	7	17.92/20.48	1	9	18.66/21.33	1	10	18.66/21.33	1	12	18.20/20.80
50 k	1	3	8.96/10.24	1	4	9.33/10.66	1	5	10.18/11.63	1	6	9.80/11.20
100 k	1	1	4.48/5.12	1	2	5.60/6.40	1	2	5.09/5.81	0	13	4.90/5.60
250 k	0	3	2.24/2.56	0	3	1.86/2.13	0	4	2.12/2.42	0	4	1.75/2.00
350 k	0	2	1.68/1.92	0	2	1.40/1.60	0	2	1.27/1.45	0	3	1.40/1.60
400 k	0	1	1.12/1.28	0	1	0.93/1.07	0	2	1.27/1.45	0	2	1.05/1.20

Bit Rate (bps)	Operating Frequency PCLK (MHz)								
	50			60			120		
	n	N	Min. Widths at High/Low Level for SCL (µs)	n	N	Min. Widths at High/Low Level for SCL (µs)	n	N	Min. Widths at High/Low Level for SCL (µs)
10 k	2	9	44.80/51.20	1	47	44.80/51.20	1	93	43.87/50.13
25 k	2	3	17.92/20.48	0	74	17.50/20.00	0	149	17.50/20.00
50 k	2	1	8.96/10.24	0	37	8.87/10.13	0	74	8.75/10.00
100 k	1	3	4.48/5.12	0	18	4.43/5.07	0	37	4.43/5.07
250 k	0	6	1.96/2.24	0	7	1.87/2.13	0	15	1.87/2.13
350 k	0	4	1.40/1.60	0	5	1.40/1.60	0	10	1.28/1.47
400 k	0	3	1.12/1.28	0	4	1.17/1.33	0	9	1.17/1.33

BRME Bit (Bit Rate Modulation Enable)

Enables and disables the bit rate modulation function. The bit rate generated by on-chip baud rate generator is evenly corrected when this function is enabled.

This bit can only be set to 1 in asynchronous mode and simple I²C mode. Set this bit to 0 in clock synchronous mode, simple SPI mode, smart card interface mode, manchester mode, and extended serial mode.

CKS[1:0] Bits (Clock Select)

These bits select the clock source for the on-chip baud rate generator.

For the relationship between the settings of these bits and the baud rate, refer to BRR[7:0] bits explanation.

MDDR[7:0] Bits (Modulation Duty Setting)

When the BRME bit is set to 1, the bit rate generated by the on-chip baud rate generator is evenly corrected according to the settings of MDDR (M/256). The relationship between the MDDR[7:0] bits setting (M) and the bit rate (B) is given in Table 33.24.

The initial value of MDDR[7:0] bits is FFh. Bit 7 in this register is fixed to 1.

Table 33.24 Relationship between MDDR[7:0] Bits Setting (M) and Bit Rate (B) When Bit Rate Modulation Function is Used

Mode*1	SCR2 Settings			BRR[7:0] Bits Setting	Error (%)
	BGDM Bit	ABCS Bit	ABCSE Bit		
Asynchronous, multi-processor communication	0	0	0	$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 64 \times 2^{2n-1} \times \frac{256}{M} \times (N+1)} - 1 \right\} \times 100$
	1	0	0	$N = \frac{PCLK \times 10^6}{32 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 32 \times 2^{2n-1} \times \frac{256}{M} \times (N+1)} - 1 \right\} \times 100$
	0	1	0		
	1	1	0	$N = \frac{PCLK \times 10^6}{16 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 16 \times 2^{2n-1} \times \frac{256}{M} \times (N+1)} - 1 \right\} \times 100$
	0 or 1	0 or 1	1	$N = \frac{PCLK \times 10^6}{12 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 12 \times 2^{2n-1} \times \frac{256}{M} \times (N+1)} - 1 \right\} \times 100$
Simple I^2C*2				$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	

B: Bit rate (bps)
M: MDDR[7:0] bits setting (128 ≤ M ≤ 255)
N: BRR[7:0] bits setting (0 ≤ N ≤ 255)
PCLK: Operating frequency (MHz)
n: Determined by the settings of the SCR2.CKS[1:0] bits as listed in Table 33.12, Clock Source Settings.

Note 1. Do not use this function in clock synchronous mode, simple SPI mode, smart card Interface mode, manchester mode and extended serial mode.

Note 2. Adjust the bit rate so that the widths at high and low level of the SCL output in simple I^2C mode satisfy the I^2C standard.

Table 33.25 and Table 33.26 list examples of N settings in BRR[7:0] bits and M settings in MDDR[7:0] bits in asynchronous mode.

Table 33.25 Examples of BRR[7:0] Bits and MDDR[7:0] Bits Settings for Various Bit Rates (Asynchronous Mode) (1)

Bit Rate (bps)	Operating Frequency PCLK (MHz)														
	8					9.8304					10				
	n	N	M	BGDM Bit	Error (%)	n	N	M	BGDM Bit	Error (%)	n	N	M	BGDM Bit	Error (%)
38400	0	5	236	0	0.03	0	7	(256) *1	0	0.00	0	10	173	1	–0.01
57600	0	3	236	0	0.03	0	4	240	0	0.00	0	4	236	0	0.03
115200	0	1	236	0	0.03	0	1	192	0	0.00	0	4	236	1	0.03
230400	0	0	236	0	0.03	0	0	192	0	0.00	0	1	189	1	0.14
460800	0	0	236	1	0.03	0	0	192	1	0.00	0	0	189	1	0.14

Bit Rate (bps)	Operating Frequency PCLK (MHz)														
	12					12.288					14				
	n	N	M	BGDM Bit	Error (%)	n	N	M	BGDM Bit	Error (%)	n	N	M	BGDM Bit	Error (%)
38400	0	8	236	0	0.03	0	9	(256) *1	0	0.00	0	16	191	1	0.00
57600	0	5	236	0	0.03	0	4	192	0	0.00	0	13	236	1	0.03
115200	0	2	236	0	0.03	0	4	192	1	0.00	0	6	236	1	0.03
230400	0	2	236	1	0.03	0	2	230	1	–0.17	0	2	202	1	–0.11
460800	0	0	157	1	–0.18	0	0	154	1	0.26	0	0	135	1	0.14

Bit Rate (bps)	Operating Frequency PCLK (MHz)														
	16					17.2032					18				
	n	N	M	BGDM Bit	Error (%)	n	N	M	BGDM Bit	Error (%)	n	N	M	BGDM Bit	Error (%)
38400	0	11	236	0	0.03	0	13	(256) *1	0	0.00	0	18	166	1	–0.01
57600	0	7	236	0	0.03	0	6	192	0	0.00	0	18	249	1	–0.01
115200	0	3	236	0	0.03	0	6	192	1	0.00	0	8	236	1	0.03
230400	0	1	236	0	0.03	0	3	219	1	–0.20	0	1	210	0	0.14
460800	0	1	236	1	0.03	0	1	219	1	–0.20	0	0	210	0	0.14

Note: This is an example when the SCR2.ABCS bit = 0 and SCR2.ABCSE bit = 0.
Note 1. It means that the bit rate modulation function is disable. (SCR2.BRME bit = 0, M = 256)

Table 33.26 Examples of BRR[7:0] Bits and MDDR[7:0] Bits Settings for Various Bit Rates (Asynchronous Mode) (2)

Bit Rate (bps)	Operating Frequency PCLK (MHz)														
	19.6608					20					25				
	n	N	M	BGDM Bit	Error (%)	n	N	M	BGDM Bit	Error (%)	n	N	M	BGDM Bit	Error (%)
38400	0	15	(256) *1	0	0.00	0	10	173	0	–0.01	0	11	151	0	0.00
57600	0	9	240	0	0.00	0	9	236	0	0.03	0	7	151	0	0.00
115200	0	4	240	0	0.00	0	4	236	0	0.03	0	3	151	0	0.00
230400	0	1	192	0	0.00	0	4	236	1	0.03	0	1	151	0	0.00
460800	0	0	192	0	0.00	0	0	189	0	0.14	0	0	151	0	0.00

Bit Rate (bps)	Operating Frequency PCLK (MHz)														
	30					33					40				
	n	N	M	BGDM Bit	Error (%)	n	N	M	BGDM Bit	Error (%)	n	N	M	BGDM Bit	Error (%)
38400	0	36	194	1	0.01	0	14	143	0	0.01	0	21	173	0	–0.01
57600	0	10	173	0	–0.01	0	9	143	0	0.01	0	38	230	1	–0.01
115200	0	10	173	1	–0.01	0	4	143	0	0.01	0	9	236	0	0.03
230400	0	6	220	1	–0.09	0	4	143	1	0.01	0	4	236	0	0.03
460800	0	3	252	1	0.14	0	1	229	0	0.10	0	4	236	1	0.03

Bit Rate (bps)	Operating Frequency PCLK (MHz)														
	50					60					120				
	n	N	M	BGDM Bit	Error (%)	n	N	M	BGDM Bit	Error (%)	n	N	M	BGDM Bit	Error (%)
38400	0	23	151	0	0.00	0	36	194	0	0.01	0	73	194	0	0.01
57600	0	15	151	0	0.00	0	21	173	0	–0.01	0	58	232	0	0.00
115200	0	7	151	0	0.00	0	10	173	0	–0.01	0	21	173	0	–0.01
230400	0	3	151	0	0.00	0	10	173	1	–0.01	0	10	173	0	–0.01
460800	0	1	151	0	0.00	0	6	220	1	–0.09	0	10	173	1	–0.01

Note: This is an example when the SCR2.ABCS bit = 0 and SCR2.ABCSE bit = 0.
Note 1. It means that the bit rate modulation function is disable. (SCR2.BRME bit = 0, M = 256)

33.2.8 Control Register 3 (SCR3)

Address(es): RSCI8.SCR3 000A 1414h, RSCI9.SCR3 000A 1494h, RSCI11.SCR3 000E 2094h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	BLK	GM	—	—	CKE[1:0]		—	—	DEEN	FM	MP	MOD[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RXDESEL	STOP	DINV	DDIR	—	—	CHR[1:0]		SYNDIS	—	—	—	—	—	CPOL	CPHA
Value after reset:	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	CPHA	Clock Phase Select	(Valid in clock synchronous mode and simple SPI mode. Set this bit only when SCR0.TE bit = 0 and RE bit = 0.) 0: Data is sampled at an odd edge and changes at an even edge. (Clock is delayed.) 1: Data changes at an odd edge and is sampled at an even edge. (Clock is not delayed.)	R/W *1
b1	CPOL	Clock Polarity Select	(Valid in clock synchronous mode and simple SPI mode. Set this bit only when SCR0.TE bit = 0 and RE bit = 0.) 0: SCKn in idle state is 0. 1: SCKn in idle state is 1.	R/W *1
b6 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b7	SYNDIS	Synchronizer Disable	Set this bit to 1.	R/W
b9, b8	CHR[1:0]	Character Length Select	(Valid in asynchronous mode and manchester mode) *2 Select the data length for transmission and reception. b9 b8 0 0: Transmit/receive in 9-bit data length 0 1: Transmit/receive in 9-bit data length 1 0: Transmit/receive in 8-bit data length (initial value) 1 1: Transmit/receive in 7-bit data length*3	R/W *1
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R
b12	DDIR	Transfer Data Direction Select	0: MSB first 1: LSB first Set this bit to 0 in simple I2C mode and set this bit to 1 in extended serial mode.	R/W *1
b13	DINV	Transfer Data Invert	0: TDR register contents are transmitted to TSR register as they are. RSR register contents are stored to RDR register as they are. 1: TDR register contents are inverted before being transmitted to TSR register. RSR register contents are inverted and stored to RDR register. Set this bit to 0 in simple I2C mode. The level of communication pins (RXDn/TXDn) are controlled by combination of this bit and SCR1.TINV/RINV. Please refer to Figure 33.2 for details.	R/W *1
b14	STOP	Stop Bit Length Select	(Valid in asynchronous mode, manchester mode, extended serial mode) 0: 1 stop bit/break delimiter length is 1bit 1: 2 stop bits/break delimiter length is 2bits	R/W *1
b15	RXDESEL	Asynchronous Start Bit Edge Detection Select	(Valid only in asynchronous mode) Set this bit to 1 in extended serial mode. 0: The low level on the RXDn pin is detected as the start bit. 1: A falling edge on the RXDn pin is detected as the start bit.	R/W *1

Bit	Symbol	Bit Name	Description	R/W
b18 to b16	MOD[2:0]	Communication Mode Select	Select the RSCI communication mode. b18 b16 0 0 0: Asynchronous mode 0 0 1: Smart card interface mode 0 1 0: Clock synchronous mode 0 1 1: Simple SPI mode 1 0 0: Simple I^2C mode 1 0 1: Manchester mode 1 1 0: Extended serial mode 1 1 1: Setting prohibited	R/W *1
b19	MP	Multi-Processor Mode	(Valid in asynchronous mode, manchester mode) 0: Multi-processor communications function is disabled 1: Multi-processor communications function is enabled	R/W *1
b20	FM	FIFO Mode Select	(Valid in asynchronous mode (including multi-processor mode), clock synchronous mode, simple SPI mode) 0: TDR register, RDR register is non-FIFO buffer configuration 1: TDR register, RDR register is FIFO buffer configuration	R/W *1
b21	DEEN	Driver Control Function Enable	(Valid only in asynchronous mode) 0: RS-485 driver control function disable. 1: RS-485 driver control function enable.	R/W *1
b23, b22	—	Reserved	These bits are read as 0. The write value should be 0.	R
b25, b24	CKE[1:0]	Clock Enable	In the case of asynchronous mode b25 b24 0 0: On-chip baud rate generator The SCKn pin is available for use as an I/O port in accord with the I/O port settings. 0 1: On-chip baud rate generator The clock with the same frequency as the bit rate is output from the SCKn pin. 1 x: External clock • When using the external clock 16 times the bit rate should be input from the SCKn pin when SCR2.ABCS bit is 0. Input a clock signal with a frequency 8 times the bit rate when the SCR2.ABCS bit is 1. In the case of manchester mode and extended serial mode b25 b24 0 0: On-chip baud rate generator The SCKn pin functions as I/O port. Settings other than above are prohibited. In the case of clock synchronous mode and simple SPI mode b25 b24 0 x: Internal clock (master operation) The SCKn pin functions as the clock output pin. 1 x: External clock (slave operation) The SCKn pin functions as the clock input pin. In the case of smart card interface mode When SCR3.GM bit = 0 b25 b24 0 0: Output disabled (The SCKn pin is available for use as an I/O port in accord with the I/O port settings.) 0 1: Clock output 1 x: Prohibited When SCR3.GM bit = 1 b25 b24 0 0: Output fixed low 0 1: Clock output 1 0: Output fixed high 1 1: Clock output	R/W *1
b26	—	Reserved	Set this bit to 0.	R/W
b27	—	Reserved	This bit is read as 0. The write value should be 0.	R
b28	GM	GSM Mode	(Valid only in smart card interface mode) 0: Non-GSM mode operation 1: GSM mode operation	R/W *1

Bit	Symbol	Bit Name	Description	R/W
b29	BLK	Block Transfer Mode	(Valid only in smart card interface mode) 0: Non-block transfer mode operation 1: Block transfer mode operation	R/W *1
b31, b30	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Writable only when SCR0.TE bit = 0 and SCR0.RE bit = 0.
Note 2. In other than asynchronous mode and manchester mode, this bit setting is invalid and a fixed data length of 8 bits is used. Set these bits to 10b in extended serial mode.
Note 3. LSB first is fixed and the MSB (bit 7) in TDR register is not transmitted in transmission.

CPHA Bit (Clock Phase Select)

This bit selects the phase of the clock signal output through the SCKn pin. Refer to Figure 33.108 for details.
Set the bit to 1 in other than simple SPI mode and clock synchronous mode.

CPOL Bit (Clock Polarity Select)

This bit selects the polarity of the clock signal output through the SCKn pin. Refer to Figure 33.108 for details.
Set the bit to 1 in other than simple SPI mode and clock synchronous mode.

CHR[1:0] Bits (Character Length Select)

Selects the data length for transmission and reception.
Except of asynchronous mode and manchester mode, a fixed data length of 8 bits is used.

DDIR Bit (Transfer Data Direction Select)

Select whether to transmit/receive data in MSB first or LSB first.

DINV Bit (Transfer Data Invert)

DINV bit can invert the transmit data bit from TDR register to TSR register, and also can invert the received data from RSR register to RDR register. This bit does not affect the logic level of the parity bit. To invert the parity bit, invert the SCR1.PM bit.

STOP Bit (Stop Bit Length Select)

Selects the stop bit length in transmission.
In reception, only the first stop bit is checked regardless of this bit setting. If the second stop bit is 0, it is treated as the start bit of the next transmit frame.
In addition, it is used as break delimiter length setting when sending Start Frame in extended serial mode.

RXDESEL Bit (Asynchronous Start Bit Edge Detection Select)

Selects the detection method of the start bit for reception in asynchronous mode. When a break occurs, data receiving operation depends on the settings of this bit. Set this bit to 1 when reception should be stopped while a break occurs or when reception should be started without retaining the RXDn pin input at high level for the period of one data frame or longer after completion of the break.
Set this bit to 1 in extended serial mode. Set this bit to 0 in modes except of asynchronous mode and extended serial mode.

MOD[2:0] Bits (Communication Mode Select)

Selects the RSCI communication mode.

Table 33.27 Relationship between Communication Mode Selection Bits (MOD[2:0]), Other Operation Mode Setting Bits

Communication mode	Asynchronous				Smart Card I/F	Clock Synchronous		Simple SPI				Simple I²C		Manchester		Extended Serial
SCR3.MOD[2:0]	000b				001b	010b		011b				100b		101b		110b
SCR3.MP	0		1		—	—		—				—		0	1	—
SCR3.FM	0	1	0	1	—	0	1	0	1			—		—		—
SCR3.DEEN	0	1	0	1	0	1	0	1	—			—		—		—
SCR3.SSE	—				—	—		0	1	0	1	—		—		—

—: Prohibited setting

MP Bit (Multi-Processor Mode)

Disables/enables the multi-processor communications function. The settings of the PE bit and PM bit are invalid in multi-processor mode.

FM Bit (FIFO Mode Select)

When the FM bit is set to 1, the TDR register/RDR register switches to FIFO configuration, and transmit FIFO (TDR register)/receive FIFO (RDR register) can be used for serial transmission/reception.

DEEN Bit (Driver Control Function Enable)

Select RS-485 Driver control function disable or enable.

CKE[1:0] Bits (Clock Enable)

These bits select the clock source and SCKn pin function.

In smart card interface mode, these bits control the clock output from the SCKn pin.

In GSM mode, clock output can be dynamically switched. For details, refer to section 33.7.8, Clock Output Control.

GM Bit (GSM Mode)

Setting this bit to 1 allows GSM mode operation.

In GSM mode, the SSR.TEND flag set timing is put forward to 11.0 etu (elementary time unit = 1-bit transfer time) from the start and the clock output control function is appended. For details, refer to section 33.7.6, Serial Data Transmission (Except in Block Transfer Mode) and section 33.7.8, Clock Output Control.

BLK Bit (Block Transfer Mode)

Setting this bit to 1 allows block transfer mode operation.

For details, refer to section 33.7.3, Block Transfer Mode.

33.2.9 Control Register 4 (SCR4)

Address(es): RSCI8.SCR4 000A 1418h, RSCI9.SCR4 000A 1498h, RSCI11.SCR4 000E 2098h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TTMG[3:0]				RTMG[3:0]				—	—	—	—	—	—	TTADJ	RTADJ
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CMPD[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b8 to b0	CMPD[8:0]	Compare Match Data	(Valid only in asynchronous mode) Set the compared data when using data match detection function	R/W *1
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	RTADJ	Receive Data Sampling Timing Adjustment	(Valid in asynchronous mode using internal clock, extended serial mode using internal clock, clock synchronous mode operating as master, simple SPI mode operating as master) 0: Adjust sampling timing disable. 1: Adjust sampling timing enable.	R/W *1
b17	TTADJ	Transmit Signal Transition Timing Adjustment	(Valid only in asynchronous mode using internal clock) 0: Adjust transmit timing disable. 1: Adjust transmit timing enable.	R/W *1
b23 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R
b27 to b24	RTMG[3:0]	Receive Data Sampling Timing Select	In the case of asynchronous mode and extended serial mode b27 b24 1 1 1 1: Data are sampled 7 clocks earlier than default point. 1 1 1 0: Data are sampled 6 clocks earlier than default point. 1 1 0 1: Data are sampled 5 clocks earlier than default point. 1 1 0 0: Data are sampled 4 clocks earlier than default point. 1 0 1 1: Data are sampled 3 clocks earlier than default point. 1 0 1 0: Data are sampled 2 clocks earlier than default point. 1 0 0 1: Data are sampled 1 clocks earlier than default point. x 0 0 0: Data are sampled at default point. 0 0 0 1: Data are sampled 1 clock later than default point. 0 0 1 0: Data are sampled 2 clock later than default point. 0 0 1 1: Data are sampled 3 clock later than default point. 0 1 0 0: Data are sampled 4 clock later than default point. 0 1 0 1: Data are sampled 5 clock later than default point. 0 1 1 0: Data are sampled 6 clock later than default point. 0 1 1 1: Data are sampled 7 clock later than default point. In the case of clock synchronous mode and simple SPI mode b27 b24 0 0 0 0: 1 PCLK delay 0 0 0 1: 2 PCLK delay 0 0 1 0: 3 PCLK delay 0 0 1 1: 4 PCLK delay Settings other than above are prohibited.	R/W *1

Bit	Symbol	Bit Name	Description	R/W
b31 to b28	TTMG[3:0]	Transmit Signal Transition Timing Select	b31 b28 1 1 1 1: Delays the 1 to 0 transitions for 7 clocks. 1 1 1 0: Delays the 1 to 0 transitions for 6 clocks. 1 1 0 1: Delays the 1 to 0 transitions for 5 clocks. 1 1 0 0: Delays the 1 to 0 transitions for 4 clocks. 1 0 1 1: Delays the 1 to 0 transitions for 3 clocks. 1 0 1 0: Delays the 1 to 0 transitions for 2 clocks. 1 0 0 1: Delays the 1 to 0 transitions for 1 clocks. x 0 0 0: Does not change the waveform. 0 0 0 1: Delays the 0 to 1 transitions for 1 clock. 0 0 1 0: Delays the 0 to 1 transitions for 2 clock. 0 0 1 1: Delays the 0 to 1 transitions for 3 clock. 0 1 0 0: Delays the 0 to 1 transitions for 4 clock. 0 1 0 1: Delays the 0 to 1 transitions for 5 clock. 0 1 1 0: Delays the 0 to 1 transitions for 6 clock. 0 1 1 1: Delays the 0 to 1 transitions for 7 clock.	R/W *1

Note 1. Writable only when SCR0.TE bit = 0 and SCR0.RE bit = 0.

CMPD[8:0] Bits (Compare Match Data)

Set the comparison data for receive data, when data match detection function is enabled (SCR0.DCME bit = 1). SCR4.CMPD[8:0] bits should be written while SCR0.DCME bit is 0.

For the comparison data, it can select length from 3 types, they are CMPD[6:0] with 7bit length enable, CMPD[7:0] with 8bit, and CMPD[8:0] with 9bit length.

RTADJ Bit (Receive Data Sampling Timing Adjustment)

When this bit is 1, the receive sampling timing adjustment function is enabled. Control is different in asynchronous mode, extended serial mode, clock synchronous mode, and simple SPI mode.

In asynchronous mode using internal clock, refer to section 33.3.10, Receive Data Sampling Timing Adjustment for details. The operation when the extended serial mode internal clock is selected is the same as when the asynchronous clock internal clock is selected.

In clock synchronous mode as master, simple SPI mode operating as master, refer to section 33.10.7, Reception Sampling Timing Adjustment Function in Clock Synchronous Mode with Internal Clock Used for details. Only the digital delay of the master mode receive sampling clock (MRCLK) can be controlled by this bit. MRCLK analog delay cannot be controlled.

TTADJ Bit (Transmit Signal Transition Timing Adjustment)

When this bit is 1, the transmit signal transition timing adjustment function is enabled. The transmit signal transition timing adjustment function can adjust the edge timing of the waveform output from the TXDn pin. Refer to section 33.3.11, Transmit Data Transition Timing Adjustment for details.

RTMG[3:0] Bits (Receive Data Sampling Timing Select)

When the RTADJ bit is 1, the receive sampling timing can be adjusted according to this bit setting value. The adjustment value in the synchronous mode and the extended serial mode is the base clock × RTMG[2:0] setting value.

TTMG[3:0] Bits (Transmit Signal Transition Timing Select)

The edge timing of the TXDn pin specified by the TTMG[3:0] bits is adjusted by the base clock × TTMG[2:0] setting value. Make sure that the TTMG[2:0] bit setting is less than the number of base clock cycles for 1-bit period.

33.2.10 I²C Mode Register (SIMR)

Address(es): RSCI8.SIMR 000A 1420h, RSCI9.SIMR 000A 14A0h, RSCI11.SIMR 000E 20A0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	IICSCLS[1:0]		IICSDAS[1:0]		—	IICSTPREQ	IICRSTAREQ	IICSTAREQ
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	IICACKT	—	—	—	IICCSC	IICINTM	—	—	—	IICDL[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	IICDL[4:0]	SDA Output Delay Select	(Cycles below are of the clock signal from the on-chip baud rate generator.) b4 b0 0 0 0 0 0: No output delay 0 0 0 0 1: 0 to 1 cycle 0 0 0 1 0: 1 to 2 cycles 0 0 0 1 1: 2 to 3 cycles 0 0 1 0 0: 3 to 4 cycles 0 0 1 0 1: 4 to 5 cycles : : 1 1 1 1 0: 29 to 30 cycles 1 1 1 1 1: 30 to 31 cycles	R/W *1
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	IICINTM	I²C Interrupt Mode Select	0: Use ACK/NACK interrupts. 1: Use reception and transmission interrupts	R/W *1
b9	IICCSC	Clock Synchronization	0: No synchronization with the clock signal 1: Synchronization with the clock signal	R/W *1
b12 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R
b13	IICACKT	ACK Transmission Data	0: ACK transmission 1: NACK transmission and reception of ACK/NACK	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	IICSTAREQ	Start Condition Generation	0: A start condition is not generated. 1: A start condition is generated.*2, *4, *5, *6	R/W
b17	IICRSTAREQ	Restart Condition Generation	0: A restart condition is not generated. 1: A restart condition is generated.*3, *4, *5, *6	R/W
b18	IICSTPREQ	Stop Condition Generation	0: A stop condition is not generated. 1: A stop condition is generated.*3, *4, *5, *6	R/W
b19	—	Reserved	This bit is read as 0. The write value should be 0.	R
b21, b20	IICSDAS[1:0]	SDA Output Select	b21 b20 0 0: Serial data output 0 1: Generate a start, restart, or stop condition. 1 0: Output the low level on the SSDAn pin. 1 1: Place the SSDAn pin in the high-impedance state.	R/W
b23, b22	IICSCLS[1:0]	SCL Output Select	b23 b22 0 0: Serial clock output 0 1: Generate a start, restart, or stop condition. 1 0: Output the low level on the SSCLn pin. 1 1: Place the SSCLn pin in the high-impedance state.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Writable only when SCR0.TE bit = 0 and SCR0.RE bit = 0.
Note 2. In the bus free state, perform the start condition generation.
Note 3. In the bus busy state, perform restart or stop condition generation when the SSCLn pin after acknowledgment described in Figure 33.78 and Figure 33.79 is low level.
Note 4. Do not set more than one from among the IICSTAREQ, IICRSTAREQ, and IICSTPREQ bits to 1 at a given time.

Note 5. Execute the generation of a condition after the value of the IICSTIF flag is 0.
Note 6. Do not write 0 to this bit while it is 1. Generation of a condition is suspended by writing 0 to this bit while it is 1.

IICDL[4:0] Bits (SDA Output Delay Select)

These bits are used to set a delay for output on the SSDAn pin relative to the falling edge of the output on the SSCLn pin. The available delay settings range from no delay to 31 cycles, with the clock signal from the on-chip baud rate generators the base. The signal obtained by frequency-dividing PCLK by the divisor set in SCR2.CKS[1:0] bits is supplied as the clock signal from the on-chip baud rate generator.
Set these bits to 00000b unless operation is in simple I^2C mode. In simple I^2C mode, set the bits to a value in the range from 00001b to 11111b.

IICINTM Bit (I^2C Interrupt Mode Select)

This bit selects the sources of interrupt requests in simple I^2C mode.

IICCSC Bit (Clock Synchronization)

Set the IICCSC bit to 1 if the internally generated SCL signal is to be synchronized when the SSCLn pin has been placed at the low level in the case of a wait inserted by the other device, etc.
The internal SCL signal is not synchronized if the IICCSC bit is 0. The internal SCL signal is generated in accordance with the rate selected in the BRR[7:0] bits regardless of the level being input on the SSCLn pin.
Set the IICCSC bit to 1 except during debugging.

IICACKT Bit (ACK Transmission Data)

Transmitted data contains ACK bits. Set this bit to 1 when ACK and NACK bits are received.

IICSTAREQ Bit (Start Condition Generation)

When a start condition is to be generated, set both the IICSDAS[1:0] and IICSCLS[1:0] bits to 01b as well as setting the IICSTAREQ bit to 1.
If you want to generate the start condition after generating the stop condition, start the generation of the start condition with a half cycle period of the bit rate from the stop condition generation interrupt request output.
[Setting condition]

- Writing 1 to the bit

[Clearing condition]

- Completion of generation of the start condition

IICRSTAREQ Bit (Restart Condition Generation)

When a restart condition is to be generated, set both the IICSDAS[1:0] and IICSCLS[1:0] bits to 01b as well as setting the IICRSTAREQ bit to 1.
[Setting condition]

- Writing 1 to the bit

[Clearing condition]

- Completion of generation of the restart condition

IICSTPREQ Bit (Stop Condition Generation)

When a stop condition is to be generated, set both the IICSDAS[1:0] and IICSCLS[1:0] bits to 01b as well as setting the IICSTPREQ bit to 1.

[Setting condition]

- Writing 1 to the bit

[Clearing condition]

- Completion of generation of the stop condition

IICSDAS[1:0] Bits (SDA Output Select)

These bits control output from the SSDAn pin.

IICSCLS[1:0] Bits (SCL Output Select)

These bits control output from the SSCLn pin.

33.2.11 FIFO Control Register (FCR)

Address(es): RSCI11.FCR 000E 20A4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	RSTRG[4:0]					RFRST	—	—	RTRG[4:0]				
Value after reset:	0	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TFRST	—	—	TTRG[4:0]					—	—	—	—	—	—	—	DRES
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DRES	Receive Data Ready Interrupt Select	(Valid in asynchronous mode) This bit select the interrupt request for a reception data ready detection. 0: Reception data full interrupt (RXI) 1: Receive error interrupt (ERI)	R/W *1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R
b12 to b8	TTRG[4:0]	Transmit FIFO Threshold Setting	(Valid in asynchronous mode (including multi-processor mode), clock synchronous mode, simple SPI mode) b12 b8 0 0 0 0 0: Trigger number 0 : : 1 1 1 1 1: Trigger number 31	R/W *1
b14, b13	—	Reserved	These bits are read as 0. The write value should be 0.	R
b15	TFRST	Transmit FIFO Reset	This bit enables only when SCR3.FM bit is 1. The read value is always 0. 0: It is invalid. It does not affect the operation. 1: The number of data stored in transmit FIFO (TDR register) are made 0 The read value is always 0.	W*1
b20 to b16	RTRG[4:0]	Receive FIFO Threshold Setting	(Valid in asynchronous mode (including multi-processor mode), clock synchronous mode, simple SPI mode) b20 b16 0 0 0 0 0: Trigger number 0 : : 1 1 1 1 1: Trigger number 31	R/W *1
b22, b21	—	Reserved	These bits are read as 0. The write value should be 0.	R
b23	RFRST	Receive FIFO Reset	This bit enables only when SCR3.FM bit is 1. The read value is always 0. 0: It is invalid. It does not affect the operation. 1: The number of data stored in receive FIFO (RDR register) are made 0 The read value is always 0.	W*1
b28 to b24	RSTRG[4:0]	RTS# Output Threshold Setting	(Valid in asynchronous mode (including multi-processor mode), clock synchronous mode) This bit enables only when SCR3.FM bit = 1, SCR1.CTSE bit = 0, and SCR0.SSE bit = 0. b28 b24 0 0 0 0 0: Trigger number 0 : : 1 1 1 1 1: Trigger number 31	R/W *1
b31 to b29	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Writable only when SCR0.TE bit = 0 and SCR0.RE bit = 0.

DRES Bit (Receive Data Ready Interrupt Select)

Select whether the detection of receive data ready (RFSR.DR flag = 1) is the cause of RXI interrupt request or the cause of ERI interrupt request.

TTRG[4:0] Bits (Transmit FIFO Threshold Setting)

The TDFE flag is set to 1 when the quantity of transmit data in the transmit FIFO (TDR register) is equal to or less than the specified transmit triggering number. If SCR0.TIE bit = 1, TXI interrupt request is occurred.

TFRST Bit (Transmit FIFO Reset)

When the TFRST bit is set to 1, the number of the transmission data stored in transmit FIFO (TDR register) is made 0.

RTRG[4:0] Bits (Receive FIFO Threshold Setting)

The SSR.RDRF flag is set to 1 when the quantity of receive data in the receive FIFO (RDR register) is equal to or greater than the specified receive triggering number. If SCR0.RIE bit = 1, RXI interrupt request is occurred. When FCR.RTRG[4:0] bits are set to 0, RDRF flag is set if the quantity of data in receive FIFO is greater than or equal to 1.

RFRST Bit (Receive FIFO Reset)

When the RFRST bit is set to 1, the number of the reception data stored in receive FIFO (RDR register) is made 0.

RSTRG[4:0] Bits (RTS# Output Threshold Setting)

When the quantity of receive data stored in the receive FIFO (RDR register) is equal to or greater than this number, the RTSn# signal is in the High state. When FCR.RSTRG[4:0] bits are set to 0, RTSn# is in the high state if the quantity of data in receive FIFO is greater than or equal to 1.

33.2.12 Manchester Mode Control Register (MMCR)

Address(es): RSCI9.MMCR 000A 14ACh, RSCI11.MMCR 000E 20ACh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	SBERIE	SYERIE	PFERIE	—	—	RPPAT[1:0]		RPLEN[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	TPPAT[1:0]		TPLEN[3:0]				—	SBLEN	SYNCE	SBPTN	—	SADJE	ENCS	DECS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DECS	Decoding Convention Select	Sets the polarity of the received manchester code 0: Low to high transition is decoded to a logic 0 and high to low transition is decoded to a logic 1. 1: high to low transition is decoded to a logic 0 and low to high transition is decoded to a logic 1.	R/W *1
b1	ENCS	Encoding Convention Select	Sets the polarity of the transmit manchester code 0: Logic 0 is encoded to a low to high transition and logic 1 is encoded to a high to low transition. 1: Logic 0 is encoded to a high to low transition and logic 1 is encoded to a low to high transition.	R/W *1
b2	SADJE	Receive Timing Self Adjustment Enable	Sets the receive retiming function 0: Disables the receive retiming function 1: Enables the receive retiming function	R/W *1
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R
b4	SBPTN	Start Bit Pattern Select	Sets the Sync type of the start bit(s) in the manchester code When the start bit area consists of one bit. (SBLEN bit = 0) • When transmitting 0: The start bit is added as a low to high transition. 1: The start bit is added as a high to low transition. • When receiving 0: Only when the start bit is a low to high transition, the data is received. The other cases are judged as an error. 1: Only when the start bit is a high to low transition, the data is received. The other cases are judged as an error. When the start bit area consists of three bits. (SBLEN bit = 1) • When transmitting 0: The start bits are added as a low to high transition. (DATA Sync) 1: The start bits are coded as a high to low transition. (COMMAND Sync) • When receiving When the start bit area consists of three bits, data is received regardless of the value of this bit.	R/W *1
b5	SYNCE	Sync Enable	0: The start bit pattern is set with the SBPTN bit. 1: The start bit pattern is set with the SYNC bit.	R/W *1
b6	SBLEN	Start Bit Length Select	0: The start bit area consists of one bit. 1: The start bit area consists of three bits. (COMMAND Sync or DATA Sync)	R/W *1
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R
b11 to b8	TPLEN[3:0]	Transmit Preface Length Setting	Set the preface length of the transmit data in manchester mode 0: Disables the transmit preface generation 1 to 15: Transmit preface length (bit length)	R/W *1

Bit	Symbol	Bit Name	Description	R/W
b13, b12	TPPAT[1:0]	Transmit Preface Pattern Select	Set the preface pattern of the transmit data b13 b12 0 0: ALL ZERO 0 1: ZERO ONE 1 0: ONE ZERO 1 1: ALL ONE	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R
b19 to b16	RPLEN[3:0]	Receive Preface Length Setting	Set the preface length in received frames when manchester mode is enabled 0: Disables the receive preface generation 1 to 15: Receive preface length (bit length)	R/W *1
b21, b20	RPPAT[1:0]	Receive Preface Pattern Select	Set the preface pattern of received frames b21 b20 0 0: ALL ZERO 0 1: ZERO ONE 1 0: ONE ZERO 1 1: ALL ONE	R/W *1
b23, b22	—	Reserved	These bits are read as 0. The write value should be 0.	R
b24	PFERIE	Preface Error Interrupt Enable	Specifies whether to handle a preface error as an interrupt source 0: Does not handle a preface error as an interrupt source 1: Handles a preface error as an interrupt source	R/W *1
b25	SYERIE	Sync Error Interrupt Enable	Specifies whether to handle a receive Sync error as an interrupt source 0: Does not handle a receive Sync error as an interrupt source 1: Handles a receive Sync error as an interrupt source	R/W *1
b26	SBERIE	Start Bit Error Interrupt Enable	Specifies whether to handle a start bit error as an interrupt source 0: Does not handle a start bit error as an interrupt source 1: Handles a start bit error as an interrupt source	R/W *1
b31 to b27	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Writable only when SCR0.TE bit = 0 and SCR0.RE bit = 0.

DECS Bit (Decoding Convention Select)

This bit sets the polarity of the received manchester code. For details on the data reception, see section 33.5.7, Serial Data Reception in Manchester Mode.

ENCS Bit (Encoding Convention Select)

This bit sets the polarity of the transmit manchester code. For details on the data transmission, see section 33.5.6, Serial Data Transmission in Manchester Mode.

SADJE Bit (Receive Timing Self Adjustment Enable)

This bit sets the receive retiming function in manchester mode.

For information on the receive retiming function, see section 33.5.9, Receive Retiming.

SBPTN Bit (Start Bit Pattern Select)

This bit is valid when the SYNCE bit of this register is set to 0.

The Sync type can be set by combining this bit and the SBLEN bit.

For the start bit area determined by the combination of this bit and the SBLEN bit, see Figure 33.36 and Figure 33.37.

SYNCE Bit (Sync Enable)

This bit is valid when the SBLEN bit of this register is set to 1. This bit determines the destination to be referred to for setting the Sync type of the start bit area added to manchester frames.
When this bit is set to 0, the SBPTN bit of this register is referred to.
When this bit is set to 1, the SYNC bit in the TDR register is referred to.

SBLEN Bit (Start Bit Length Select)

This bit sets the start bit area in manchester frames.
When this bit is set to 1, the start bit area added to each frame consists of three bits, and the SYNCE and SBPTN bits in this register are valid.
When this bit is set to 0, the start bit area added to each frame consists of one bit.

TPLEN[3:0] Bits (Transmit Preface Length Setting)

These bits set the preface bit length of the transmit data in manchester mode.
The settable range is 0h to Fh (0 to 15). 0h disables the transmit preface, which is not added.

TPPAT[1:0] Bits (Transmit Preface Pattern Select)

These bits set one of the four preface patterns in manchester mode. For the transmit data when the TPPAT[1:0] bits are set, see Figure 33.35.
When these bits are set to 00b, the preface area is set to all zeros.
When these bits are set to 01b, the preface area is set to the zero-one-zero-one pattern.
When these bits are set to 10b, the preface area is set to the one-zero-one-zero pattern.
When these bits are set to 11b, the preface area is set to all ones.

RPLEN[3:0] Bits (Receive Preface Length Setting)

These bits set the preface bit length of the received frames in manchester mode.
The settable range is 0h to Fh (0 to 15). 0h disables the receive preface, which is not added. When 1h to Fh is set, the set value is handled as the receive preface bit length.

RPPAT[1:0] Bits (Receive Preface Pattern Select)

These bits set one of the four preface patterns in manchester mode. For the transmit and receive data when the TPPAT[1:0] bits are set, see Figure 33.35.
When these bits are set to 00b, the preface area is handled as all zeros.
When these bits are set to 01b, the preface area is handled as the zero-one-zero-one pattern.
When these bits are set to 10b, the preface area is handled as the one-zero-one-zero pattern.
When these bits are set to 11b, the preface area is handled as all ones.

PFERIE Bit (Preface Error Interrupt Enable)

This bit specifies whether to handle a preface error as an interrupt source.
When it is set to 0, a preface error is not handled as an interrupt source. When it is set to 1, a preface error is handled as an interrupt source.

SYERIE Bit (Sync Error Interrupt Enable)

This bit specifies whether to handle a receive Sync error as an interrupt source.
When it is set to 0, a receive Sync error is not handled as an interrupt source. When it is set to 1, a receive Sync error is handled as an interrupt source.

SBERIE Bit (Start Bit Error Interrupt Enable)

This bit specifies whether to handle a start bit error as an interrupt source.

When it is set to 0, a start bit error is not handled as an interrupt source. When it is set to 1, a start bit error is handled as an interrupt source.

33.2.13 DE Signal Control Register (DECR)

Address(es): RSCI8.DECR 000A 1430h, RSCI9.DECR 000A 14B0h, RSCI11.DECR 000E 20B0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	DEHLD[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	DESU[4:0]					—	—	—	—	—	—	—	DELVL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DELVL	DE Signal Active Level Select	(Valid only in asynchronous mode) 0: The DE signal is active high. 1: The DE signal is active low.	R/W *1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R
b12 to b8	DESU[4:0]	DE Signal Setup Time Setting	(Valid only in asynchronous mode) Set the DE signal setup time in the number of the base clock cycles. The bit setting is enabled when the SCR3.DEEN bit is 1. Do not set 00000b.	R/W *1
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R
b20 to b16	DEHLD[4:0]	DE Signal Hold Time Setting	(Valid only in asynchronous mode) Set the DE signal hold time in the number of the base clock cycles. The bit setting is enabled when the SCR3.DEEN bit is 1. Do not set 00000b.	R/W *1
b31 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Writable only when SCR0.TE bit = 0 and SCR0.RE bit = 0.

DELVL Bit (DE Signal Active Level Select)

Select the active level of the DE (driver enable) signal.

DESU[4:0] Bits (DE Signal Setup Time Setting)

Set the DE signal setup time (time from the assertion of the DE signal to the start of transmission of the start bit). It is expressed in the number of the base clock cycles (1/6, 1/8, or 1/16 bit period). The actual transmission of the start bit starts after the setup time and transmission wait time have elapsed.

DEHLD[4:0] Bits (DE Signal Hold Time Setting)

Set the DE signal hold time (time from the completion of transmission of the stop bit of the last transmission message to negation of the DE signal). It is expressed in the number of the base clock cycles (1/6, 1/8, or 1/16 bit period).
If the transmission data is written during the hold time, transmit starting operation is different depends on the writing timing (following two cases: the transmission of the start bit starts after the transmission wait time has elapsed without negating the DE signal, or it starts after the DE signal is negated and asserted again and then the setup time and transmission wait time have elapsed.).

33.2.14 Extended Serial Mode Control Register 0 (XCR0)

Address(es): RSCI9.XCR0 000A 14B4h, RSCI11.XCR0 000E 20B4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	BCCS[1:0]		—	AEDIE	COFIE	BFDIE	—	—	BCDIE	BFOIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PIBS[2:0]			PIBE	CF1DS[1:0]		CF0RE	BFE	—	—	—	—	—	—	TCSS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TCSS[1:0]	Timer Count Clock Source Select	(Valid in extended serial mode) Select the clock source of the timer in the extended serial module. b1 b0 0 0: PCLK 0 1: PCLK/4 1 0: PCLK/16 1 1: PCLK/64	R/W *1, *2
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	BFE	Break Field Detection Enable	Set the presence or absence of Break Field of Start Frame. 0: No Break Field 1: With Break Field	R/W *1, *4
b9	CF0RE	Control Field 0 Reception Enable	Set the presence or absence of Control Field 0 of Start Frame 0: No Control Field 0 1: With Control Field 0	R/W *1, *4
b11, b10	CF1DS[1:0]	Control Field 1 Compare Data Select	Select the compare data for Control Field 1 b11 b10 0 0: Select XCR1.PCF1D[7:0] bits as the compare data 0 1: Select XCR1.SCF1D[7:0] bits as the compare data 1 0: Select both XCR1.PCF1D[7:0] bits and XCR1.SCF1D[7:0] bits as the compare data 1 1: Prohibition	R/W *1, *4
b12	PIBE	Priority Interrupt Bit Enable	0: Priority interrupt bit disable 1: Priority interrupt bit enable	R/W *1, *4
b15 to b13	PIBS[2:0]	Priority Interrupt Bit Select	Specify one of bits 0 to 7 of Control Field 1 as the priority interrupt bit. b15 b13 0 0 0: Bit 0 of Control Field 1 0 0 1: Bit 1 of Control Field 1 0 1 0: Bit 2 of Control Field 1 0 1 1: Bit 3 of Control Field 1 1 0 0: Bit 4 of Control Field 1 1 0 1: Bit 5 of Control Field 1 1 1 0: Bit 6 of Control Field 1 1 1 1: Bit 7 of Control Field 1	R/W *1, *4
b16	BFOIE	Break Field Low Width Output Complete Interrupt Enable	Select whether to include Break Field transmission completion as a TXI interrupt factor. 0: Break Field transmission completion is not included in TXI interrupt factor 1: Break Field transmission completion is included in TXI interrupt factor	R/W *1
b17	BCDIE	Bus Collision Detected Interrupt Enable	Select whether to output an ERI interrupt when a bus collision is detected. 0: Bus conflict detection is not included in ERI interrupt factor 1: Bus conflict detection is included in ERI interrupt factor	R/W *1
b19, b18	—	Reserved	These bits are read as 0. The write value should be 0.	R

Bit	Symbol	Bit Name	Description	R/W
b20	BFDIE	Break Field Low Width Detected Interrupt Enable	Select whether to output a BFD interrupt when a Break Field is detected. 0: Break Field detection interrupt disable 1: Break Field detection interrupt enable	R/W *1
b21	COFIE	Count Overflow Interrupt Enable	Select whether to include counter overflow as an ERI interrupt factor. 0: Counter overflow is not included in ERI interrupt factor 1: Counter overflow is included in ERI interrupt factor	R/W *1
b22	AEDIE	Effective Edge Detected Interrupt Enable	Select whether to output an AED interrupt when a valid edge is detected. 0: Active edge detection interrupt disable 1: Active edge detection interrupt enable	R/W *1
b23	—	Reserved	This bit is read as 0. The write value should be 0.	R
b25, b24	BCCS[1:0]	Bus Collision Detection Clock Select	Select the sampling clock for the bus conflict detection circuit. When SCR2.ABCS bit = 1, setting BCCS[1:0] bits = 1x is prohibited. b25 b24 0 0: Base clock*3 0 1: Base clock/2 1 0: Base clock/4 1 1: Prohibition	R/W *1
b31 to b26	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Writable only when SCR0.TE bit = 0 and SCR0.RE bit = 0.
Note 2. Rewrite the TCSS[1:0] bits only when the timer is stopped (XCR1.TCST bit = 0, XCR1.SDST bit = 0, and XCR1.BRME bit = 0).
Note 3. Base clock: 1/16 period of 1 bit period when SCR2.ABCS bit = 0, 1/8 period of 1 bit period when SCR2.ABCS bit = 1.
Note 4. This bit is a setting bit required for Start Frame reception operation. Rewrite this bit when Start Frame reception or transmission is not in progress (XCR1.SDST bit = 0 and XCR1.TCST bit = 0).

TCSS[1:0] Bits (Timer Count Clock Source Select)

Select clock source of timer in extended serial module.

BFE Bit (Break Field Detection Enable)

Set the presence or absence of Break Field of Start Frame.

CF0RE Bit (Control Field 0 Reception Enable)

Set the presence or absence of Control Field 0 of Start Frame.

CF1DS[1:0] Bits (Control Field 1 Compare Data Select)

Select the compare data for Control Field 1.

PIBE Bit (Priority Interrupt Bit Enable)

Select whether to enable priority interrupt bit comparison of Control Field 1. When this bit is 1, regardless of the XCR1.CF1CE[7:0] bits setting value, the bit specified in PIBS[2:0] is compared with the primary comparison data for Control Field 1 (XCR1.PCF1D[7:0] bits).

PIBS[2:0] Bits (Priority Interrupt Bit Select)

Specify bit N (N = 0 to 7) of Control Field 1 as the priority interrupt bit.

BFOIE Bit (Break Field Low Width Output Complete Interrupt Enable)

Select whether to include Break Field transmission completion as a TXI interrupt factor. Set SCR0.TIE bit = 1 and SCR3.MOD[2:0] bits = 110b, to output TXI upon completion of Break Field transmission.

BCDIE Bit (Bus Collision Detected Interrupt Enable)

Select whether to output an ERI interrupt when a bus collision is detected. In extended serial mode (SCR3.MOD[2:0] bits = 110b), ERI output control is performed with this bit. When SCR3.MOD[2:0] bits = 110b and BCDIE bit = 1, an ERI interrupt is issued when a bus collision is detected even if SCR0.RIE bit = 0.

COFIE Bit (Count Overflow Interrupt Enable)

Select whether to include counter overflow as an ERI interrupt factor. Set SCR0.RIE bit = 1 and SCR3.MOD[2:0] bits = 110b are required to output ERI upon counter overflow.

AEDIE Bit (Effective Edge Detected Interrupt Enable)

Select whether to output an AED interrupt when a valid edge is detected. To output AED with valid edge detection, XCR1.BRME bit = 1 and SCR3.MOD[2:0] bits = 110b must be set.

BCCS[1:0] Bits (Bus Collision Detection Clock Select)

Select the sampling clock for the bus conflict detection circuit.

33.2.15 Extended Serial Mode Control Register 1 (XCR1)

Address(es): RSCI9.XCR1 000A 14B8h, RSCI11.XCR1 000E 20B8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CF1CE[7:0]								SCF1D[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PCF1D[7:0]								—	—	BRME	SDST	—	—	—	TCST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TCST	Break Field Low Width Output Timer Count Start	0: Break Field transmission timer count stopped 1: Break Field transmission timer count start Do not set this bit and SDST bit to 1 at the same time.	R/W *1
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R
b4	SDST	Start Frame Detection Start	0: Start Frame/Break Field detection disabled 1: Start Frame/Break Field detection enabled Do not set this bit and TCST bit to 1 at the same time.	R/W *1
b5	BRME	Bit Rate Measurement Enable	0: Bit rate measurement disabled 1: Bit rate measurement enabled Set this bit to 1 simultaneously with the SDST bit. When this bit is set to 0, it can be set to 0 at any timing.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R
b15 to b8	PCF1D[7:0]	Primary Control Field 1 Compare Data	The priority compare data for Control Field 1	R/W *1
b23 to b16	SCF1D[7:0]	Secondary Control Field 1 Compare Data	The secondary compare data for Control Field 1	R/W *1
b31 to b24	CF1CE[7:0]	Control Field 1 Compare Enable	Select whether to compare bit N of Control Field 1. (N = 0 to 7) 0: Control Field 1 bit N compare disabled 1: Control Field 1 bit N compare enabled	R/W *1

Note 1. Writable only when SCR0.TE bit = 0 and SCR0.RE bit = 0.

TCST Bit (Break Field Low Width Output Timer Count Start)

[Clearing condition]

- When 0 is written to TCST bit. Break Field transmission timer count is stopped and TXDn pin output becomes idle level.
- When Break Field transmission for the period set in XCR2.BFLW[15:0] bits is completed.

[Setting condition]

- When 1 is written to TCST bit. Start Break Field transmission from TXDn pin. Holds 1 during Break Field transmission.

SDST Bit (Start Frame Detection Start)

When 1 is written to this bit, Start Frame detection starts. When XCR0.BFE bit = 1 is set, Break Field can be detected during Start Frame is detected and after Start frame is detected. When XCR0.BFE bit = 0 is set, Break Field is not detected.

When 0 is written to this bit, Start Frame detection and Break Field detection are stopped. However, if XSR0.RXDSF flag = 0 at the time of stop, it is not possible to stop data reception with this bit. Write 0 to SCR0.RE bit to stop the

reception operation or perform reception completion processing (clearing the SSR.RDRF flag or reading the RDR register) after reception is completed.

BRME Bit (Bit Rate Measurement Enable)

Set this bit to 1 simultaneously with the SDST bit. When this bit is set to 1, the valid edge interval of Control Field 0 and Control Field 1 data is measured.

PCF1D[7:0] Bits (Primary Control Field 1 Compare Data)

Set the priority compare data for Control Field 1.

SCF1D[7:0] Bits (Secondary Control Field 1 Compare Data)

Set the secondary compare data for Control Field 1.

CF1CE[7:0] Bits (Control Field 1 Compare Enable)

Select whether to compare bit N of Control Field 1. (N = 0 to 7)
When all of these bits are set to 0 (CF1CE[7:0] bits = 00h), it is always judged that Control Field 1 matches when reception is completed, and XSR0.CF1MF flag is set. This bit is a comparison enable with PCF1D[7:0] bits or SCF1D[7:0] bits, and it is not a priority interrupt bit comparison enable.

33.2.16 Extended Serial Mode Control Register 2 (XCR2)

Address(es): RSCI9.XCR2 000A 14BCh, RSCI11.XCR2 000E 20BCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	BFLW[15:0]															
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CF0CE[7:0]								CF0D[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	CF0D[7:0]	Control Field Compare Data	The compare data for Control Field 0	R/W *1
b15 to b8	CF0CE[7:0]	Control Field Compare Enable	Select whether to compare bit N of Control Field 0. (N = 0 to 7) 0: Control Field 0 bit N compare disabled 1: Control Field 0 bit N compare enabled	R/W *1
b31 to b16	BFLW[15:0]	Break Field Low Width Setting	This bit sets the Break Field length. The break field length is (BFLW[15:0] setting value + 1) × clock of the timer. The upper limit for setting this register is FFFEh. (Setting prohibited for FFFFh)	R/W *1

Note 1. Writable only when SCR0.TE bit = 0 and SCR0.RE bit = 0.

CF0D[7:0] Bits (Control Field Compare Data)

The compare data for Control Field 0.

CF0CE[7:0] Bits (Control Field Compare Enable)

Select whether to compare bit N of Control Field 0. (N = 0 to 7)

When all of these bits are set to 0 (CF0CE[7:0] bits = 00h), it is always judged that Control Field 0 matches when reception is completed, and the XSR0.CF0MF flag is set.

BFLW[15:0] Bits (Break Field Low Width Setting)

The BFLW[15:0] bits are 16-bit Break Field length setting bits and the initial value is FFFEh.

Set the break field length to 1 frame or more. The LIN standard stipulates that the Break Field length is 13 bits or more.

When sending the Break Field, writing 1 to the TCST bit leads RSCI to output the Break Field on the TXDn pin. At the same time, the timer starts counting with the timer count clock source selected by the XCR0.TCSS[1:0] bits. When the count value matches the value set in this register, up-counting is stopped and Break Field transmission from the TXDn pin is also stopped.

When detecting the Break Field, writing 1 to the SDST bit leads Start Frame detection to be enabled. RSCI starts counting from the negative edge of the RXDn signal. The timer count clock source is selected by the XCR0.TCSS[1:0] bits. When the count value matches the value set in this register, it is determined that a break field has been detected. Up-counting continues until the next valid edge or counter overflow.

33.2.17 Status Register (SSR)

Address(es): RSCI8.SSR 000A 1448h, RSCI9.SSR 000A 14C8h, RSCI11.SSR 000E 20C8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RDRF	TEND	TDRE	AFER	APER	MFF	—	ORER	—	—	—	—	—	DFER	DPER	DCMF
Value after reset:	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RXDMON	—	—	—	—	—	—	—	—	—	—	ERS	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	This bit is read as 0. The write value should be 0.	R
b4	ERS	Error Signal Status Flag	(Valid only in Smart card interface mode) 0: Error signal Low not responded 1: Error signal Low responded	R
b14 to b5	—	Reserved	These bits are read as 0.	R
b15	RXDMON	RXD Line Monitoring Flag	The state of the RXDn pin is shown. When RINV is 0, 0: RXDn pin is the Low level. 1: RXDn pin is the High level. When RINV is 1, 0: RXDn pin is the High level. 1: RXDn pin is the Low level.	R
b16	DCMF	Data Match Flag	(Valid only in asynchronous mode) 0: No matched 1: Matched	R
b17	DPER	Matched Data Parity Error Flag	(Valid only in asynchronous mode) 0: No parity error occurred at data match detection 1: A parity error has occurred at data match detection	R
b18	DFER	Matched Data Framing Error Flag	(Valid only in asynchronous mode) 0: No framing error occurred at data match detection 1: A framing error has occurred at data match detection	R
b23 to b19	—	Reserved	These bits are read as 0.	R
b24	ORER	Overrun Error Flag	0: No overrun error occurred 1: An overrun error has occurred	R
b25	—	Reserved	This bit is read as 0.	R
b26	MFF	Mode Fault Flag	(Valid only in simple SPI mode.) 0: No mode fault 1: Mode fault	R
b27	APER	Aggregate Parity Error Flag	[Non-FIFO selected (SCR3.FM bit = 0)] 0: No parity error occurred 1: A parity error has occurred [FIFO selected (SCR3.FM bit = 1)] 0: No parity error in all received data in receive FIFO 1: One or more parity errors occurred in received data in receive FIFO	R
b28	AFER	Aggregate Framing Error Flag	[Non-FIFO selected (SCR3.FM bit = 0)] 0: No framing error occurred 1: A framing error has occurred [FIFO selected (SCR3.FM bit = 1)] 0: No framing error in all received data in receive FIFO 1: One or more framing errors occurred in received data in receive FIFO	R

Bit	Symbol	Bit Name	Description	R/W
b29	TDRE	Transmit Data Empty Flag	[Non-FIFO selected (SCR3.FM bit = 0)] 0: Transmit data is in TDR register 1: No transmit data is in TDR register [FIFO selected (SCR3.FM bit = 1)] 0: The quantity of transmit data written in transmit FIFO exceeds the specified transmit triggering number. 1: The quantity of transmit data written in transmit FIFO is equal to or less than the specified transmit triggering number.	R
b30	TEND	Transmit End Flag	0: A character is being transmitted or standing by for transmission. 1: Character transfer has been completed, or sending Break Field.	R
b31	RDRF	Receive Data Full Flag	[Non-FIFO selected (SCR3.FM bit = 0)] 0: No received data is in RDR register 1: Received data is in RDR register [FIFO selected (SCR3.FM bit = 1)] 0: The quantity of receive data written in receive FIFO falls below the specified receive triggering number. 1: The quantity of receive data written in receive FIFO is equal to or greater than the specified receive triggering number.	R

ERS Flag (Error Signal Status Flag)

[Setting condition]

- When an error signal Low is sampled.

[Clearing condition]

- When write 1 to SSCR.ERSC bit.

DCMF Flag (Data Match Flag)

Indicates that RSCI detects the match to the comparison data (SCR4.CMPD[8:0] bits) with receive data.

Clearing the SCR0.RE bit to 0 does not affect the DCMF flag, which retains its previous state.

[Setting condition]

- Matched to the comparison data (SCR4.CMPD[8:0] bits) with receive data, while SCR0.DCME bit = 1.

[Clearing condition]

- When write 1 to SSCR.DCMFC bit.

DPER Flag (Matched Data Parity Error Flag)

It indicates that a parity error occurred at data match detection.

Clearing the SCR0.RE bit to 0 does not affect the DPER flag, which retains its previous state.

[Setting condition]

- When a parity error was detected by the frame in which an data match was detected.

[Clearing condition]

- When write 1 to SSCR.DPERC bit.

Refer to section 14.5.2, Level Detection to exit from the interrupt handling routine after setting the DPER flag to 0.

DFER Flag (Matched Data Framing Error Flag)

It indicates that a framing error occurred at data match detection.
Clearing the SCR0.RE bit to 0 does not affect the DFER flag, which retains its previous state.
[Setting condition]

- When a stop bit of the frame in which an data match was detected is 0.
  When it is a 2-stop mode, the 1st bit of stop bit judges only whether it's 1 and doesn't check the 2nd bit of stop bit.

[Clearing condition]

- When write 1 to SSCR.DFERC bit.

Refer to section 14.5.2, Level Detection to exit from the interrupt handling routine after setting the DFER flag to 0.

ORER Flag (Overrun Error Flag)

Indicates that an overrun error has occurred during reception and the reception ends abnormally.
Clearing the SCR0.RE bit to 0 does not affect the ORER flag, which retains its previous state. In simple I^2C mode, this bit is not use.
[Setting condition with non-FIFO mode (SCR3.FM bit = 0)]

- When the next data is received before reading out RDR register with no error reception data stored in RDR register.

In RDR register, receive data prior to an overrun error occurrence is retained, but data received after the overrun error occurrence is lost. When the ORER flag is set to 1, reception data isn't forwarded to RDR register. Note that, in clock synchronous mode and simple SPI mode, serial reception will be stop.
[Setting condition with FIFO mode (SCR3.FM bit = 1)]

- When the next serial reception is completed while the receive FIFO is full of 32 receive data.

[Clearing condition]

- When write 1 to SSCR.ORERC bit.

Refer to section 14.5.2, Level Detection to exit from the interrupt handling routine after setting the ORER flag to 0.

MFF Flag (Mode Fault Flag)

This bit indicates mode faults. In a multi-master configuration, determine the mode fault occurrence by reading the MFF flag.
[Setting condition]

- Input on the SSn# pin being at the low level during master operation (SCR3.CKE[1:0] bits = 00b or 01b) in simple SPI mode.

[Clearing condition]

- When write 1 to SSCR.MFFC bit.

APER Flag (Aggregate Parity Error Flag)

Indicates that a parity error has occurred during reception and the reception ends abnormally.
Clearing the SCR0.RE bit to 0 does not affect the APER flag, which retains its previous state.
In clock synchronous mode, simple SPI mode and simple I^2C mode, this bit not used.
[Setting condition]

- In non-FIFO mode, when a parity error is detected during reception.
- In FIFO mode, when one or more parity error is detected for data in receive FIFO.

In non-FIFO mode, the received data when the parity error occurs is transferred to RDR register, but no RXI interrupt request occurs. Note that when the APER flag is being set to 1, the subsequent receive data is not transferred to RDR register.
[Clearing condition]

- When write 1 to SSCR.APERC bit.

Refer to section 14.5.2, Level Detection to exit from the interrupt handling routine after setting the APER flag to 0.

AFER Flag (Aggregate Framing Error Flag)

Indicates that a framing error has occurred during reception and the reception ends abnormally. Clearing the SCR0.RE bit to 0 does not affect the AFER flag, which retains its previous state.
In clock synchronous mode, simple SPI mode and simple I^2C mode, this bit not used.
[Setting condition]

- In non-FIFO mode, when 0 is sampled as the stop bit during reception.
- In FIFO mode, when one or more framing error is detected for data in receive FIFO.
- In manchester mode, when both sampling results (1/4 and 3/4 sampling points) are not 1 for 1 stop bit.

In 2 stop bit mode, only the first stop bit is checked whether it is 1 but the second stop bit is not checked.
In non-FIFO mode, the received data when the framing error occurs is transferred to RDR register, but no RXI interrupt request occurs. In addition, when the AFER flag is being set to 1, the subsequent receive data is not transferred to RDR register.
In extended serial mode, even if a condition that changes to 1 occurs when XCR1.SDST bit = 1, the AFER flag set timing is delayed up to the Break Field judgment timing at the longest, since it may be a Break Field. If an edge is detected on the RXD signal before the Break Field judgment timing, AFER is detected. If no edge is detected in the RXD signal before the Break Field judgment timing, Break Field is detected.
[Clearing condition]

- When write 1 to SSCR.AFERC bit.

Refer to section 14.5.2, Level Detection to exit from the interrupt handling routine after setting the AFER flag to 0.

TDRE Flag (Transmit Data Empty Flag)

(1) Non-FIFO selected (SCR3.FM bit = 0)
Indicates the presence of transmit data in the TDR register.
The condition of SCR0.TE bit = 0 has priority over the condition of 0.
If other conditions that become 1 and conditions that become 0 are satisfied at the same time, the TDRE flag is set to 0.
[Setting condition]

- When SCR0.TE bit is 0.
- When data is transmitted from the TDR register to the TSR register.

[Clearing condition]

- When write 1 to SSCR.TDREC bit.
- When the transmission data is written to the TDR register during SCR0.TE bit = 1.

(2) FIFO selected (SCR3.FM bit = 1)
Indicates that data has been transferred from the transmit FIFO (TDR register) into the transmit shift register (TSR), the quantity of data in transmit FIFO has fallen below the specified transmit triggering number.
When the condition which becomes 1 and the condition which becomes 0 were formed at the same time, TDRE flag will be 0. After that, when the number of data stored in transmit FIFO is judged, and if that is equal to or less than TTRG value, TDRE flag is set to 1 after 1 PCLK cycle.
[Setting condition]

- When the quantity of transmit data written in transmit FIFO is equal to or less than the specified transmit triggering number*1.

[Clearing condition]

- When write 1 to SSCR.TDREC bit.
- When the transmission data is written to transmit FIFO by the DTC or DMAC (the last block transfer when block transfer).

Note 1. Because the transmit FIFO is a 32-stage FIFO register, the maximum number of data that can be written to the TDR register when the TDRE flag is 1 is “32 – TFSR.T[5:0]”. All other data written to the TDR register above that value is ignored.

TEND Flag (Transmit End Flag)

(1) Non-FIFO selected (SCR3.FM bit = 0), and not smart card interface mode (SCR3.MOD[2:0] bits ≠ 001b)

Indicates completion of transmission.

[Setting condition]

- When SCR0.TE bit is 0.
- When the SCR0.TE bit is changed from 0 to 1, the TEND flag is not affected and retains the value 1.
- When the TDR register is not updated at the time of transmission of the tail-end bit of a character being transmitted.
- When the TDR register is not updated at the end of DE signal hold time with DE control function enable (SCR3.DEEN bit = 1).
- When Break Field is sending.

[Clearing condition]

- When the transmission data was written to the TDR register during SCR0.TE bit = 1.
- When write 1 to SSCR.TDREC bit during SCR0.TE bit = 1.

Refer to section 14.5.2, Level Detection to exit from the interrupt handling routine after setting the TEND flag to 0.

(2) Non-FIFO selected (SCR3.FM bit = 0), and smart card interface mode (SCR3.MOD[2:0] bits = 001b)

With no error signal from the receiving side, this bit is set to 1 when further data for transfer is ready to be transferred to the TDR register.

[Setting condition]

- When SCR0.TE bit is 0.
- When the SCR0.TE bit is changed from 0 to 1, the TEND flag is not affected and retains the value 1.
- When a specified period has elapsed after the latest transmission of 1 byte, the ERS flag is 0, and the TDR register is not updated.
  The set timing is determined by register settings as listed below.
  When GM bit = 0 and BLK bit = 0, 12.5 etu after the start of transmission
  When GM bit = 0 and BLK bit = 1, 11.5 etu after the start of transmission
  When GM bit = 1 and BLK bit = 0, 11.0 etu after the start of transmission
  When GM bit = 1 and BLK bit = 1, 11.0 etu after the start of transmission

[Clearing condition]

- When the transmission data was written to the TDR register during SCR0.TE bit = 1.
- When write 1 to SSCR.TDREC bit during SCR0.TE bit = 1.

(3) FIFO selected (SCR3.FM bit = 1)

Indicates that the transmit FIFO does not contain valid data when transmitting the last bit of a serial character, so the transmission is halted.

[Setting condition]

- TEND is set to 1 when transmit FIFO does not contain transmit data when the last bit of a one-byte serial character is transmitted.
- When the TDR register is not updated at the end of DE signal hold time with DE control function enable (SCR3.DEEN bit = 1).

[Clearing condition]

- When the transmission data was written to the TDR.TDAT[7:0] bits during SCR0.TE bit = 1.

RDRF Flag (Receive Data Full Flag)

(1) Non-FIFO selected (SCR3.FM bit = 0)

Indicates the presence of receive data in the RDR register.

[Setting condition]

- When the reception ends normally, and receive data is forwarded from the RSR register to the RDR register.

[Clearing condition]

- When write 1 to SSCR.RDRFC bit.
- When the read data is read from the RDR register.

(2) FIFO selected (SCR3.FM bit = 1)

Indicates that receive data has been transferred to the receive FIFO (RDR register), and the quantity of data in receive FIFO is equal to or greater than the specified receive triggering number. When FCR.RTRG[4:0] bits are set to 0, RDRF flag is set if the quantity of data in receive FIFO is greater than or equal to 1.

[Setting condition]

RDRF is set to 1 when the quantity of receive data in receive FIFO is equal to or greater than the specified receive triggering number*2.

[Clearing condition]

- When write 1 to SSCR.RDRFC bit.
- When the reception data is read form receive FIFO by the DTC or DMAC (the last block transfer when block transfer).

When the condition which becomes 1 and the condition which becomes 0 were formed at the same time, RDRF flag will be 0. After that, when the number of stored data in receive FIFO is judged, and if that is equal to or greater than RTRG value, RDRF flag is set to 1 after 1 PCLK cycle.

Note 2. Since the receive FIFO is a 32-stage FIFO register, the maximum number of data that can be read when RDF is 1 is indicated by the RFSR.R[5:0] bits. After reading all the data in the RDR register, continuing a read access results in an undefined value.

Note: In non-FIFO mode, do not clear the RDRF and TDRE flags by the SSCR register except when the communication is to be aborted.

33.2.18 I2C Status Register (SISR)

Address(es): RSCI8.SISR 000A 144Ch, RSCI9.SISR 000A 14CCh, RSCI11.SISR 000E 20CCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	IICSTIF	—	—	IICACKR
Value after reset:	0	0	0	0	0	0	0	0	0	0	x	x	0	x	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IICACKR	ACK Reception Data Flag	0: ACK received 1: NACK received	R
b1	—	Reserved	This bit is read as 0.	R
b2	—	Reserved	The read value is undefined.	R
b3	IICSTIF	Condition Generation Completed Flag	0: There are no requests for generating conditions or a condition is being generated. 1: A start, restart, or stop condition is completely generated.	R
b5, b4	—	Reserved	The read value is undefined.	R
b31 to b6	—	Reserved	These bits are read as 0.	R

IICACKR Flag (ACK Reception Data Flag)

Received ACK and NACK bits can be read from this flag.
The IICACKR flag is updated at the rising of SSCLn signal for the ACK/NACK receiving bit.

IICSTIF Flag (Condition Generation Completed Flag)

After generating a condition, this flag indicates that the generation is completed. When using the IICSTAREQ, IICRSTAREQ, or IICSTPREQ bit to cause generation of a condition, do so after setting the IICSTIF flag to 0.
When the IICSTIF flag is 1 while an interrupt request is enabled by setting the SCR0.TEIE bit, an STI request is output.
[Setting condition]

- Completion of the generation of a start, restart, or stop condition (however, in cases where this conflicts with any of the conditions for the flag becoming 0 listed below, the clearing condition takes precedence)

[Clearing condition]

- Writing 1 to SISCR.IICSTIFC bit
- When operation is not in simple I2C
- Writing 0 to SCR0.TE bit

33.2.19 Receive FIFO Status Register (RFSR)

Address(es): RSCI11.RFSR 000E 20D0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	FEC[5:0]						—	—	PEC[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	R[5:0]						—	—	—	—	—	—	—	DR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DR	Receive Data Ready Flag	0: Receiving is in progress, or no received data has remained in receive FIFO after normally completed receiving. (receive FIFO is empty) 1: The following receive data does not come for a fixed period after storing data below the threshold in the receive FIFO	R
b1	—	Reserved	This bit is undefined when read.	R
b7 to b2	—	Reserved	These bits are read as 0.	R
b13 to b8	R[5:0]	Receive FIFO Data Count	(Valid in asynchronous mode (including multi-processor), clock synchronous mode, simple SPI mode, when SCR3.FM bit is 1.) Indicate the quantity of receive data stored in receive FIFO	R
b15, b14	—	Reserved	These bits are read as 0.	R
b21 to b16	PEC[5:0]	Parity Error Count	(Valid only in asynchronous mode) Indicates the quantity of data with a parity error among the receive data stored in the receive FIFO data register.	R
b23, b22	—	Reserved	These bits are read as 0.	R
b29 to b24	FEC[5:0]	Framing Error Count	(Valid only in asynchronous mode) Indicates the quantity of data with a framing error among the receive data stored in the receive FIFO data register.	R
b31, b30	—	Reserved	This bit is read as 0.	R

DR Flag (Receive Data Ready Flag)

Indicates that the quantity of data stored in the receive FIFO (RDR register) falls below the specified receive triggering number, and that no next data has been received yet after the elapse of 15 etu from the last stop bit in asynchronous mode. This flag becomes 1 only when the FIFO buffer is enabled in synchronous mode (including multiprocessor mode), and does not become 1 in other modes.

[Setting conditions]

DR is set to 1 when the following conditions are met.

- After receive FIFO (RDR register) receives less data than the specified receive triggering number, no next data has been received yet after the elapse of 15 etu*1 from the last stop bit
- SSR.AFER and APER flags are 0.

[Clearing conditions]

- When all receive data in the receive FIFO (RDR register) is read and 1 is written to RFSCR.DRC bit.
- When SCR3.FM bit is 0.

Refer to section 14.5.2, Level Detection to exit from the interrupt handling routine after setting the DR flag to 0 when the DR flag is set to the source of the receive error interrupt by setting the FCR.DRES bit to 1.

Note 1. This is equivalent to one and a half (1.5) frames in the 8-bit format with one stop bit (etu: elementary time unit).

R[5:0] Bits (Receive FIFO Data Count)

Indicate the quantity of receive data stored in receive FIFO.
00h means no receive data. 20h means receive FIFO is full.

PEC[5:0] Bits (Parity Error Count)

The value indicates the quantity of data stored in the receive FIFO registers with a parity error.

FEC[5:0] Bits (Framing Error Count)

The value indicates the quantity of data stored in the receive FIFO registers with a framing error.

33.2.20 Transmit FIFO Status Register (TFSR)

Address(es): RSCI11.TFSR 000E 20D4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	T[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	T[5:0]	Transmit FIFO Data Count	(Valid in asynchronous mode (including multi-processor), clock synchronous mode, simple SPI mode, when SCR3.FM bit is 1.) Indicate the quantity of non-transmit data stored in transmit FIFO	R
b31 to b6	—	Reserved	These bits are read as 0.	R

T[5:0] Bits (Transmit FIFO Data Count)

Indicate the quantity of non-transmitted data stored in transmit FIFO.

00h means no untransmitted data. 20h means transmit FIFO is full.

33.2.21 Manchester Mode Status Register (MMSR)

Address(es): RSCI9.MMSR 000A 14D8h, RSCI11.MMSR 000E 20D8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	RSYNC	—	MCER	—	SBER	SYER	PFER
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PFER	Preface Error Flag	This bit is set when a preface error (pattern mismatch) is detected 0: No preface error detected 1: Preface error detected	R
b1	SYER	Sync Error Flag	This bit is set when no edge is detected in the adjustable range during receive retiming 0: No receive Sync error detected 1: Receive Sync error detected	R
b2	SBER	Start Bit Error Flag	This bit is set when a pattern mismatch in the start bit area is detected 0: No start bit error detected 1: Start bit error detected	R
b3	—	Reserved	This bit is read as 0.	R
b4	MCER	Manchester Code Error Flag	Valid for manchester mode only 0: No Manchester code error occurred 1: Manchester code error has occurred	R
b5	—	Reserved	This bit is read as 0.	R
b6	RSYNC	Received Sync Data	It is valid when MMCR.SBLEN bit = 1 in manchester mode, 0 is read otherwise. 0: The received the Start Bit is DATA Sync 1: The received the Start Bit is COMMAND Sync	R
b31 to b7	—	Reserved	These bits are read as 0.	R

PFER Flag (Preface Error Flag)

This bit indicates that a preface error was detected when receiving frames in manchester mode.

Even when the SCR0.RE bit is set to 0 (serial reception is disabled), the PFER flag is not affected and retains its previous value.

[Setting condition]

When detecting a preface error when receiving frames in manchester mode

The following operations are performed when a preface error occurs.

<When MMCR.PFERIE = 1>

The received data is not transferred to the RDR register and no RXI interrupt request occurs. Instead, an ERI interrupt request occurs. Note that when the PFER flag is being set to 1, the subsequently received data is not transferred to the RDR register.

<When MMCR.PFERIE = 0>

The received data is transferred to the RDR register and an RXI interrupt request occurs. An ERI interrupt request is not generated. The subsequent receive operations are not affected even with the PFER flag being set to 1.

[Clearing condition]
Write 1 to MMSCR.PFERC bit.
Refer to section 14.5.2, Level Detection to exit from the interrupt handling routine after setting the PFER flag to 0.

SYER Flag (Sync Error Flag)

This bit indicates that a receive Sync error was detected when receiving frames in manchester mode with MMCR.SADJE bit = 1 (manchester edge retiming enabled).
Even when the SCR0.RE bit is set to 0 (serial reception is disabled), the SYER flag is not affected and retains its previous value.
[Setting condition]
When detecting a receive Sync error when receiving frames in manchester mode
The following operations are performed when a receive Sync error occurs.
<When MMCR.SYERIE = 1>
Although the received data is transferred to the RDR register, no RXI interrupt request occurs. Instead, an ERI interrupt request occurs. Note that when the SYER flag is being set to 1, the subsequently received data is not transferred to the RDR register.
<When MMCR.SYERIE = 0>
The received data is transferred to the RDR register and an RXI interrupt request occurs. An ERI interrupt request is not generated. The subsequent receive operations are not affected even with the SYER flag being set to 1.
[Clearing condition]
Write 1 to MMSCR.SYERC bit.
Refer to section 14.5.2, Level Detection to exit from the interrupt handling routine after setting the SYER flag to 0.

SBER Flag (Start Bit Error Flag)

This bit indicates that a start bit error was detected when receiving frames in manchester mode.
Even when the SCR0.RE bit is set to 0 (serial reception is disabled), the SBER flag is not affected and retains its previous value.
[Setting condition]
When detecting a start bit error when receiving frames in manchester mode
The following operations are performed when a start bit error occurs.
<When MMCR.SBERIE bit = 1>
The received data is not transferred to the RDR register and no RXI interrupt request occurs. Instead, an ERI interrupt request occurs. Note that when the SBER flag is being set to 1, the subsequently received data is not transferred to the RDR register.
<When MMCR.SBERIE bit = 0>
The received data is transferred to the RDR register and an RXI interrupt request occurs. An ERI interrupt request is not generated. The subsequent receive operations are not affected even with the SBER flag being set to 1.
[Clearing condition]
Write 1 to MMSCR.SBERC bit.
Refer to section 14.5.2, Level Detection to exit from the interrupt handling routine after setting the SBER flag to 0.

MCER Flag (Manchester Code Error Flag)

When data is received in manchester mode, Manchester code error is detected, and it is displayed. Even when the SCR0.RE bit is set to 0 (serial reception is disabled), the MCER flag is not affected and retains its previous value.
[Setting conditions]
When receiving in manchester mode and detecting manchester code error in data area of received frame.
Received data when an error occurs is transferred to the RDR register, but the RXI interrupt request is not generated.
When the manchester code error flag is set to 1, subsequent receive data is not transferred to the RDR register.
For details on manchester code error, see section 33.5.11, Errors in Manchester Mode.
[Clearing condition]
Write 1 to MMSCR.MCERC bit.
Refer to section 14.5.2, Level Detection to exit from the interrupt handling routine after setting the MCER flag to 0.

RSYNC Flag (Received Sync Data)

When manchester mode (SCR3.MOD[2:0] = 101b) and MMCR.SBLEN = 1, this bit indicates the type of Sync of the received the start bit. For other settings, it is fixed to 0.

33.2.22 Extended Serial Mode Status Register 0 (XSR0)

Address(es): RSCI9.XSR0 000A 14DCh, RSCI11.XSR0 000E 20DCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CF1RD[7:0]								CF0RD[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	AEDF	COF	PIBDF	CF1MF	CF0MF	BFDF	BCDF	BFOF	—	—	—	—	—	—	RXDSF	SFSF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SFSF	Start Frame Status Flag	0: Start Frame detection disabled or Start Frame detection complete 1: Before Start Frame detection or during detection	R*1
b1	RXDSF	RXD Input Status Flag	0: RXD input to RSCI core is enabled 1: RXD input to RSCI core is disabled (RXD is not input to the RSCI core)	R*1
b7 to b2	—	Reserved	These bits are read as 0.	R
b8	BFOF	Break Field Low Width Output Complete Flag	0: When Break Field transmission is not completed 1: When Break Field transmission is completed	R
b9	BCDF	Bus Collision Detected Flag	0: When bus conflict is not detected 1: When bus conflict is detected	R
b10	BFDF	Break Field Low Width Detection Flag	0: When Break Field is not detected 1: When Break Field is detected	R
b11	CF0MF	Control Field 0 Match Flag	0: When Control Field 0 data and the compare data do not match 1: When Control Field 0 data and the compare data match	R
b12	CF1MF	Control Field 1 Match Flag	0: When Control Field 1 data and the compare data do not match 1: When Control Field 1 data and the compare data match	R
b13	PIBDF	Priority Interrupt Bit Detection Flag	0: When priority interrupt bit is not detected 1: When Priority interrupt bit is detected	R
b14	COF	Count Overflow Flag	0: When the counter for Break Field detection does not overflow 1: When the counter for Break Field detection overflows	R
b15	AEDF	Effective Edge Detection Flag	0: When Active edge is not detected 1: When Active edge is detected	R
b23 to b16	CF0RD[7:0]	Control Field 0 Received Data	Control Field 0 received data.	R
b31 to b24	CF1RD[7:0]	Control Field 1 Received Data	Control Field 1 received data.	R

Note 1. Wait at least 1 PCLK cycle after the receive data full interrupt (RXI) before reading this register.

SFSF Flag (Start Frame Status Flag)

Indicates whether detect Start Frame is being detected.
[Setting condition]

- When 1 is written to XCR1.SDST bit.
- When a Break Field is detected in the Control Field 0/Control Field 1/Information Field phase and the transition to the Control Field 0 or Control Field 1 reception state occurs.

[Clearing condition]

- When XCR1.SDST bit is 0.
- When Start Frame detection is completed.

RXDSF Flag (RXD Input Status Flag)

Indicates the RXD input status to the RSCI core. When this bit is 1, RXD input is received only by the extended serial module and the Break Field is detected and is not input to the RSCI core.

33.2.23 Extended Serial Mode Status Register 1 (XSR1)

Address(es): RSCI9.XSR1 000A 14E0h, RSCI11.XSR1 000E 20E0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CCV[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	CCV[15:0]	Captured Count Value	Stores the 16-bit counter capture value.	R
b31 to b16	—	Reserved	These bits are read as 0.	R

CCV[15:0] Bits (Captured Count Value)

Stores the capture value of the 16-bit counter of the extended serial module.

When sending Start Frame

This register holds the previous value.

When receiving Start Frame with bit rate measurement disabled

If a Break Field is detected in the Break Field detection state (refer to Figure 33.73), the Break Field length is captured and held (counter value is captured at the rising edge of RXD).

If a Break Field is detected in a state other than the Break Field detection state, the previous value is retained.

If the counter overflows, it will not be captured.

When receiving Start Frame with bit rate measurement enabled

The count value is captured and held at the valid edge (both RXD edges). However, in the Break Field detection state, the count value is not captured even if a valid edge occurs. Counter capture value retention is canceled by this register read. Even if a valid edge occurs before reading, the counter value is not captured.

33.2.24 Status Clear Register (SSCR)

Address(es): RSCI8.SSCR 000A 1468h, RSCI9.SSCR 000A 14E8h, RSCI11.SSCR 000E 20E8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RDRFC	—	TDREC	AFERC	APERC	MFFC	—	ORERC	—	—	—	—	—	DFERC	DPERC	DCMFC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	ERSC	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b4	ERSC	ERS Clear	Setting this bit to 1 clears the SSR.ERS flag. The read value is always 0.	W
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	DCMFC	DCMF Clear	Setting this bit to 1 clears the SSR.DCMF flag. The read value is always 0.	W
b17	DPERC	DPER Clear	Setting this bit to 1 clears the SSR.DPER flag. The read value is always 0.	W
b18	DFERC	DFER Clear	Setting this bit to 1 clears the SSR.DFER flag. The read value is always 0.	W
b23 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R
b24	ORERC	ORER Clear	Setting this bit to 1 clears the SSR.ORER flag. The read value is always 0.	W
b25	—	Reserved	This bit is read as 0. The write value should be 0.	R
b26	MFFC	MFF Clear	Setting this bit to 1 clears the SSR.MFF flag. The read value is always 0.	W
b27	APERC	APER Clear	Setting this bit to 1 clears the SSR.APER flag. The read value is always 0.	W
b28	AFERC	AFER Clear	Setting this bit to 1 clears the SSR.AFER flag. The read value is always 0.	W
b29	TDREC	TDRE Clear	Setting this bit to 1 clears the SSR.TDRE flag. The read value is always 0.	W
b30	—	Reserved	This bit is read as 0. The write value should be 0.	R
b31	RDRFC	RDRF Clear	Setting this bit to 1 clears the SSR.RDRF flag. The read value is always 0.	W

33.2.25 I2C Status Clear Register (SISCR)

Address(es): RSCI8.SISCR 000A 146Ch, RSCI9.SISCR 000A 14ECh, RSCI11.SISCR 000E 20ECh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	IICSTIFC	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b2	—	Reserved	This bit is read as 0.	R
b3	IICSTIFC	IICSTIF Clear	Setting this bit to 1 clears the SISR.IICSTIF flag. The read value is always 0.	W
b31 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R

33.2.26 Receive FIFO Status Clear Register (RFSCR)

Address(es): RSCI11.RFSCR 000E 20F0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DRC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DRC	DR Clear	Setting this bit to 1 clears the RFSR.DR flag. The read value is always 0.	W
b1	—	Reserved	This bit is read as 0.	R
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R

33.2.27 Manchester Mode Status Clear Register (MMSCR)

Address(es): RSCI9.MMSCR 000A 14F4h, RSCI11.MMSCR 000E 20F4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	MCERC	—	SBERC	SYERC	PFERC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PFERC	PFER Clear	Setting this bit to 1 clears the MMSR.PFER flag. The read value is always 0.	W
b1	SYERC	SYER Clear	Setting this bit to 1 clears the MMSR.SYER flag. The read value is always 0.	W
b2	SBERC	SBER Clear	Setting this bit to 1 clears the MMSR.SBER flag. The read value is always 0.	W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R
b4	MCERC	MCER Clear	Setting this bit to 1 clears the MMSR.MCER flag. The read value is always 0.	W
b31 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R

33.2.28 Extended Serial Mode Status Clear Register (XSCR)

Address(es): RSCI9.XSCR 000A 14F8h, RSCI11.XSCR 000E 20F8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	AEDCL	COFC	PIBDCL	CF1MCL	CF0MCL	BFDCL	BCDCL	BFOC	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	BFOC	BFOF Clear	Setting this bit to 1 clears the XSR0.BFOF flag. The read value is always 0.	W
b9	BCDCL	BCDF Clear	Setting this bit to 1 clears the XSR0.BCDF flag. The read value is always 0.	W
b10	BFDCL	BFDF Clear	Setting this bit to 1 clears the XSR0.BFDF flag. The read value is always 0.	W
b11	CF0MCL	CF0MF Clear	Setting this bit to 1 clears the XSR0.CF0MF flag. The read value is always 0.	W
b12	CF1MCL	CF1MF Clear	Setting this bit to 1 clears the XSR0.CF1MF flag. The read value is always 0.	W
b13	PIBDCL	PIBDF Clear	Setting this bit to 1 clears the XSR0.PIBDF flag. The read value is always 0.	W
b14	COFC	COF Clear	Setting this bit to 1 clears the XSR0.COF flag. The read value is always 0.	W
b15	AEDCL	AEDF Clear	Setting this bit to 1 clears the XSR0.AEDF flag. The read value is always 0.	W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R

33.2.29 HBS Support Mode Control Register (HBSCR)

Address(es): RSCI8.HBSCR 000A 141Eh, RSCI9.HBSCR 000A 149Eh, RSCI11.HBSCR 000E 209Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	LPS	AOE	—	HBSE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	HBSE	HBS Support Mode Enable	0: Pulse width for data 0 is specified as 100% of a bit period (NRZ coding). 1: Pulse width for data 0 is specified as 50% of a bit period (RZI coding and inversion).	R/W *1
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R
b2	AOE	Alternate Output Enable	0: Data is output from the TXDn pin. 1: Data 0 is alternately output from the TXDAn and TXDBn pins.	R/W *1
b3	LPS	Leading Output Pin Select	0: When HBSE bit = 1 and AOE bit = 1, transmission starts from TXDAn pin 1: When HBSE bit = 1 and AOE bit = 1, transmission starts from TXDBn pin	R/W *1
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Writable only when SCR0.TE bit = 0 and SCR0.RE bit = 0.

HBSE Bit (HBS Support Mode Enable)

When this bit is 1, the transmitter encodes the transmit data to the negative logic RZI code and the receiver decodes the receive data to the NRZ code. Also, transmit data can be output from the TXDAn/TXDBn pin. This function should be used only in asynchronous mode.

AOE Bit (Alternate Output Enable)

This bit is used to select whether the data is output from the TXDn pin or data 0 is alternately output from the TXDAn and TXDBn pins in HBS support mode.

LPS Bit (Leading Output Pin Select)

This bit is used when the AOE bit is 1 in HBS support mode.

When it is set to 0, the start bit is transmitted from TXDAn pin and the data 0 is output alternately from the TXDBn and TXDAn pins.

When it is set to 1, the start bit is transmitted from TXDBn pin and the data 0 is output alternately from the TXDAn and TXDBn pins.

For details, refer to the operation description in section 33.6, HBS Support Mode.

33.2.30 Device Function Select Register 0 (PRDFR0)

Address(es): SYSTEM.PRDFR0 0008 00D0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	SCI11RXD[1:0]		—	—	SCI9RXD[1:0]		SCI8RXD[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	SCI5RXD[1:0]		—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11, b10	SCI5RXD[1:0]	SCI5 RXD Input Signal Select	These bits select the signal to be input to the RXD pin of SCI5. b11 b10 0 0: Input signal from the RXD5 pin 0 1: COMP3 level detection signal 1 0: COMP4 level detection signal Settings other than above are prohibited.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b17, b16	SCI8RXD[1:0]	RSCI8 RXD Input Signal Select	These bits select the signal to be input to the RXD pin of RSCI8. b17 b16 0 0: Input signal from the RXD008 pin 0 1: COMP3 level detection signal 1 0: COMP4 level detection signal Settings other than above are prohibited.	R/W
b19, b18	SCI9RXD[1:0]	RSCI9 RXD Input Signal Select	These bits select the signal to be input to the RXD pin of RSCI9. b19 b18 0 0: Input signal from the RXD009 pin 0 1: COMP3 level detection signal 1 0: COMP4 level detection signal Settings other than above are prohibited.	R/W
b21, b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23, b22	SCI11RXD[1:0]	RSCI11 RXD Input Signal Select	These bits select the signal to be input to the RXD pin of RSCI11. b23 b22 0 0: Input signal from the RXD011 pin 0 1: COMP3 level detection signal 1 0: COMP4 level detection signal Settings other than above are prohibited.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

If the RXD signal is attenuated by the transmission line effects and does not meet the VIH/VIL specifications, it can be improved by intervening a comparator.
When using a comparator, connect the RXD signal to one of the CMPC30 to CMPC33 or CMPC40 to CMPC43 pins. Also, disable the noise filter in the comparator. In this case, the RXD pin of the channel is not used.

33.3 Operation in Asynchronous Mode

Figure 33.3 shows the general format for asynchronous serial communications.
One frame consists of a start bit (low level), transmit/receive data, a parity bit, and stop bits (high level).
In asynchronous serial communications, the communications line is held in the mark state (high level) when not communicating.
The RSCI monitors the communications line. When the RSCI detects a start bit, it starts serial communication. The detection condition of the start bit changes according to the SCR3.RXDESEL bit setting. RSCI regards space (low level) as a start bit when SCR3.RXDESEL bit is 0. RSCI regards a fall edge as a start bit when RXDESEL bit is 1.
Inside the RSCI, the transmitter and receiver are independent units, enabling full-duplex communications. Both the transmitter and the receiver also have a double-buffered structure (it has also FIFO mode), so that data can be read or written during transmission or reception, enabling continuous data transmission and reception.

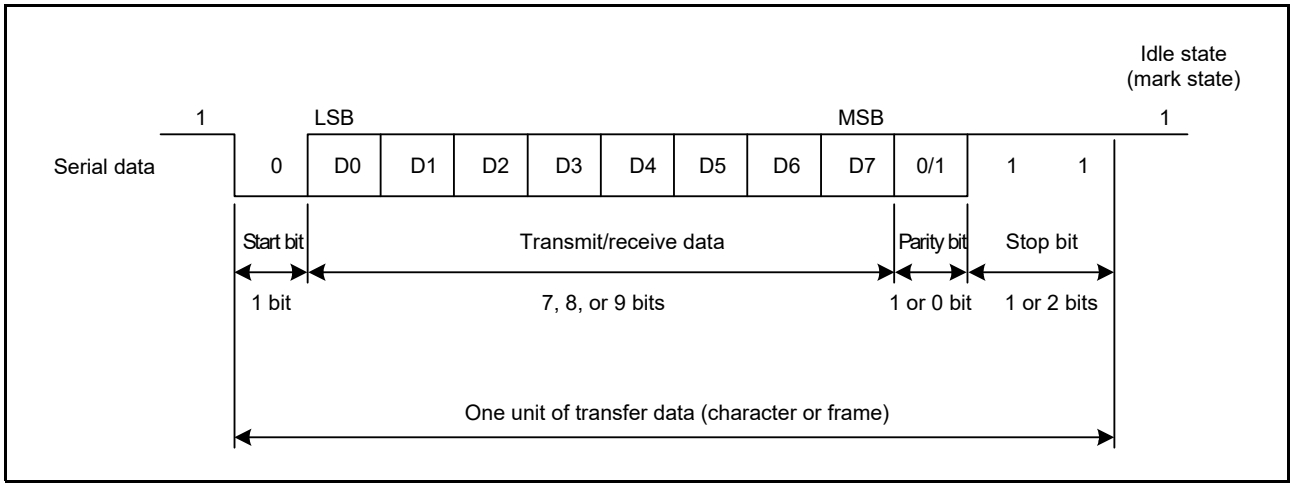


Figure 33.3 Data Format in Asynchronous Serial Communications (Example with 8-Bit Data, Parity, 2 Stop Bits)

33.3.1 Serial Data Transfer Format

Table 33.28 lists the serial data transfer formats that can be used in asynchronous mode.
Any of 18 transfer formats can be selected according to the SCR1 and SCR3 setting. For details of multi-processor function, refer to section 33.4, Multi-Processor Communication Function.

Table 33.28 Serial Transfer Formats (Asynchronous Mode)

SCR3		SCR1	SCR3		Serial Transfer Format and Frame Length
CHR[1]	CHR[0]	PE	MP	STOP	1 2 3 4 5 6 7 8 9 10 11 12 13
0	0	0	0	0	S 9-bit data STOP
0	0	0	0	1	S 9-bit data STOP STOP
0	0	1	0	0	S 9-bit data P STOP
0	0	1	0	1	S 9-bit data P STOP STOP
1	0	0	0	0	S 8-bit data STOP
1	0	0	0	1	S 8-bit data STOP STOP
1	0	1	0	0	S 8-bit data P STOP
1	0	1	0	1	S 8-bit data P STOP STOP
1	1	0	0	0	S 7-bit data STOP
1	1	0	0	1	S 7-bit data STOP STOP
1	1	1	0	0	S 7-bit data P STOP
1	1	1	0	1	S 7-bit data P STOP STOP
0	0	—	1	0	S 9-bit data MPB STOP
0	0	—	1	1	S 9-bit data MPB STOP STOP
1	0	—	1	0	S 8-bit data MPB STOP
1	0	—	1	1	S 8-bit data MPB STOP STOP
1	1	—	1	0	S 7-bit data MPB STOP
1	1	—	1	1	S 7-bit data MPB STOP STOP

S: Start bit
STOP: Stop bit
P: Parity bit
MPB: Multi-processor bit

33.3.2 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode

In asynchronous mode, the RSCI operates on a base clock with a frequency of 16 times*1 the bit rate.
In reception, the RSCI samples the falling edge of the start bit using the base clock, and performs internal synchronization*2. When sampling timing does not adjust (SCR4.RTADJ bit = 0 or SCR4.RTADJ bit = 1 and SCR4.RTMG[2:0] bits = 000b), receive data is sampled at the rising edge of the 8th pulse*1 of the base clock, data is latched at the middle of each bit, as shown in Figure 33.4. Thus the reception margin in asynchronous mode is determined by formula (1) below.

$$M = \left| \left(0.5 - \frac{1}{2N}\right) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\ (\%) \quad \cdots \text{Formula (1)}$$

M: Reception margin
N: Ratio of bit rate to clock
(N = 16 when SCR2.ABCSE = 0 and SCR2.ABCS = 0, N = 8 when SCR2.ABCSE = 0 and SCR2.ABCS = 1, N = 6 when SCR2.ABCSE = 1)
D: Duty cycle of clock (D = 0.5 to 1.0)
L: Frame length (L = 9 to 13)
F: Absolute value of clock frequency deviation

Assuming values of F = 0 and D = 0.5 in formula (1), the reception margin is determined by the formula below.

$M = \{0.5 - 1/(2 \times 16)\} \times 100\ (\%) = 46.875\ (\%)$

However, this is only the computed value, and a margin of 20% to 30% should be allowed in system design.

Note 1. This is an example when the ABCSE and ABCS bits in the SCR2 register is 0. When the ABCSE bit is 0 and the ABCS bit is 1, a frequency of 8 times the bit rate is used as a base clock and receive data is sampled at the rising edge of the 4th pulse of the base clock. When the ABCSE bit is 1, a frequency of 6 times the bit rate is used as a base clock and receive data is sampled at the rising edge of the 3rd pulse of the base clock.
Note 2. The determination condition of the start bit is as follows.
In the case of the function of adjust sampling timing is OFF (RTADJ bit = 0):
The determination condition of a start bit is that Low beyond half bit length continues. It is same as the sampling timing.
In Figure 33.4, the low period should be kept over 8-cycles to detect a start bit. If Low period does not keep over 8-cycles, the RSCI judges this as a noise. So, the RSCI does not start reception and wait start bit.
In the case of the function of adjust sampling timing is ON (RTADJ bit = 1):
The determination condition of a start bit is that Low keeps up until the sampling timing.
Adjusting the sampling timing forward (RTMG[3] bit = 1) increases the possibility of erroneously determining a noise as the start bit.

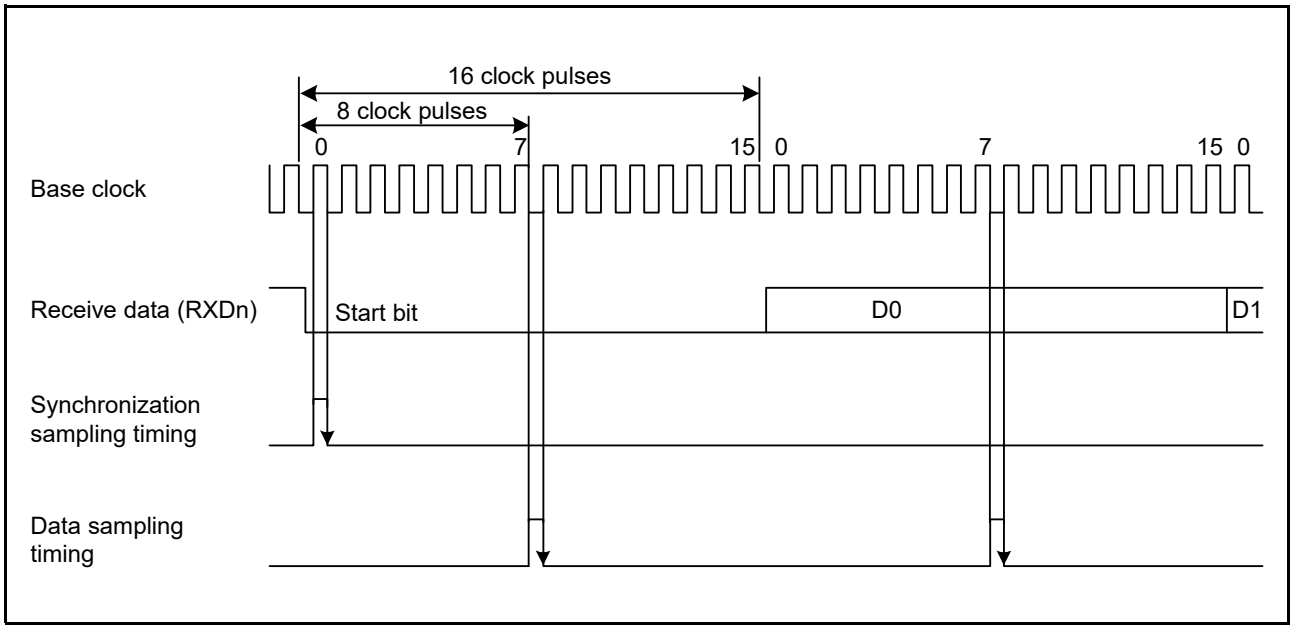


Figure 33.4 Receive Data Sampling Timing in Asynchronous Mode

33.3.3 Clock

Either an internal clock generated by the on-chip baud rate generator or an external clock input to the SCKn pin can be selected as the RSCI's transfer clock, according to the setting of the SCR3.CKE[1:0] bits.
When an external clock is input to the SCKn pin, the clock frequency should be 16 times the bit rate (when SCR2.ABCS bit = 0) and 8 times the bit rate (when SCR2.ABCS bit = 1).
When the RSCI is operated on an internal clock, the clock can be output from the SCKn pin. The frequency of the clock output in this case is equal to the bit rate, and the phase is such that the rising edge of the clock is in the middle of the transmit data, as shown in Figure 33.5.
If you selected an internal clock, the SCKn pin is outputted after SCR0.TE bit is set to 1 or SCR0.RE bit is set to 1.

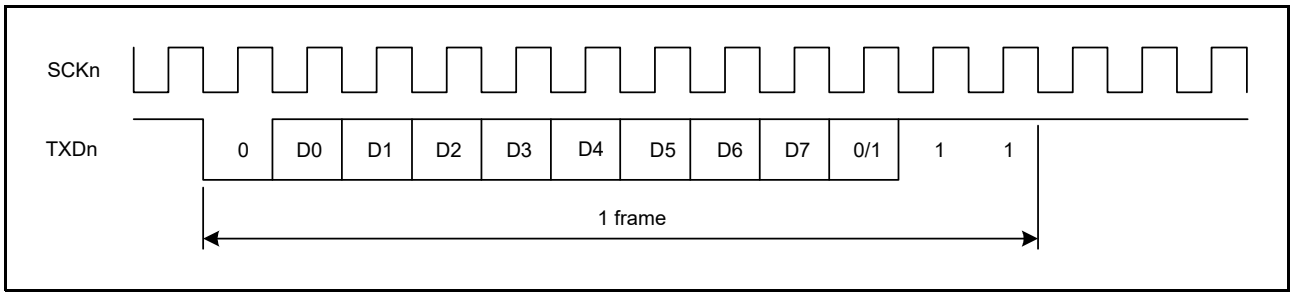


Figure 33.5 Phase Relationship between Output Clock and Transmit Data (Asynchronous Mode: SCR1.PE Bit = 1, SCR3.CHR[1:0] Bits = 10b, MP Bit = 0, STOP Bit = 1)

33.3.4 Double-Speed Mode and Divide-by-6 Mode

When SCR2.ABCS bit is set to 1, the RSCI operates on the bit rate twice that in the case where ABCS bit is set to 0. And when SCR2.BGDM bit is set to 1, the cycle of the base clock is halved and the bit rate is doubled from that in the case where BGDM bit is set to 0. When SCR3.CKE[1] bit is set to 0 and the on-chip baud rate generator is selected, setting the ABCS and BGDM bits to 1 allows the RSCI to operate on a bit rate four times that in the case ABCS bit = 0 and BGDM bit = 0.
When SCR2.ABCSE bit is set to 1, the number of base clock pulses are 6 during a period of 1 bit, and the base clock frequency is half. And RSCI works 16/3 times of bit rate compared with a case of SCR2.ABCS bit = 0, SCR2.BGDM bit = 0 and SCR2.ABCSE bit = 0.
As shown by Formula (1) in section 33.3.2, Receive Data Sampling Timing and Reception Margin in Asynchronous Mode, the reception margin decreases when SCR2.ABCS bit is set to 1 or SCR2.ABCSE bit is set to 1. Therefore, if the desired bit rate can be obtained with SCR2.ABCS bit set to 0 and SCR2.ABCSE bit set to 0, it is recommended to use the RSCI with SCR2.ABCS bit set to 0 and SCR2.ABCSE bit set to 0.

33.3.5 CTS and RTS Functions

The CTS function is the transmission control function by the CTSn# pin. Setting the SCR1.CTSE bit to 1 enables the CTS function. For the functions of CTS and RTS, you can select the alternate setting that uses either function with one pin or the dedicated setting that uses each function independently with two pins. This setting is done with the SCR1.CRSEP bit.
When the CTS function is enabled, placing the low level on the CTSn# pin causes transmission to start.
Even if the CTSn# pin goes high after transmission starts, the frame being transmitted is not affected and transmission will continue.
The RTS function is the transmission request function by the RTSn# pin. In the RTS function, the RTSn# pin output low level when reception becomes possible. Conditions for output of the low and high level are shown below.

(a) When the SCR3.FM Bit is 0 (Non-FIFO Mode)

[Conditions for low-level output]
When the following conditions are all satisfied

- The SCR0.RE bit is 1
- There are no received data yet to be read and reception is not in progress
- The ORER, AFER, and APER flags in the SSR are all 0

[Condition for high-level output]
When the conditions for low-level output are not satisfied

(b) When the SCR3.FM Bit is 1 (FIFO Mode)

[Conditions for low-level output]
When the following conditions are all satisfied

- The SCR0.RE bit is 1
- The number of data stored in the receive FIFO is less than the threshold (FCR.RSTRG[4:0] bits)
- The SSR.ORER (RDR.ORER) flag is 0

[Condition for high-level output]
When the conditions for low-level output are not satisfied

33.3.6 Data Match Detection

The data match detection function can use only the asynchronous mode.
If SCR0.DCME bit is set to 1*2, when one frame of data has been received, RSCI compares that receive data with the data which is set to SCR4.CMPD[8:0] bits. If RSCI detects the match to the comparison data (SCR4.CMPD[8:0] bits*1) with receive data, RSCI can issue RXI interrupt request.
If SCR3.MP bit is set to 0, this comparative target in communication data is valid only data field in receive format. In multi-processor mode (SCR3.MP bit = 1), if SCR0.IDSEL bit is set to 1, the reception data at which MPB is 1 detects data match or unmatch, and the reception data at which MPB is 0 detects always unmatch. If SCR0.IDSEL bit is set to 0, RSCI detects data match or unmatch despite the value of the MPB of the reception data at every reception complete.
Until RSCI detects the match to the comparison data (SCR4.CMPD[8:0] bits) with receive data, the communication data is skipped (discarded), and RSCI can not detect parity error, framing error. When RSCI detects the match, the SCR0.DCME bit is automatically cleared, and SSR.DCMF flag is set to 1. If SCR0.IDSEL bit is set to 1 at this time, SCR0.MPIE bit is automatically cleared. And if SCR0.IDSEL bit is set to 0 at this time, SCR0.MPIE bit is kept. At the same time, if SCR0.RIE bit is set to 1, RSCI issues RXI interrupt request. If RSCI detects framing error in comparative receive data which is detected the match, SSR.DFER flag is set to 1, and if RSCI detects parity error in that frame, SSR.DPER flag is set to 1. That comparative receive data and MPB are not stored to RDR register, and SSR.RDRF flag is retained to 0.
After RSCI detects the match, and the SCR0.DCME bit is automatically cleared, it receives next data continuously in current register setting.
When SSR.DFER flag or SSR.DPER flag is set, the data match isn't detected. Before making the data match detection function effective, please be sure to set SSR.DFER and SSR.DPER flags as 0.
Figure 33.6 and Figure 33.7 show the data match detection example.

Note 1. This comparative target can select one length of 3 types, they are CMPD[6:0] bits with 7bit length enable, CMPD[7:0] bits with 8bit, and CMPD[8:0] bits with 9bit length.
Note 2. Set the SCR0.DCME bit to 1 before receiving the start bit of the received frame that performs data matching.

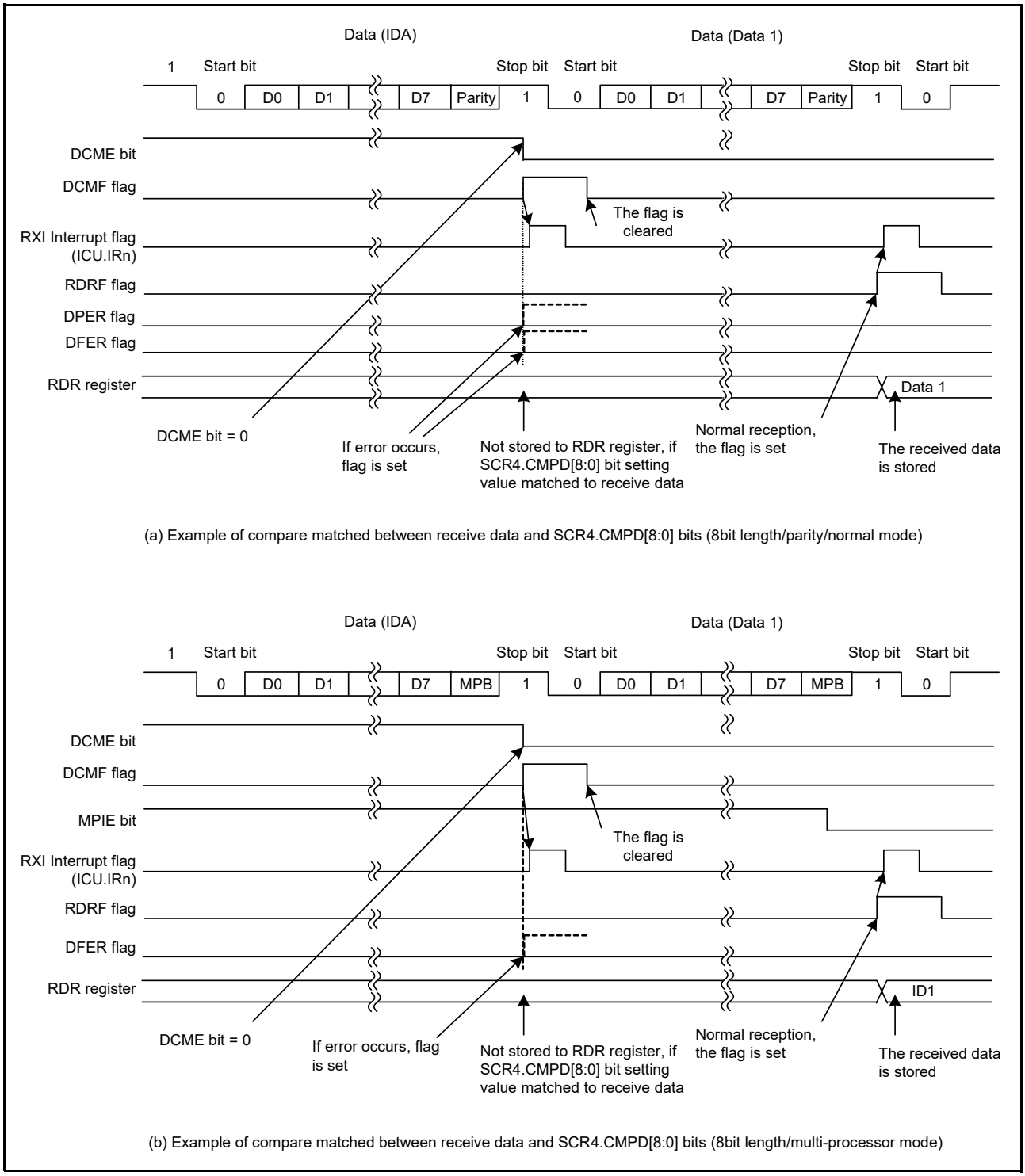


Figure 33.6 Example of Data Match Detection (1) 8bit-Data

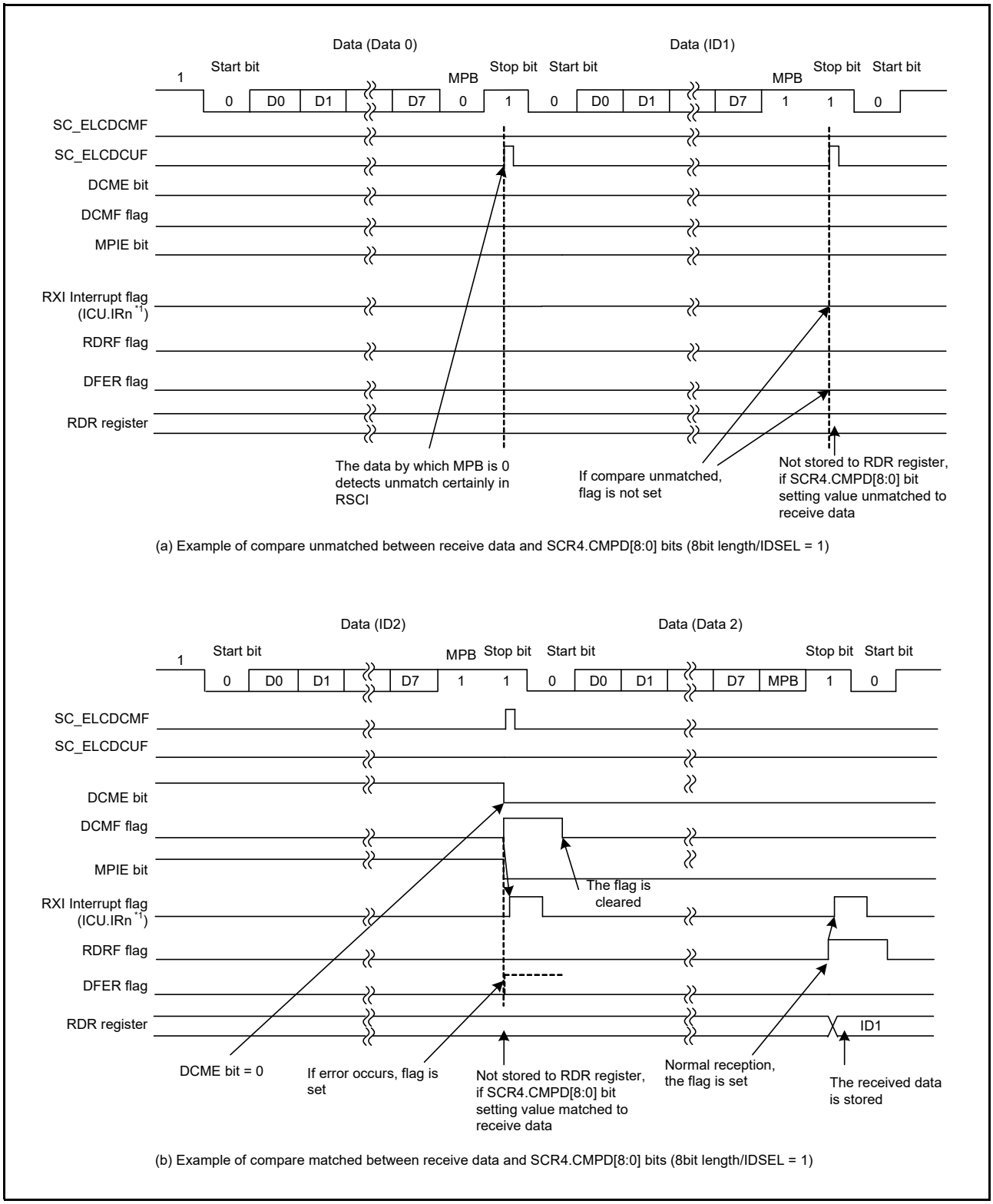


Figure 33.7 Example of Data Match Detection (2) Multi-Processor Mode/8bit-Data

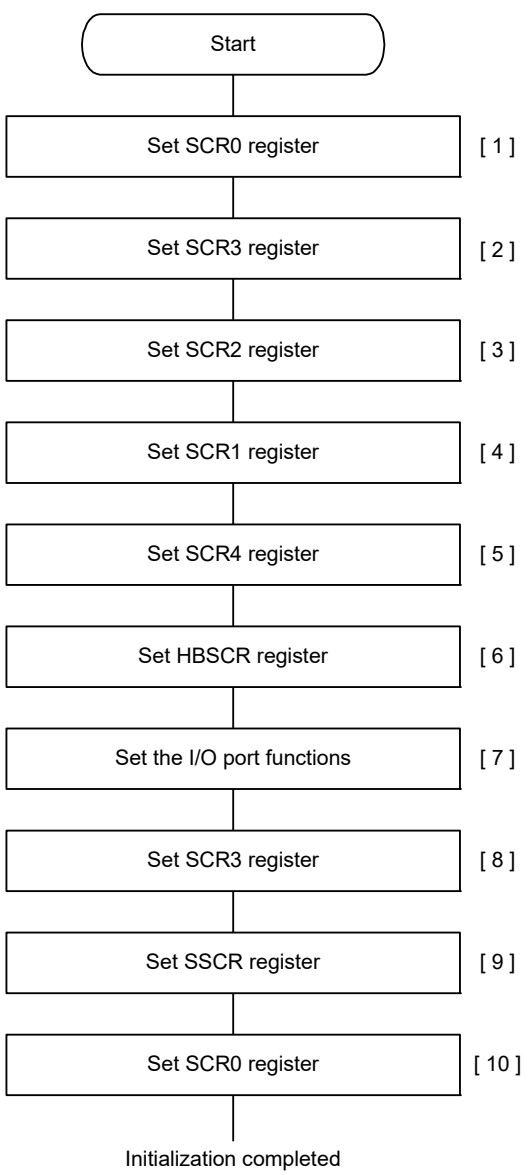
33.3.7 RSCI Initialization (Asynchronous Mode)

Before transmitting and receiving data, start by writing 0 to SCR0.TE and SCR0.RE bits (or writing the initial value to SCR0 register) and then continue through the procedure (select to non-FIFO or FIFO) for RSCI given in Figure 33.8 or Figure 33.9. Whenever the operating mode or transfer format is changed, SCR0.TE and SCR0.RE bits must be initialized before the change is made.

When the external clock is used in asynchronous mode, ensure that the clock signal is supplied even during initialization.

Note that setting the SCR0.RE bit to 0 initializes neither the ORER flags, AFER flags, APER flags, RDRF flags, DR flags, and RDR register. In FIFO mode, even if the TE bit is set as 0, the TEND flag isn't initialized, so please be careful. Please be also careful at the time of change in the operation mode.

Moreover, note that switching the value of the SCR0.TE bit from 0 to 1 while the SCR0.TIE bit is 1 leads to the generation of a TXI interrupt request.



[1] Set the following bits of SCR0 register to 0.
TEIE, TIE, RIE, TE, RE
If you have not changed from the initial settings, you can skip this step.

[2] Set up following function and communication mode.
Driver control function for RS-485
FIFO no-use
Multi-Processor mode
Communication mode (MOD[2:0] bits = 000b)
Transmission/reception format
Clock enable (Leave the CKE[1:0] bits at their initial values when outputting the clock)
Leave unused bits at their initial values.

[3] Set up the bit-rate-modulation function*1, 2, select clock, set bit rate*2.

[4] Set up the Noise-filter function, the loop-back function, communication terminal status, the parity check function, and the CTS/RTS function.

[5] Set up the adjust sampling timing function and the adjust transmit timing function.

[6] When using the HBS support mode, enable or set the function.
If this function is not used, leave the default value.

[7] Make I/O port settings to enable input and output functions as required for TXDn (TXDAn/TXDBn), RXDn, and SCKn pins.

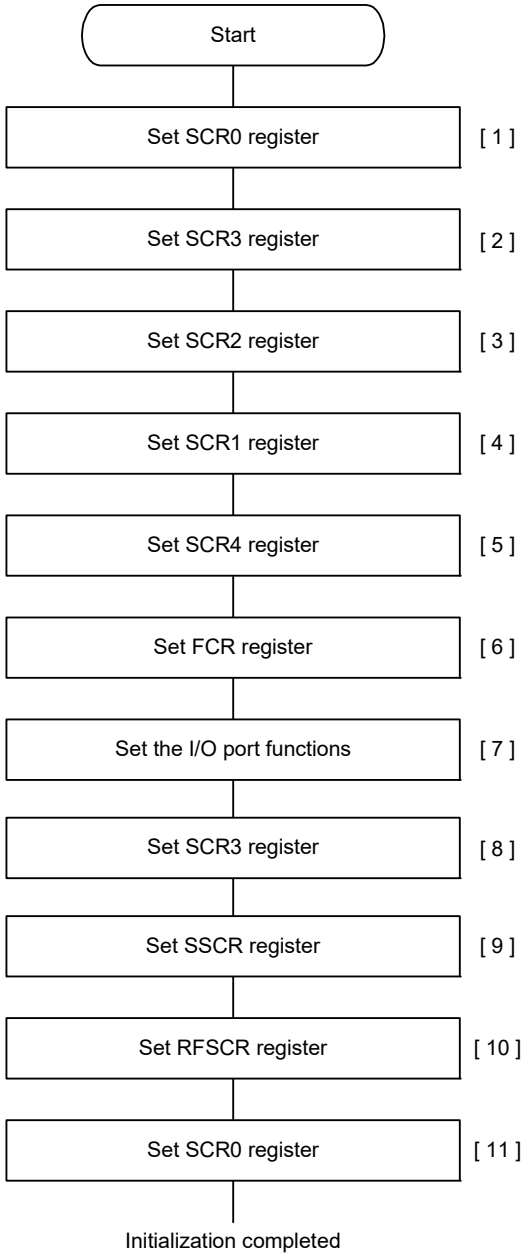
[8] Set clock enable bit (CKE[1:0] bits) at this step when outputting the clock. After setting the bits, the SCKn pin becomes output mode (the clock is output after setting the TE or RE bit to 1).

[9] Write 1 to the following bits and clear the corresponding flag.
SSCR.RDRFC, AFERC, APERC, MFFC, ORERC, DFERC, DPERC, DCMFC, ERSC
If it's initialization flow after a reset, you can skip this step.

[10] Set the TE or RE bit to 1.
To enable interrupts, set the TE bit and TIE bit, and the RE bit and RIE bit to 1 with one instruction at the same time. Also, set the data match detection function and multiprocessor function.
Setting the TE and RE bits allows TXDn (TXDAn/TXDBn) and RXDn to be used.

Note 1. If you do not use the bit rate modulation function, you do not need to set it.
Note 2. If you use an external clock, you do not need to set it.

Figure 33.8 Sample RSCI Initialization Flowchart (Asynchronous Mode/Non-FIFO Mode)



[1] Set the following bits of SCR0 register to 0.
TEIE, TIE, RIE, TE, RE
If you have not changed from the initial settings, you can skip this step.

[2] Set up following function and communication mode.
Driver control function for RS-485
FIFO use
Multi-Processor mode
Communication mode (MOD[2:0] bits = 000b)
Transmission/reception format
Clock enable (Leave the CKE[1:0] bits at their initial values when outputting the clock)
Leave unused bits at their initial values.

[3] Set up the bit-rate-modulation function*1, 2,select clock, set bit rate*2.

[4] Set up the Noise-filter function, the loop-back function, communication terminal status, the parity check function, and the CTS/RTS function.

[5] Set up the adjust sampling timing function and the adjust transmit timing function.

[6] Set the TFRST and RFRST to 1 to empty FIFO.
Set the DRES, TTRG[4:0], RTRG[4:0], and RSTRG[4:0] bits

[7] Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.

[8] Set clock enable bit (CKE[1:0] bits) at this step when outputting the clock. After setting the bits, the SCKn pin becomes output mode (the clock is output after setting the TE or RE bit to 1).

[9] Write 1 to the following bits and clear the corresponding flag.
SSCR.RDRFC, AFERC, APERC, MFFC, ORERC, DFERC, DPERC, DCMFC, ERSC
If it is initialization flow after a reset, you can skip this step.

[10] Write 1 to The RFSCR.DRC bit and clear the corresponding flag.
If it is initialization flow after a reset, you can skip this step.

[11] Set the TE or RE bit to 1.
To enable interrupts, set the TE bit and TIE bit, and the RE bit and RIE bit to 1 with one instruction at the same time. Also, set the data match detection function and multiprocessor function.
Setting the TE and RE bits allows TXDn and RXDn to be used.

Note 1. If you do not use the bit rate modulation function, you do not need to set it.
Note 2. If you use an external clock, you do not need to set it.

Figure 33.9 Sample RSCI Initialization Flowchart (Asynchronous Mode, FIFO Mode)

Figure 33.10 shows an example of the timing when data is transmitted after reset is released, and RSCI is set to asynchronous mode according to Figure 33.8 or Figure 33.9. As shown in the figure, when the pin function is set to the TXDn pin, the SCR0.TE bit is 0, so the pin is high impedance. When transmit data is written after setting the SCR0.TE bit to 1, data transmission starts. There is a transmission wait time from writing TDR register to data transmission starts. In asynchronous mode, TXDn is high during this period.

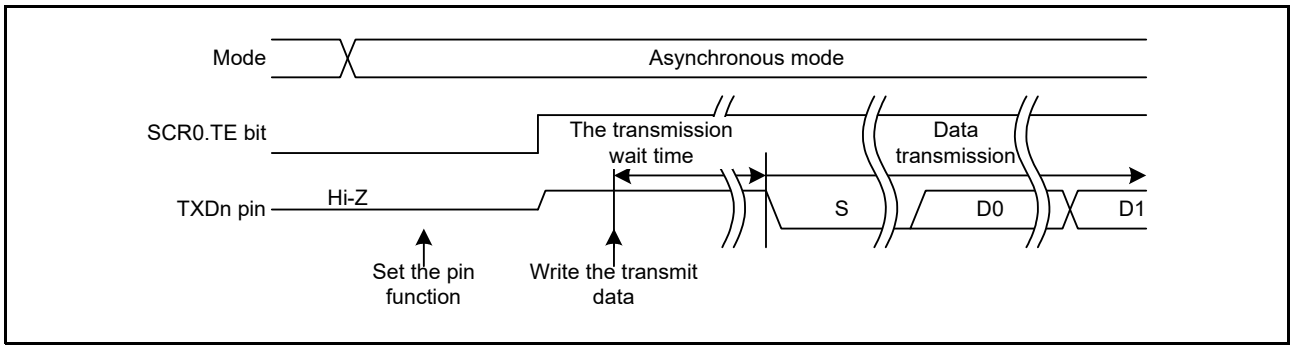


Figure 33.10 Example of Data Transmission Timing in Asynchronous Mode

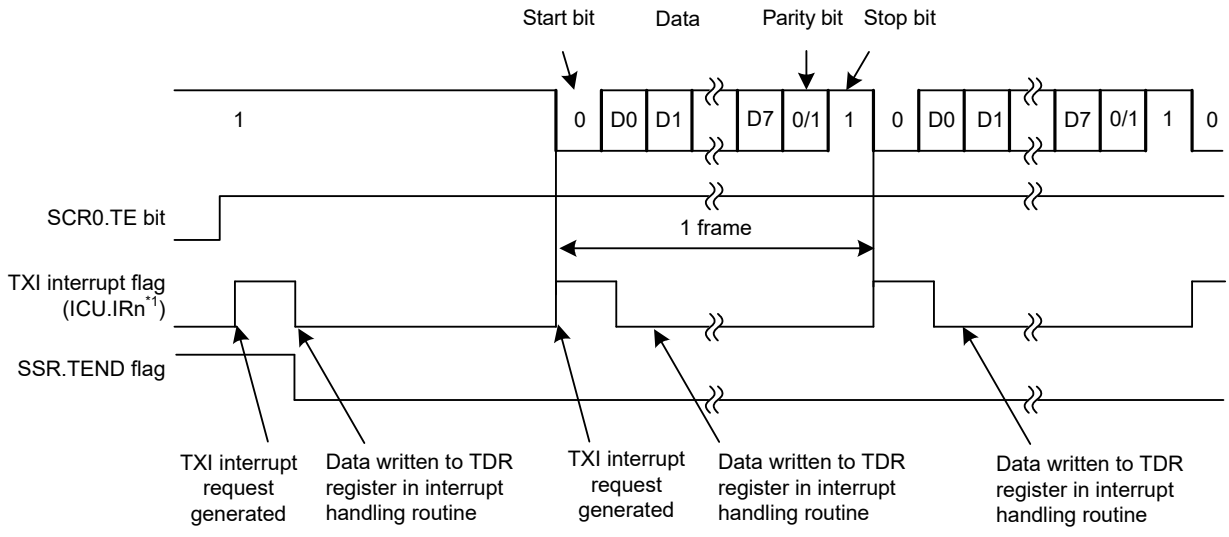
33.3.8 Serial Data Transmission (Asynchronous Mode)

(1) Non-FIFO Mode

Figure 33.11 to Figure 33.13 show an example of the operation for serial transmission in asynchronous mode.
In serial transmission, the RSCI operates as described below.

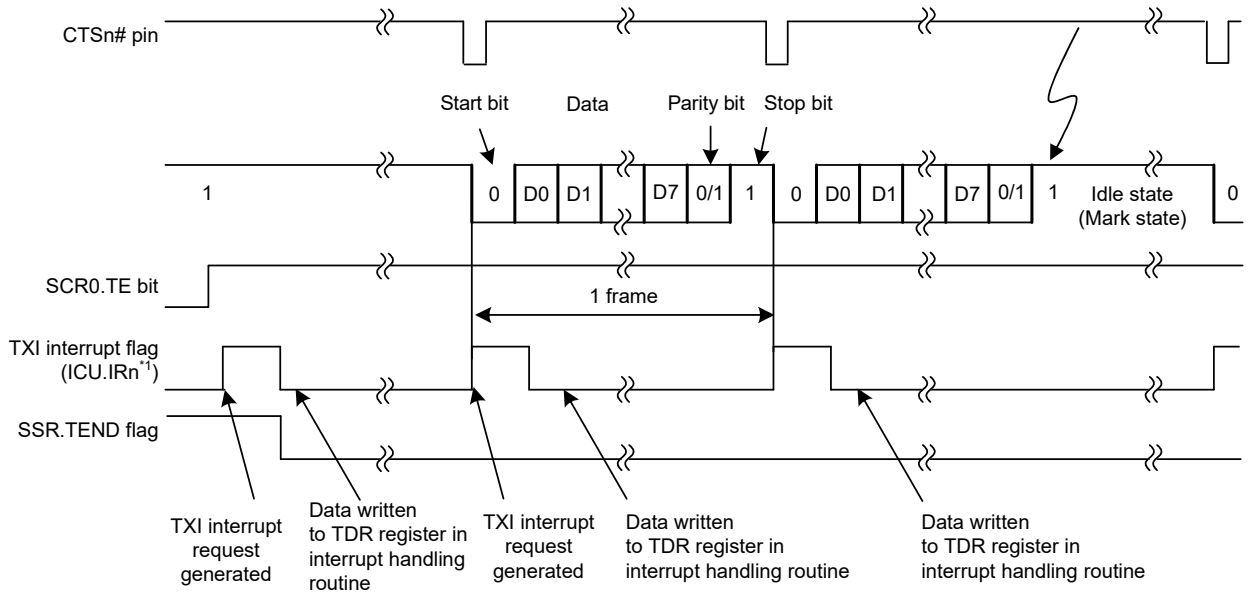
1. The RSCI transfers data from the TDR register to the TSR register when data is written to the TDR register in the TXI interrupt handling routine. At the beginning of transmission, set 1 to the SCR0.TE bit and the SCR0.TIE bit simultaneously. Then the TXI interrupt request is generated.
2. Transmission starts at data transfer from the TDR register to the TSR register when the SCR1.CTSE bit = 0 (CTS function is disabled) or when the CTS# pin level is low. If the SCR0.TIE bit is 1 at this time, a TXI interrupt request is generated. Continuous transmission is obtainable by writing the next transmit data to the TDR register in the TXI interrupt handling routine before transmission of the current transmit data is completed. Write the last transmission data, then the last data transmit start, and TXI is generated. When TEI interrupt requests are in use, before the last data transmit end, set the SCR0.TIE bit to 0 (a TXI interrupt request is disabled) and the SCR0.TEIE bit to 1 (a TEI interrupt request is enabled) using the TXI interrupt handling routine.
3. Data is sent from the TXDn pin in the following order: start bit, transmit data, parity bit or multi-processor bit (may be omitted depending on the format), and stop bit.
4. The RSCI checks for updating of (writing to) the TDR register at the time of stop bit output.
5. When the TDR register is updated, setting of the SCR1.CTSE bit to 0 (CTS function is disabled) or a low level input on the CTSn# pin cause the next transfer of the next transmit data from the TDR register to the TSR register and sending of the stop bit, after which serial transmission of the next frame starts.
6. If the TDR register is not updated, the SSR.TEND flag is set to 1, the stop bit is sent, and then the mark state is entered in which 1 is output. If the SCR0.TEIE bit is 1 at this time, a TEI interrupt request is generated.

Figure 33.15 shows the example of Serial Transmission Flowchart.



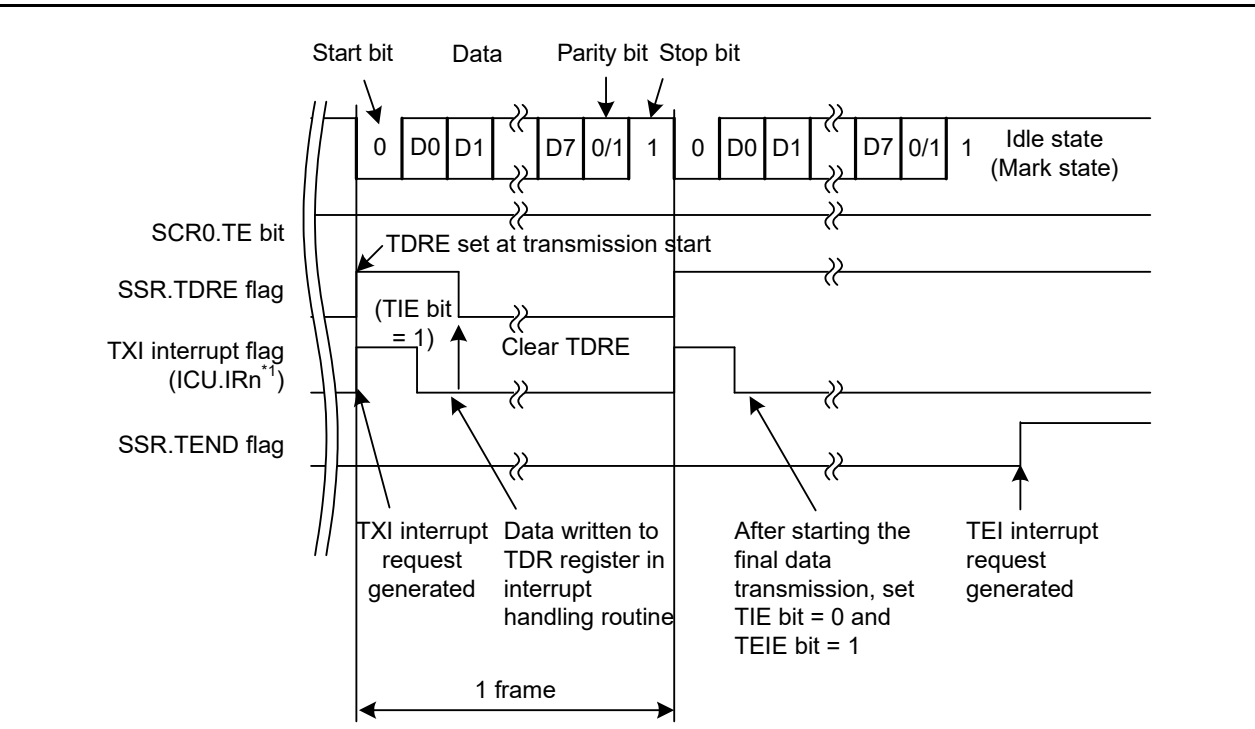
Note 1. Refer to section 14, Interrupt Controller (ICUG) for details on the corresponding interrupt vector number.

**Figure 33.11 Example of Operation for Serial Transmission in Asynchronous Mode (1)
(with 8-Bit Data, Parity, 1 Stop Bit, CTS Function Not Used, at the Beginning of Transmission)**



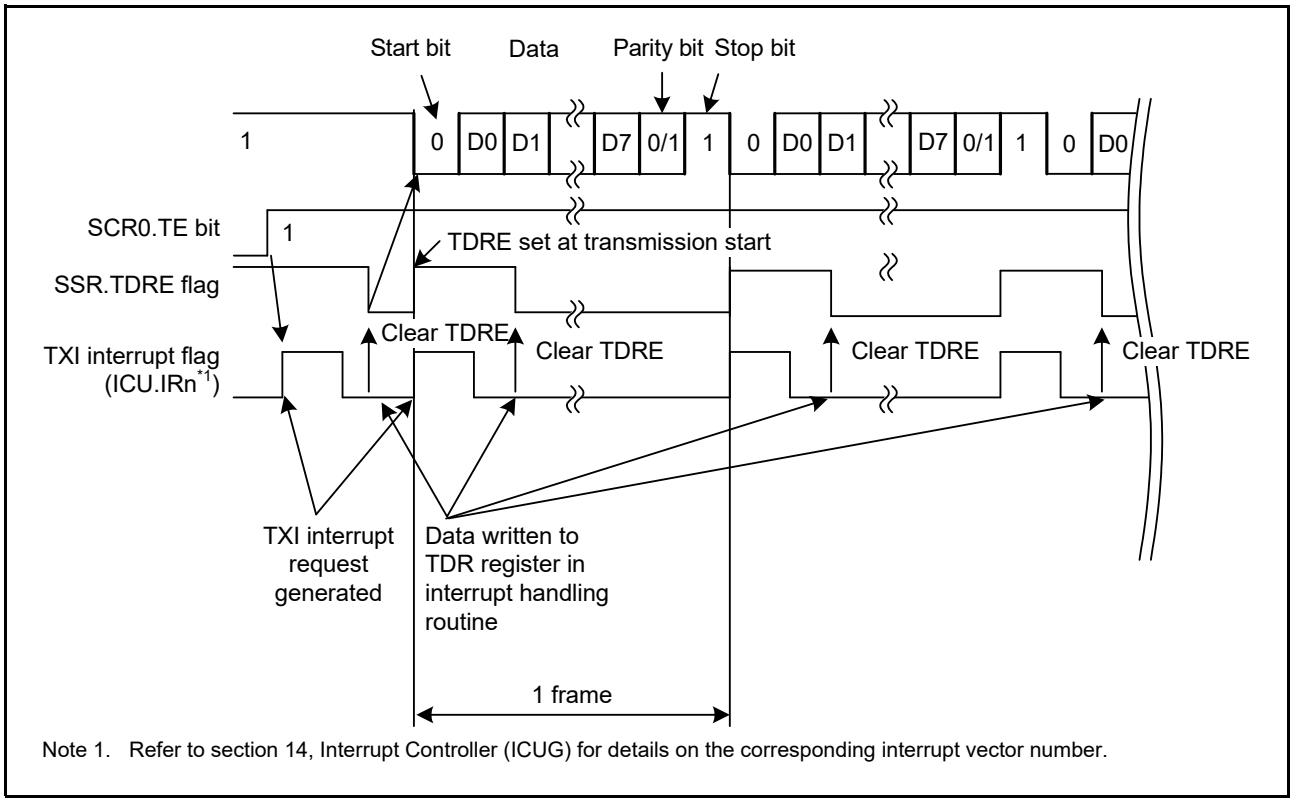
Note 1. Refer to section 14, Interrupt Controller (ICUG) for details on the corresponding interrupt vector number.

Figure 33.12 Example of Operation for Serial Transmission in Asynchronous Mode (2) (with 8-Bit Data, Parity, 1 Stop Bit, CTS Function Used, at the Beginning of Transmission)



Note 1. Refer to section 14, Interrupt Controller (ICUG) for details on the corresponding interrupt vector number.

Figure 33.13 Example of Operation for Serial Transmission in Asynchronous Mode (3) (with 8-Bit Data, Parity, 1 Stop Bit, CTS Function Not Used, from the Middle of Transmission until Transmission Completion)



Note 1. Refer to section 14, Interrupt Controller (ICUG) for details on the corresponding interrupt vector number.

Figure 33.14 Example of Operation for Serial Transmission in Asynchronous Mode (4) (with 8-Bit Data, Parity, 1 Stop Bit, CTS Function Not Used, during Transmission)

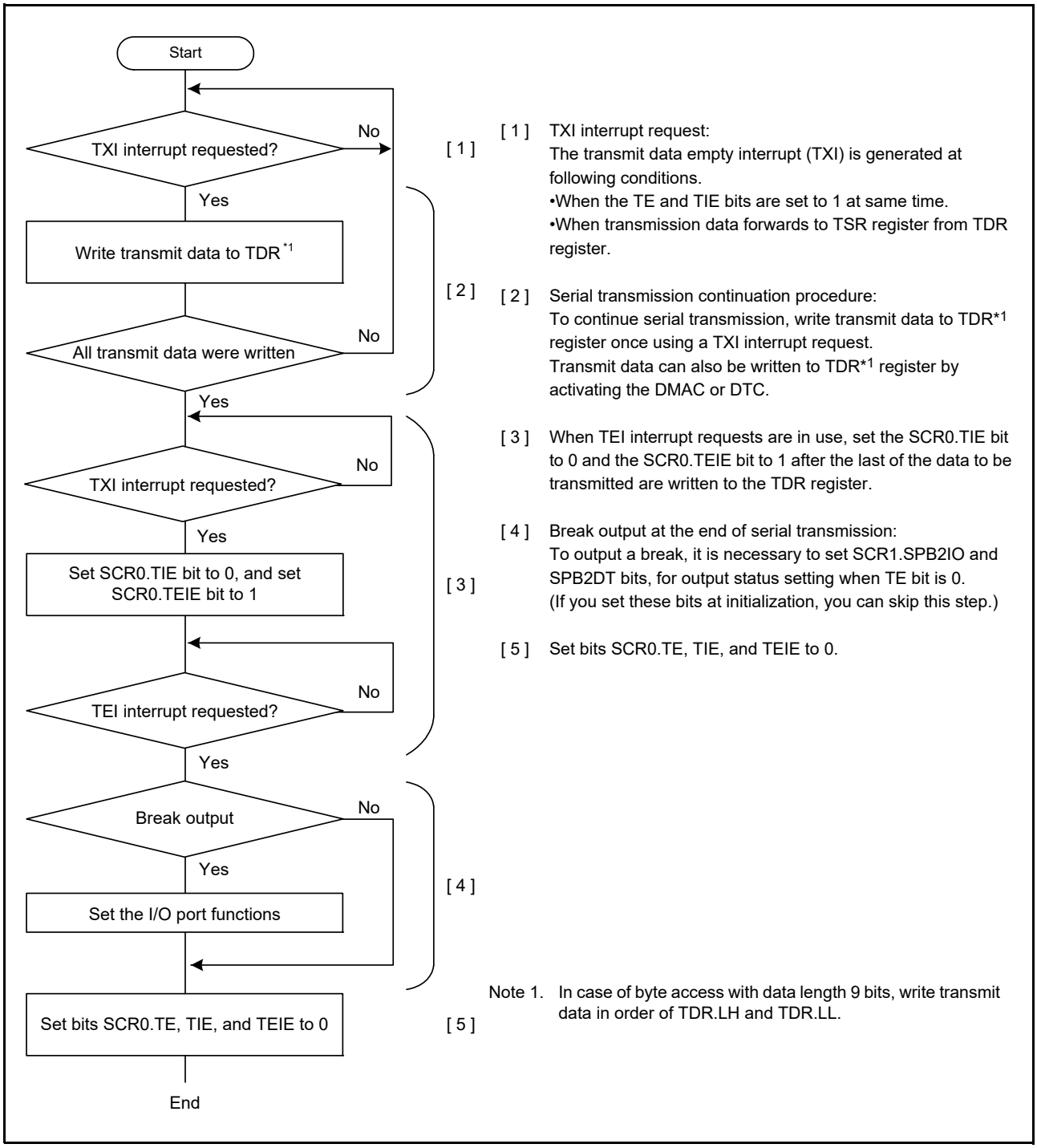


Figure 33.15 Example of Serial Transmission Flowchart in Asynchronous Mode (Non-FIFO Mode)

(2) FIFO Mode

Table 33.29 shows an example of data format that is written to transmit FIFO (TDR register) in asynchronous mode with FIFO selected.

Write the MPBT bit to bit 9 of the transmit FIFO (TDR register). When 7-bit data length is selected, write data to the TDR.TDAT[6:0] bits; when 8-bit data length is selected, write data to the TDR.TDAT[7:0] bits; when 9-bit data length is selected, write data to the TDR.TDAT[8:0] bits. Write 0 to unused bits. When accessing in byte units, write data in the order of the TDR.LH to the TDR.LL.

Table 33.29 Data Format That is Written to Transmit FIFO (TDR) (FIFO Mode)

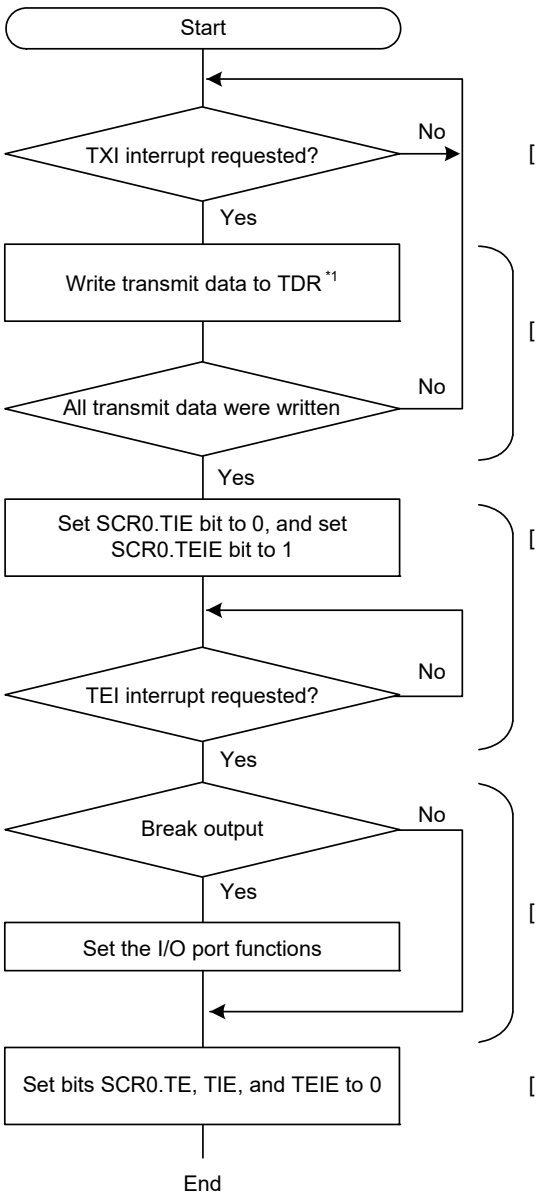
Data Length	Register setting		Transmit Data in TDR.L															
	SCR3	CHR[1:0]	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
7 bits	1	1	—	—	—	—	—	—	MPBT	—	—	TDAT[6:0]						
8 bits	1	0	—	—	—	—	—	—	MPBT	—	TDAT[7:0]							
9 bits	0	0 or 1	—	—	—	—	—	—	MPBT	TDAT[8:0]								

—: Do not used. It should write to 0.

In serial transmission, the RSCI operates as described below.

1. The RSCI transfers data from transmit FIFO (TDR register) to TSR register when data is written to transmit FIFO (TDR register) in the TXI interrupt handling routine. The writable transmit data number is until (32 – transmit FIFO (TDR register)) bytes. At the beginning of transmission, set 1 to SCR0.TE and SCR0.TIE bits simultaneously. Then the TXI interrupt request is generated.
2. Transmission starts at data transfer from transmit FIFO (TDR register) to TSR register when SCR1.CTSE bit = 0 (CTS function is disabled) or when the CTS# pin level is low. When the quantity of transmit data written in transmit FIFO (TDR register) is equal to or less than the specified transmit triggering number, SSR.TDRE flag is set to 1. If SCR0.TIE bit is 1 at this time, a TXI interrupt request is generated. Continuous transmission is obtainable by writing the next transmit data to transmit FIFO (TDR register) in the TXI interrupt handling routine before transmission of the current transmit data is completed. Write the last transmission data, then the last data transmit start, and TXI is generated. When TEI interrupt requests are in use, before the last data transmit end, set the SCR0.TIE bit to 0 (a TXI interrupt request is disabled) and the SCR0.TEIE bit to 1 (a TEI interrupt request is enabled) using the TXI interrupt handling routine.
3. Data is sent from the TXDn pin in the following order: start bit, transmit data, parity bit or multi-processor bit (may be omitted depending on the format), and stop bit.
4. The RSCI checks whether non-transmitted data in transmit FIFO (TDR register) or not at the time of stop bit output.
5. When data is set to transmit FIFO (TDR register), setting of SCR1.CTSE bit to 0 (CTS function is disabled) or a low level input on the CTSn# pin cause the next transfer of the next transmit data from transmit FIFO (TDR register) to TSR register and sending of the stop bit, after which serial transmission of the next frame starts.
6. If data is not set to transmit FIFO (TDR register), SSR.TEND flag is set to 1, the stop bit is sent, and then the mark state is entered in which 1 is output. If SCR0.TEIE bit is 1 at this time, a TEI interrupt request is generated.

Figure 33.16 shows a sample flowchart for serial transmission in asynchronous mode at FIFO selected.



[1] TXI interrupt request:
The transmit data empty interrupt (TXI) is generated at following conditions.
•When the TE and TIE bits are set to 1 at same time.
•When transmission data forwards to TSR from the transmit FIFO (TDR register) and the number of transmit data written in the transmit FIFO is equal to or less than the transmit FIFO threshold.

[2] Serial transmission continuation procedure:
To continue serial transmission, write all transmit data to the transmit FIFO (TDR register)*1 by using a TXI interrupt and clear the SSR.TDRE flag to 0. (Transmit data can also be written to the transmit FIFO (TDR register)*1 by activating the DMAC or DTC. In this case, the TDRE flag is cleared automatically, so please don't operate the TDRE flag.)
The number of transmission data it's possible to write in is 32 (the number of stored transmit FIFO data).

[3] If you use TEI, set TIE bit = 0 and TEIE bit = 1 after writing the last transmit data to TDR*1 register.

[4] Break output at the end of serial transmission:
To output a break, it is necessary to set SCR1.SPB2IO and SPB2DT bits, for output status setting when TE bit is 0.
(If you set these bits at initialization, you can skip this step.)

[5] Set bits SCR0.TE, TIE, and TEIE to 0.

Note 1. In case of byte access with data length 9 bits, write transmit data in order of TDR.LH and TDR.LL.

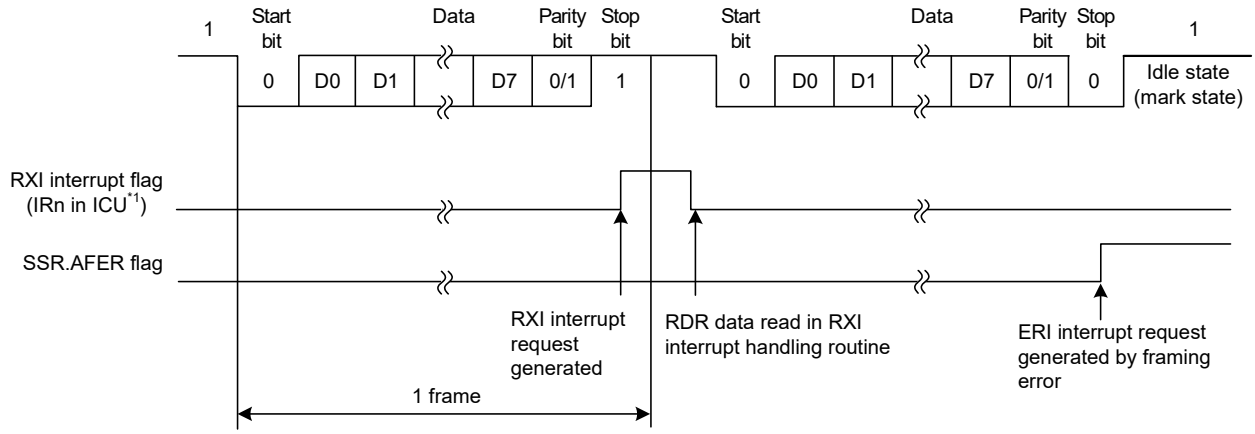
Figure 33.16 Example of Serial Transmission Flowchart in Asynchronous Mode (FIFO Mode)

33.3.9 Serial Data Reception (Asynchronous Mode)

(1) Non-FIFO Mode

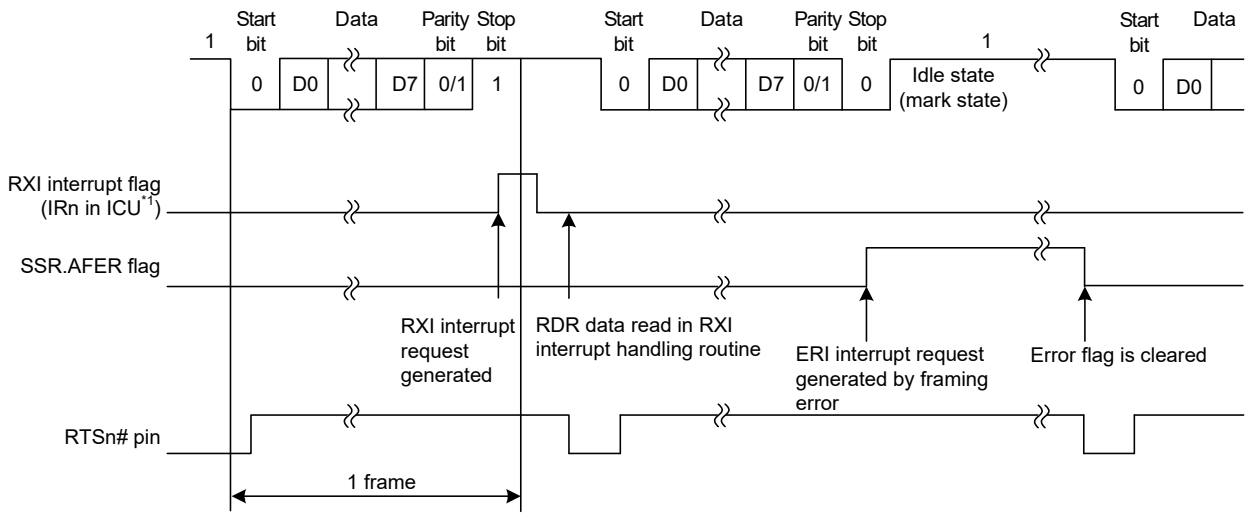
Figure 33.17 and Figure 33.18 show an example of the operation for serial data reception in asynchronous mode. In serial data reception, the RSCI operates as described below.

1. When the value of SCR0.RE bit becomes 1, the output signal on the RTSn# pin goes to the low level in the case of RTS function use.
2. When the RSCI monitors the communications line and detects a start bit, it performs internal synchronization, stores receive data in RSR register, and checks the parity bit and stop bit.
3. If an overrun error occurs, SSR.ORER flag is set to 1. If SCR0.RIE bit is 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to RDR register.
4. If a parity error is detected, SSR.APER flag is set to 1 and receive data is transferred to RDR register. If the SCR0.RIE bit is 1 at this time, an ERI interrupt request is generated.
5. If a framing error (when the stop bit is 0) is detected, SSR.AFER flag is set to 1 and receive data is transferred to RDR register. If SCR0.RIE bit is 1 at this time, an ERI interrupt request is generated.
6. When reception finishes successfully, receive data is transferred to RDR register. If SCR0.RIE bit is 1 at this time, an RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to RDR register in this RXI interrupt handling routine before reception of the next receive data is completed. Reading the received data that have been transferred to RDR register causes the RTSn# pin to output the low level in the case of RTS function use. If you do not want to turn the RTSn # pin output low after receiving the last data, set SCR0.RE bit to 0, before reading the RDR register.



Note 1. Refer to section 14, Interrupt Controller (ICUG) for details on the corresponding interrupt vector number.

Figure 33.17 Example of RSCI Operation for Serial Reception in Asynchronous Mode (1) (8-Bit Data, Parity, 1 Stop Bit, RTS Function is Not Used)



Note 1. Refer to section 14, Interrupt Controller (ICUG) for details on the corresponding interrupt vector number.

Figure 33.18 Example of RSCI Operation for Serial Reception in Asynchronous Mode (2) (8-Bit Data, Parity, 1 Stop Bit, RTS Function is Used)

Table 33.30 lists the states of the flags in SSR status register and receive data handling when a receive error is detected. If a receive error is detected, an ERI interrupt request is generated but an RXI interrupt request is not generated. Data reception cannot be resumed while the receive error flag is 1. Accordingly, set the ORER, AFER, and APER flags to 0 before resuming reception. Moreover, be sure to read the RDR register during overrun error processing. When a reception is forcibly terminated by setting SCR0.RE bit to 0 during operation, read the RDR register because received data which has not yet been read may be left in RDR register.
Figure 33.19 and Figure 33.20 show samples of flowcharts for serial data reception.

Table 33.30 Flags in the SSR Status Register and Receive Data Handling

Flags in the SSR Status Register				
ORER	**AFER**	**APER**	**Receive Data**	**Receive Error Type**
1	0	0	Lost	Overrun error
0	1	0	Transferred to RDR	Framing error
0	0	1	Transferred to RDR	Parity error
1	1	0	Lost	Overrun error + framing error
1	0	1	Lost	Overrun error + parity error
0	1	1	Transferred to RDR	Framing error + parity error
1	1	1	Lost	Overrun error + framing error + parity error

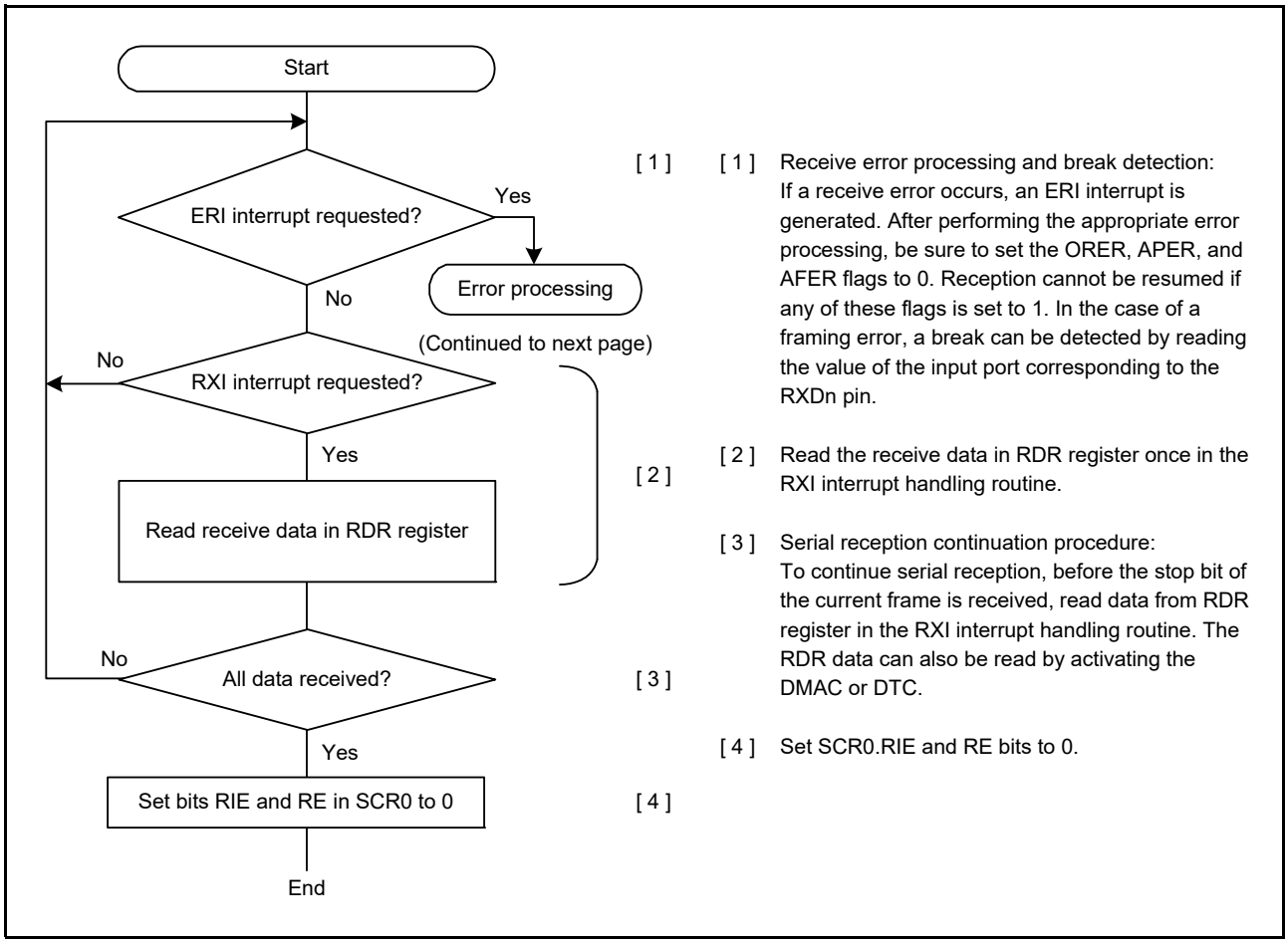


Figure 33.19 Example Flowchart of Serial Reception in Asynchronous Mode (Non-FIFO Mode) (1)

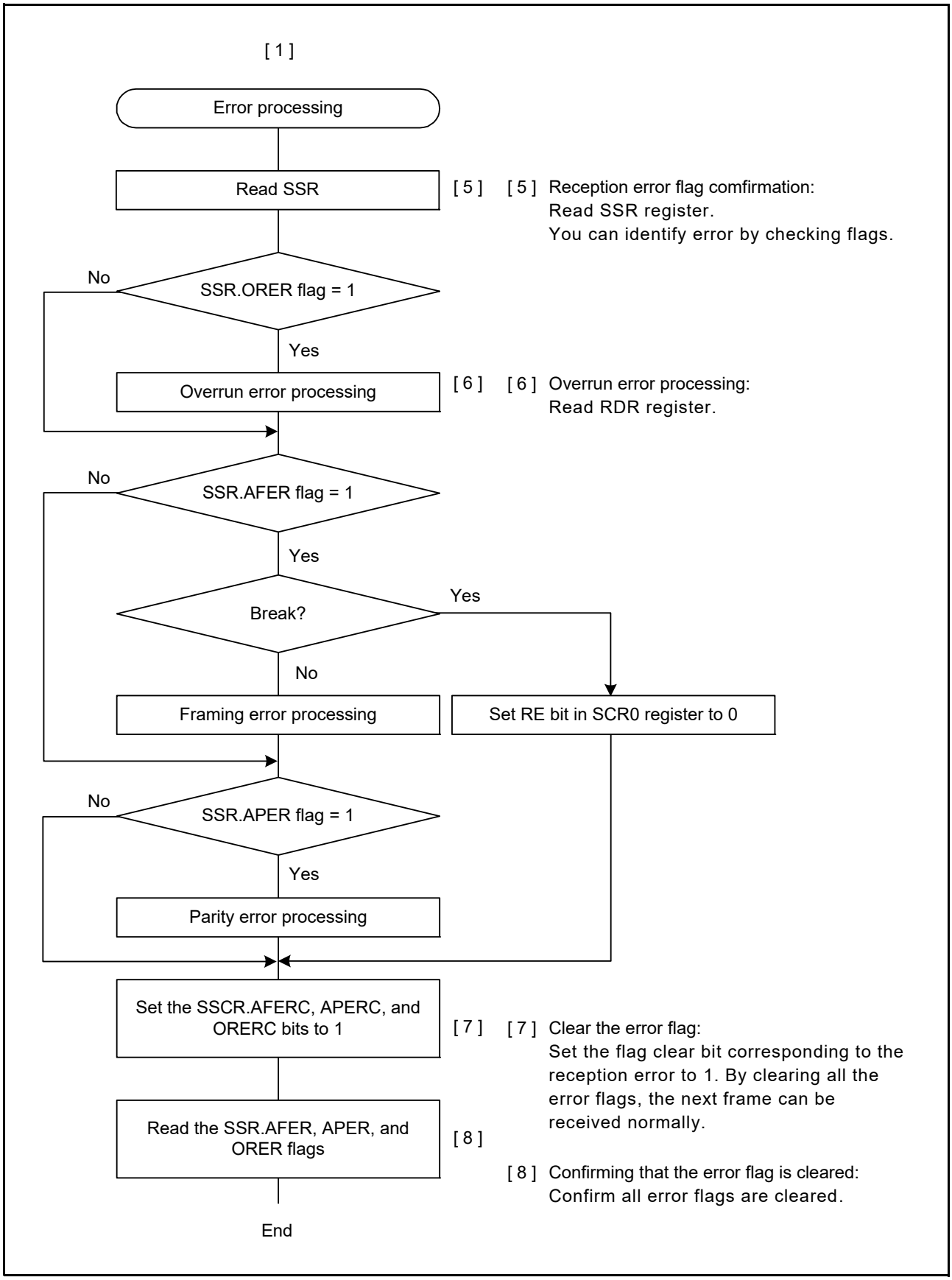


Figure 33.20 Example Flowchart of Serial Reception in Asynchronous Mode (Non-FIFO Mode) (2)

(2) FIFO Mode

Table 33.31 shows an example of data format stored to receive FIFO (RDR register) in asynchronous mode.
MPB flag (Receive FIFO (RDR register) bit9) is stored 0. Data is stored to receive FIFO (RDR register) corresponded to data length. It is stored to 0 for unused bits. If receive FIFO (RDR register) is read, RSCI updates to next data which are FER flag, PER flag, and receive data (RDAT[8:0] bits) in receive FIFO. The flags which are AFER, APER, ORER, and DR flag in receive FIFO, are always indicated to the flags corresponded to SSR register and RFSR register.

Table 33.31 Data Format Stored in the Receive FIFO (RDR) (FIFO Mode)

Data Length	Register setting		Arrangement of Receive Flag, MPB Flag and Received Data in RDR Register															
	SCR3	CHR[1:0]	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
7 bits	1	1	—	—	—	FER	PER	DR	MPB	0	0	RDAT[6:0]						
8 bits	1	0	—	—	—	FER	PER	DR	MPB	0	RDAT[7:0]							
9 bits	0	0 or 1	—	—	—	FER	PER	DR	MPB	RDAT[8:0]								
Data Length	**SCR3**	**CHR[1:0]**	**b31**	**b30**	**b29**	**b28**	**b27**	**b26**	**b25**	**b24**	**b23**	**b22**	**b21**	**b20**	**b19**	**b18**	**b17**	**b16**
7 bits	1	1	—	—	—	AFER	APER	—	—	ORER	—	—	—	—	—	—	—	—
8 bits	1	0	—	—	—	AFER	APER	—	—	ORER	—	—	—	—	—	—	—	—
9 bits	0	0 or 1	—	—	—	AFER	APER	—	—	ORER	—	—	—	—	—	—	—	—

Note: 0 is always read from the MPB flag (RDR register bit9).
When a 7-bit data length is selected, 0 is read from the RDAT[8:7] bits.
When a 8-bit data length is selected, 0 is read from the RDAT[8] bit.

Table 33.32 lists the states of the flags in SSR register status register and receive data handling when a receive error is detected in FIFO mode. Figure 33.21 and Figure 33.22 show samples of flowcharts for serial data reception in FIFO mode.
In serial data reception, the RSCI operates as described below.

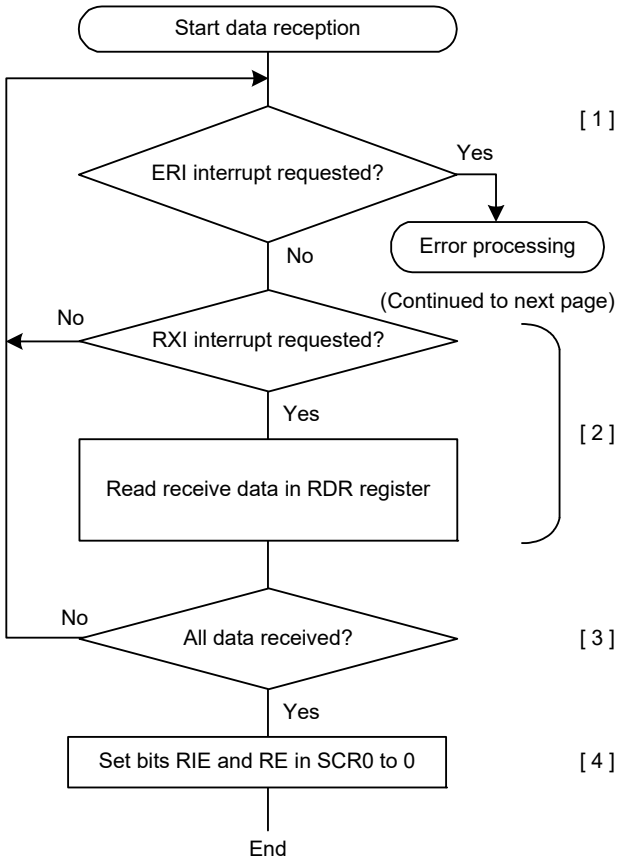
1. When the value of SCR0.RE bit becomes 1, the output signal on the RTSn# pin goes to the low level in the case of RTS function use.
2. When the RSCI monitors the communications line and detects a start bit, it performs internal synchronization, stores receive data in RSR, and checks the parity bit and stop bit.
3. If there is no space in receive FIFO (RDR register), an overrun error occurs and the SSR.ORER flag is set to 1. If SCR0.RIE bit is 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to receive FIFO (RDR register).
4. If a parity error is detected, SSR.APER flag is set to 1 and receive data is transferred to receive FIFO (RDR register). If the SCR0.RIE bit is 1 at this time, an ERI interrupt request is generated.
5. If a framing error (when the stop bit is 0) is detected, SSR.AFER flag is set to 1 and receive data is transferred to RDR register. If SCR0.RIE bit is 1 at this time, an ERI interrupt request is generated.
6. After framing error is detected, when RSCI detects that continuous receive data is 0 for 1 frame, the reception stops.
7. When quantity of data stored in the receive FIFO data register (RDR register) falls the specified receive triggering number, and that no next data has been received yet after the elapse of 15 etu from the last stop bit in asynchronous mode, RFSR.DR flag is set to 1. If RIE bit is set to 1, RSCI occurs RXI interrupt request when FCR.DRES bit is 0 and RSCI occurs ERI interrupt request when FCR.DRES bit is 1.
8. When reception finishes successfully, receive data is transferred to receive FIFO (RDR register). The SSR.RDRF flag is set to 1 when the quantity of receive data which is equal to or greater than the specified receive triggering number are stored in receive FIFO (RDR register). If SCR0.RIE bit is 1 at this time, an RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to receive FIFO (RDR register)

in this RXI interrupt handling routine, before overrun error is occurred. Reading the received data that have been transferred to receive FIFO (RDR register), and if it is less than RTS trigger number, causes the RTSn# pin to output the low level. in the case of RTS function use.

Table 33.32 Flags in the SSR Status Register and Receive Data Handling (FIFO Mode)

SSR Register				
ORER	**AFER*1**	**APER*1**	**Receive Data**	**Receive Error Type**
1	0	0	Lost	Overrun error
0	1	0	Transferred to RDR	Framing error
0	0	1	Transferred to RDR	Parity error
1	1	0	Lost	Overrun error + framing error
1	0	1	Lost	Overrun error + parity error
0	1	1	Transferred to RDR	Framing error + parity error
1	0	0	Lost	Overrun error + framing error + parity error

Note 1. This flag indicates whether there is an error in received data when reception is completed.



[1] Receive error processing and break detection:
If a receive error occurs, an ERI interrupt is generated. After performing the appropriate error processing, be sure to set the ORER flag to 0. Reception cannot be resumed if ORER flag is set to 1. Reception can continue even if APER, AFER, and DR flags are 1.

[2] Read the receive data in the receive FIFO (RDR register) in the RXI interrupt handling routine. The reception data stored in the receive FIFO (RDR register) is read until the number of stored data will be below the FCR.RTRG value at least. You can confirm the number of the reception data in the FIFO by RFSR.R[5:0] bits.

[3] Serial reception continuation procedure:
To continue serial reception, before an overrun error occurs, please read data from the receive FIFO (RDR register) in the RXI interrupt handling routine and clear SSR.RDRF flag and RFSR.DR flag to 0. The receive FIFO data can also be read by activating the DMA or DTC. The RDRF flag is cleared automatically in this case, so please don't operate the RDRF flag.

[4] Set SCR0.RIE and RE bits to 0.

Figure 33.21 Example Flowchart of Serial Reception in Asynchronous Mode (FIFO Mode) (1)

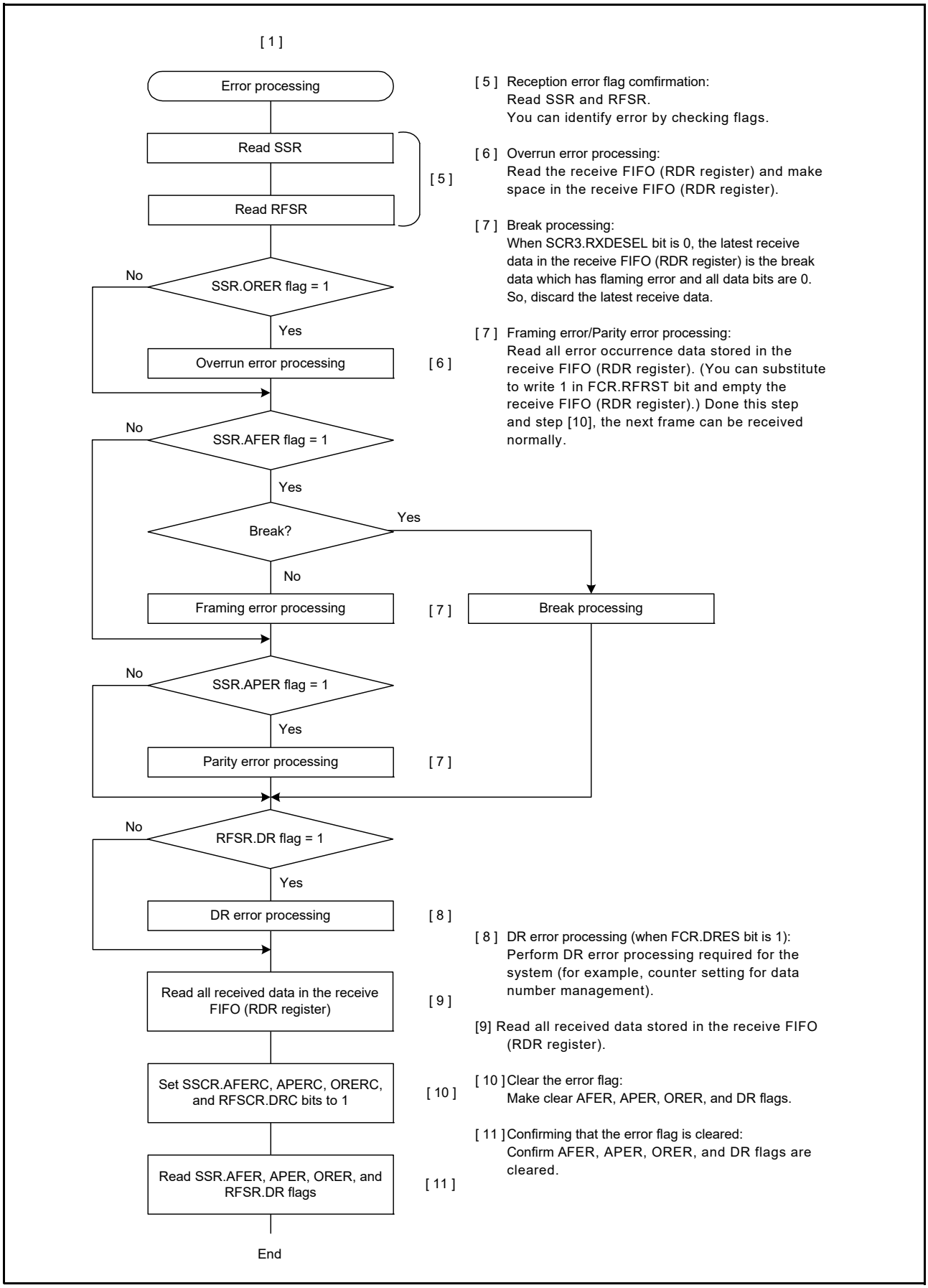


Figure 33.22 Example Flowchart of Serial Reception in Asynchronous Mode (FIFO Mode) (2)

33.3.10 Receive Data Sampling Timing Adjustment

When a waveform such that high width is different from low width because the difference between rising time and falling time is large is received, adjust the timing for data to be sampled at the center of the narrower pulse. When low width is narrower than high width, move the sampling timing forward, and when high width is narrower than low width, move the sampling timing backward.

When the SCR4.RTMG[3:0] bits are set to an offset to the default sampling point and then the SCR4.RTADJ bit is set to 1, received data is sampled at the specified position.

Figure 33.23 shows an example of the sampling timing adjustment.

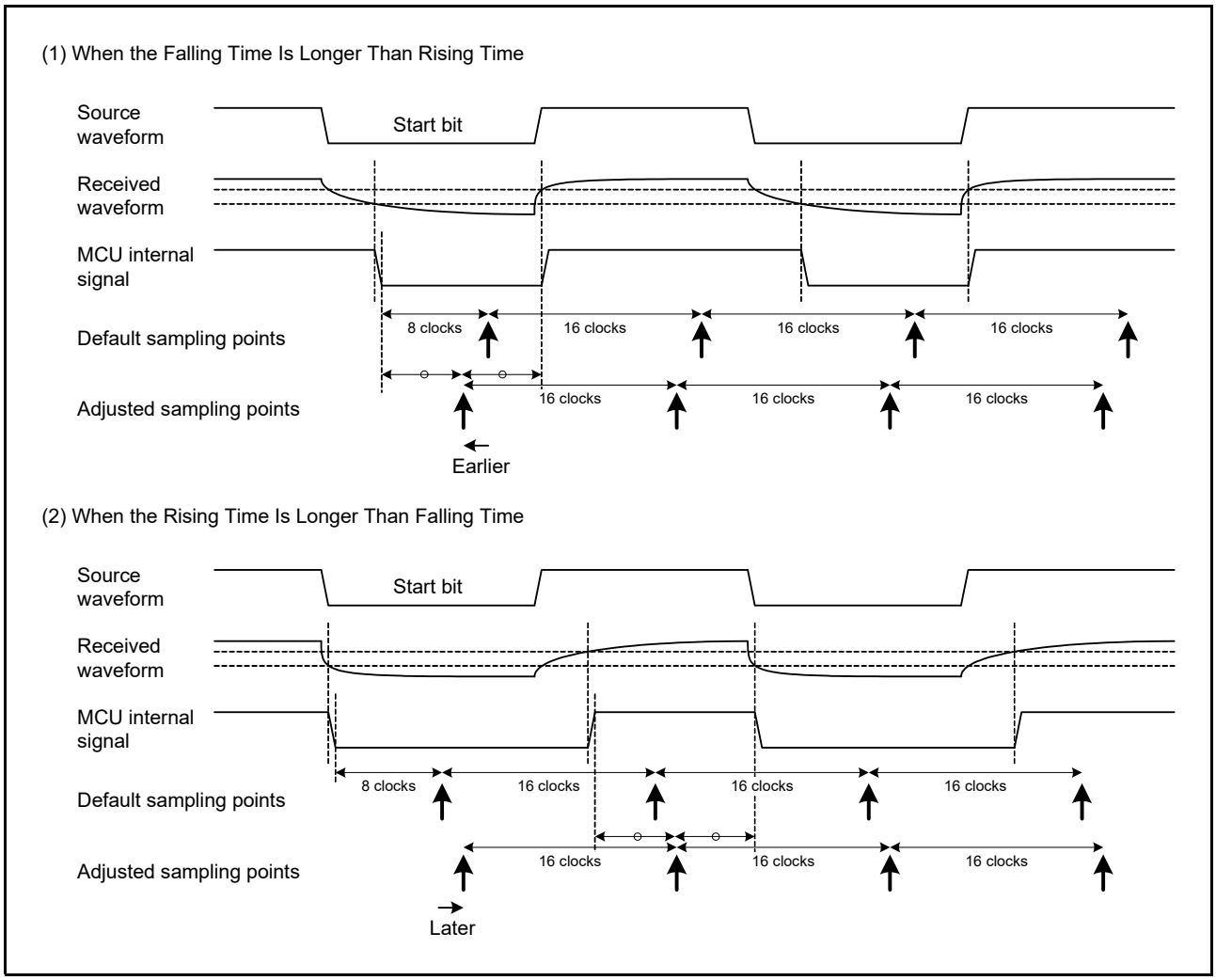


Figure 33.23 Example of Sampling Timing Adjustment (SCR2.ABCSE Bit = 0 and SCR2.ABCS Bit = 0)

33.3.11 Transmit Data Transition Timing Adjustment

When a difference between high width and low width for a waveform transmitted by this MCU arises at the receiver, it is also possible to make a difference between the high width and the low width beforehand at transmission to adjust so that the difference disappears at the receiver. When high width becomes narrower at the receiver, expand the high width of the transmit signal by delaying the falling edges, and when low width becomes narrower at the receiver, expand the low width of the transmit signal by delaying the rising edges.

When the SCR4.TTMG[3:0] bits are set to the transition direction and the delay amount and the SCR4.TTADJ bit is set to 1, transmit data changes at the specified point.

Figure 33.24 shows an example of the transition timing adjustment.

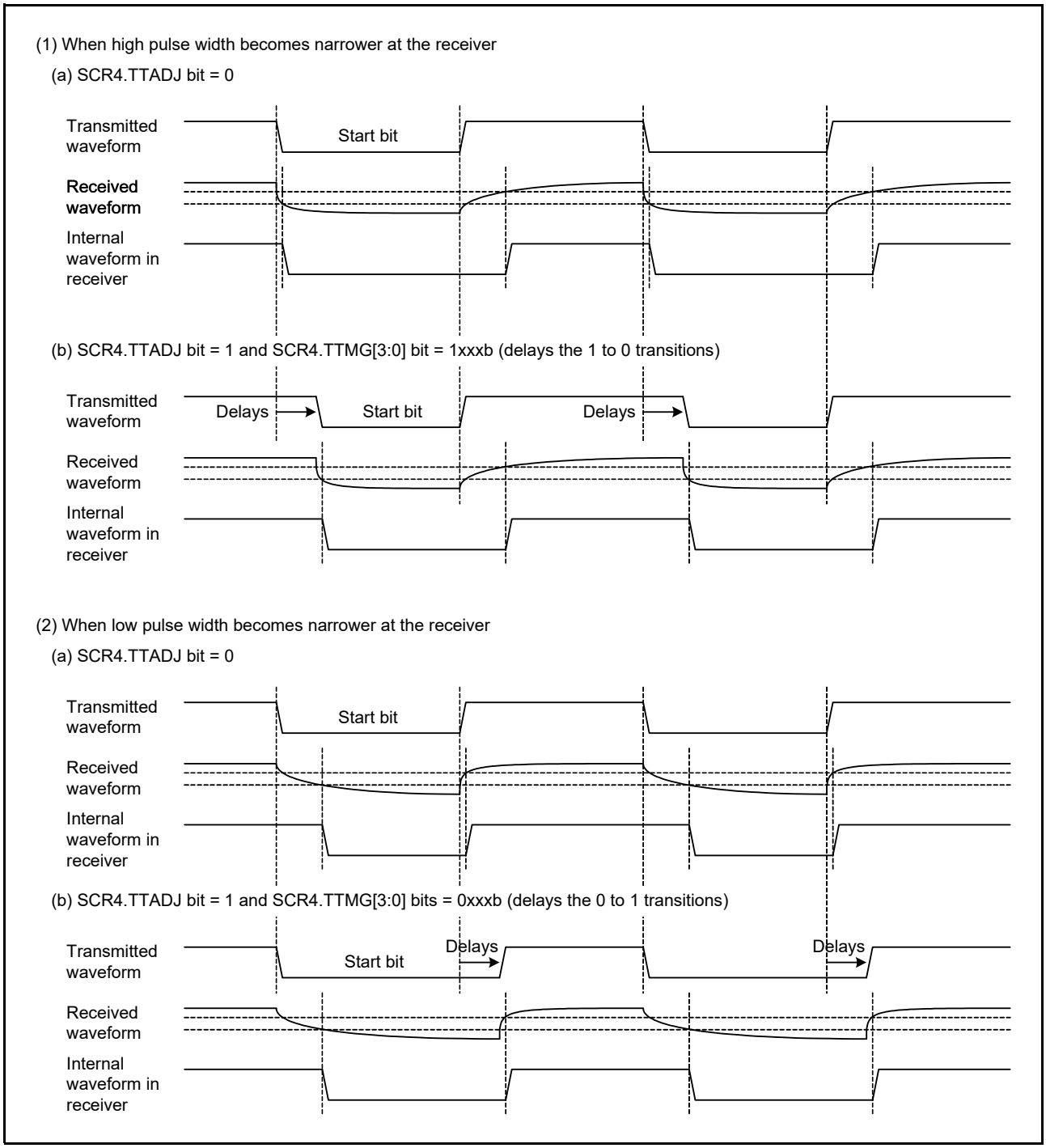


Figure 33.24 Example of Transition Timing Adjustment

33.4 Multi-Processor Communication Function

Using the multi-processor communication functions enables to transmit and receive data by sharing a communication line between multiple processors by using asynchronous serial communication in which the multi-processor bit is added. In multi-processor communication, a unique ID code is allocated to each receiving station. Serial communication cycles consist of an ID transmission cycle to specify the receiving station and a data transmission cycle to transmit data to the specified receiving station. The multi-processor bit is used to distinguish between the ID transmission cycle and the data transmission cycle. When the multi-processor bit is set to 1, it indicates the ID transmission cycle and when the multiprocessor bit is set to 0, it indicates the data transmission cycle. Figure 33.25 shows an example of communication between processors by using a multi-processor format. First, a transmitting station transmits communication data in which the multi-processor bit set to 1 is added to the ID code of the receiving station. Next, the transmitting station transmits the communication data in which the multi-processor bit set to 0 is added to the transmit data. Upon receiving the communication data in which the multi-processor bit is set to 1, the receiving station compares the received ID with the ID of the receiving station itself and if the two matches, receives the communication data that is subsequently transmitted. If the received ID does not match with the ID of the receiving station, the receiving station skips the communication data until again receiving the communication data in which the multi-processor bit is set to 1.
RTS control cannot be used at the time of multi-processor communication function use, because this is a function corresponding to one-to-many communications.

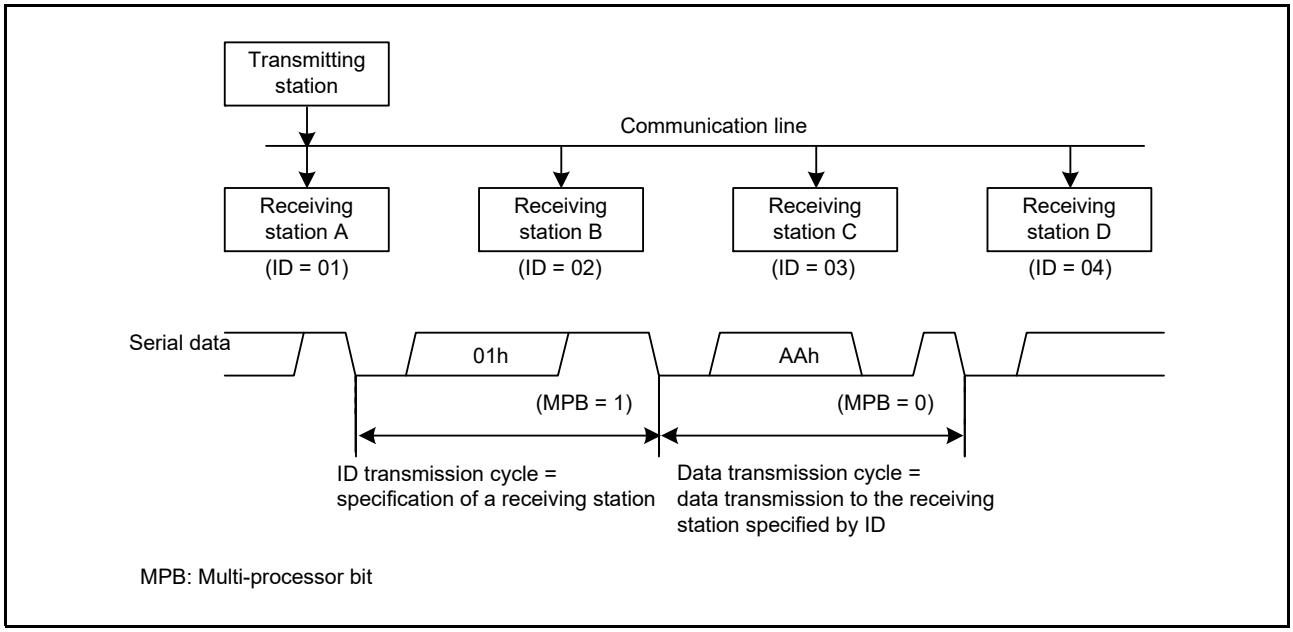


Figure 33.25 An Example of Communication using the Multi-Processor Format (Example of Transmission of Data AAh to Receiving Station A)

(1) Non-FIFO Mode

For supporting this function, the RSCI provides the MPIE bit in SCR0 register. When the MPIE bit is set to 1, transfer of receive data from the RSR register to the RDR register, detection of a receive error, and setting the respective status flags RDRF, ORER and AFER in SSR are disabled until reception of data in which the multi-processor bit is set to 1.
Upon receiving a reception character in which the multi-processor bit is set to 1, the MPB flag in RDR register is set to 1 and the MPIE bit in SCR0 register is automatically cleared, thus returning to a non-multi-processor reception operation. At this time, an RXI interrupt is generated if the RIE bit in SCR0 register is set.
When the multi-processor format is specified, specification of the parity bit is disabled.
Apart from this, there is no difference from the operation in the non-multi-processor asynchronous mode. A clock which is used for the multi-processor communication is also the same as the clock used in the non-multi-processor asynchronous mode.

(2) FIFO Mode

For transmission, SW should write to TDR.MPBT (Multi-Processor Bit Transfer) which corresponds to transmit data in TDR.TDAT[8:0] bits. For reception, multi-processor bit that is a part of receive data, is stored to RDR.MPB flag, and receive data is stored to RDR.RDAT[8:0] bits. When the MPIE bit is set to 1, transfer of receive data from the RSR to the RDR.RDAT[8:0] bits, detection of a receive error, and detection of DR flag, and setting the respective status flags RDRF, ORER, and AFER in SSR are disabled until reception of data in which the multi-processor bit is set to 1.

Upon receiving a reception character in which the multi-processor bit is set to 1, the MPB flag in RDR register is set to 1,and receive data is stored to receive FIFO (RDR.RDAT[8:0] bits), and the MPIE bit in SCR0 register is automatically cleared, thus returning to a non-multi-processor reception operation. At this time, an RXI interrupt is generated if the RIE bit in SCR0 register is set.

When the multi-processor format is specified, specification of the parity bit is disabled. Apart from this, there is no difference from the operation in the non-multi-processor asynchronous mode and FIFO mode.

33.4.1 Multi-Processor Serial Data Transmission

(1) Non-FIFO Mode

Figure 33.26 shows a sample flowchart of multi-processor data transmission. In the ID transmission cycle, the ID should be transmitted with the MPBT bit in TDR register set to 1. In the data transmission cycle, the data should be transmitted with the MPBT bit set to 0. The other operations are the same as the operations in asynchronous mode.

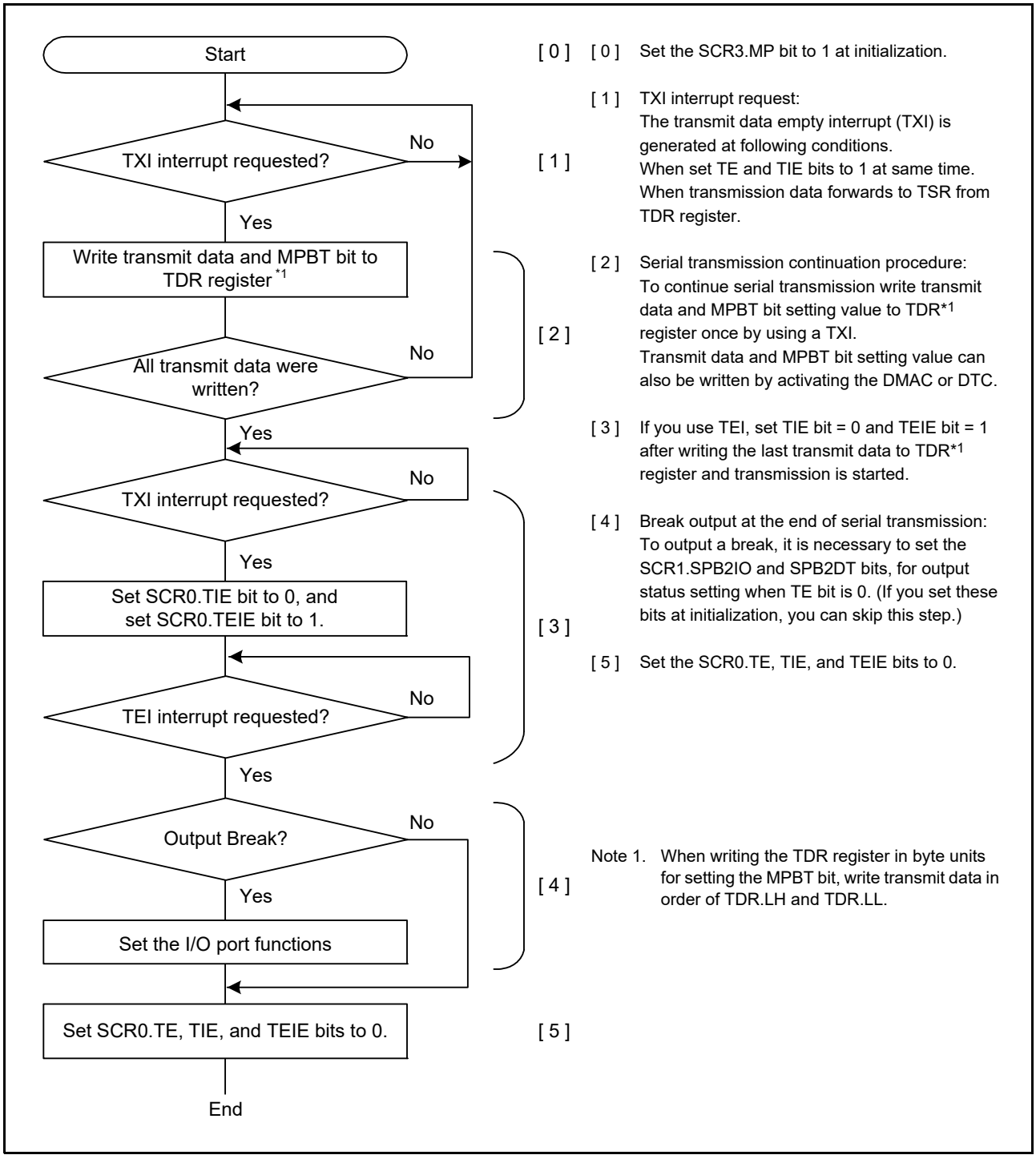


Figure 33.26 Example of Multi-Processor Serial Transmission Flowchart (Non-FIFO Mode)

(2) FIFO Mode

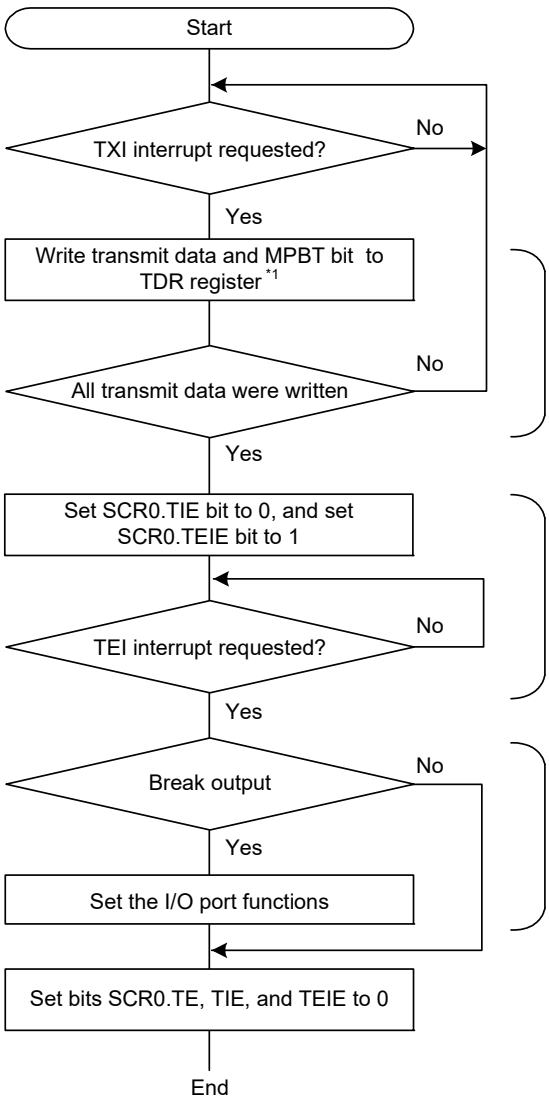
Table 33.33 shows an example of data format that is written to transmit FIFO (TDR register) in multi-processor mode. Write MPBT in bit9 of TDR. And write data to transmit FIFO (TDR register) corresponded to data length. It should write to 0 for unused bits.

Table 33.33 Data Format in Multi-processor Mode that is Written to Transmit FIFO (TDR Register)

Data Length	Register setting		Transmit Data in TDR.L															
	SCR3	CHR[1:0]	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
7 bits	1	1	—	—	—	—	—	—	MPBT	—	—	TDAT[6:0]						
8 bits	1	0	—	—	—	—	—	—	MPBT	—	TDAT[7:0]							
9 bits	0	0 or 1	—	—	—	—	—	—	MPBT	TDAT[8:0]								

—: Do not used. It should write to 0.

Figure 33.27 shows a sample flowchart for multi-processor data transmission in FIFO mode. In the ID transmission cycle, the ID should be transmitted with the MPBT bit in TDR register set to 1. In the data transmission cycle, the data should be transmitted with the MPBT bit set to 0. The other operations are the same as the operations in asynchronous mode with FIFO enabled.



[0] Set the SCR3.FM and MP bits to 1 at initialization.

[1] TXI interrupt request:
The transmit data empty interrupt (TXI) is generated when transmission data forwards to TSR from the transmit FIFO (TDR register) and the number of transmit data written in the transmit FIFO is equal to or less than the transmit FIFO threshold.

[2] Serial transmission continuation procedure:
To continue serial transmission, write all transmit data and MPBT bit setting value to the transmit FIFO (TDR register) by using a TXI interrupt. Transmit data and MPBT bit setting value can also be written to the transmit FIFO (TDR register) by activating the DMAC or DTC.

[4] If you use TEI, set TIE bit = 0 and TEIE bit = 1 after writing the last transmit data to TDR*1 register.

[5] Break output at the end of serial transmission:
To output a break, it is necessary to set the SCR1.SPB2IO and SPB2DT bits, for output status setting when TE bit is 0.
(If you set these bits at initialization, you can skip this step.)

[6] Set bits SCR0.TE, TIE, and TEIE to 0.

Note 1. When writing the TDR register in byte units for setting the MPBT bit, write transmit data in order of TDR.LH and TDR.LL.

Figure 33.27 Example of Multi-Processor Serial Transmission Flowchart (FIFO Mode)

33.4.2 Multi-Processor Serial Data Reception

(1) Non-FIFO Mode

Figure 33.29 and Figure 33.30 are sample flowcharts of multi-processor data reception. When the MPIE bit in SCR0 register is set to 1, reading the communication data is skipped until reception of the communication data in which the multi-processor bit is set to 1. When the communication data in which the multi-processor bit is set to 1 is received, the received data is transferred to RDR register. At this time, the RXI interrupt request is generated. The other operations are the same as the operations in asynchronous mode.
Figure 33.28 is the example of operation for reception.

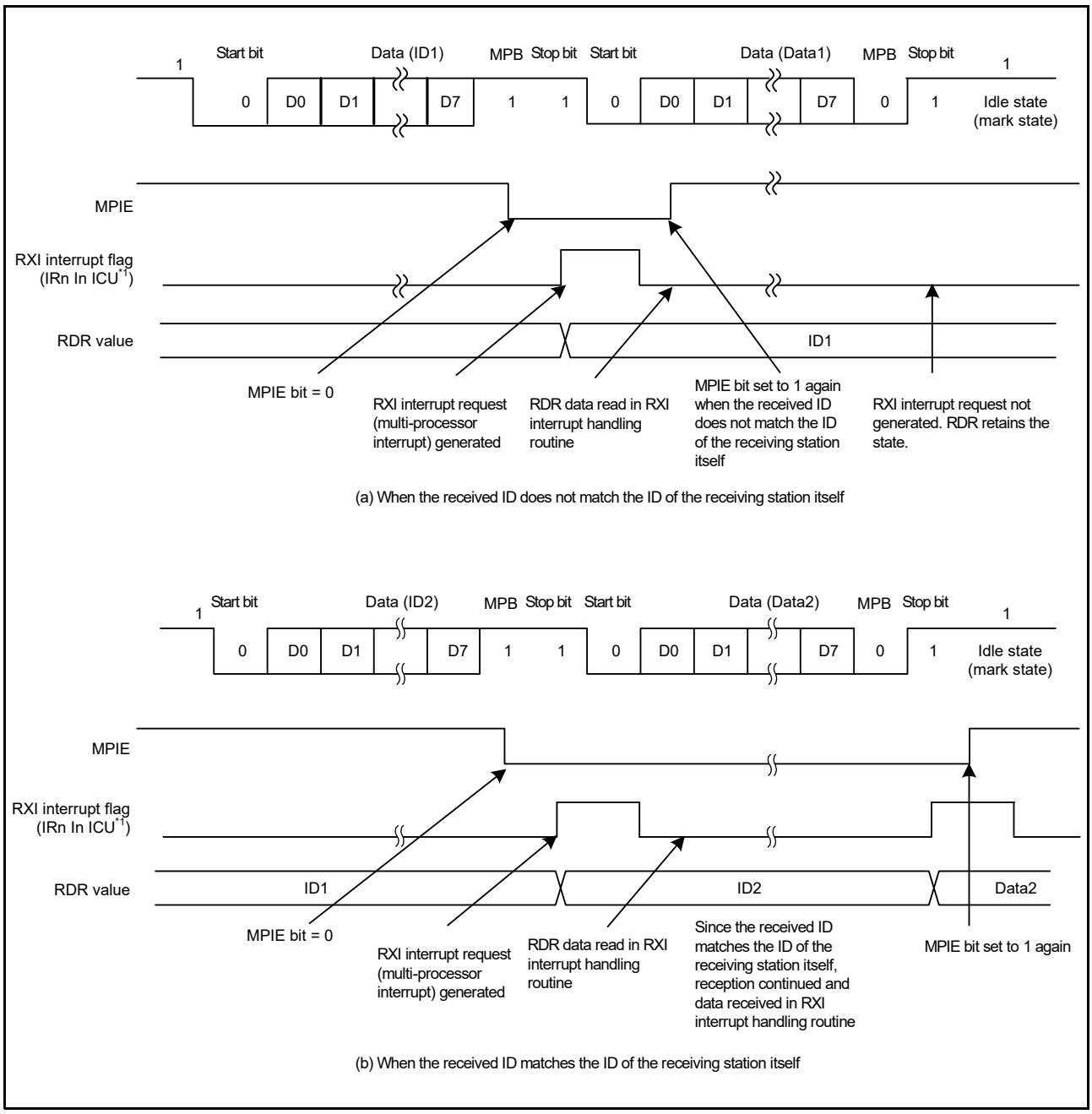
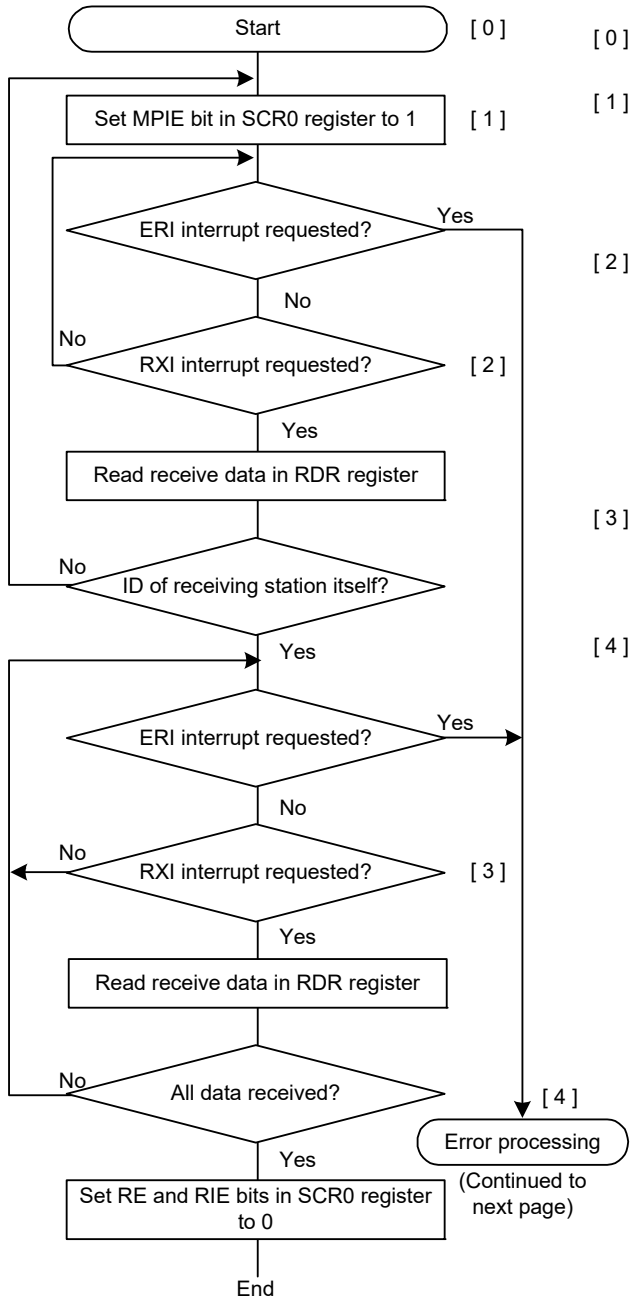


Figure 33.28 Example of RSCI Reception (8-Bit Data/Multi-Processor Bit/1 Stop Bit)



[0] Set the SCR3.MP bit to 1 at initialization.

[1] ID reception cycle:
Set SCR0.MPIE bit to 1 and wait for ID reception.
If you set this bits at initialization, you can skip this Step.

[2] RSCI status confirmation and reception and comparison of ID:
Read data in RDR register at the first RXI interrupt and compare it with the ID of the receiving station itself. If the ID does not match the ID of the receiving station itself, set the MPIE bit to 1 again, and wait for another RXI interrupt request.

[3] Data reception at an RXI interrupt:
Read data in RDR register once in the RXI interrupt routine.

[4] Receive error processing and break detection:
If a receive error occurs, an error is identified by reading ORER and AFER flags in SSR. After performing the appropriate error processing be sure to clear the ORER and AFER flags. Reception cannot be resumed if any of these flags is set to 1.
In the case of a framing error, a break can be detected by reading SSR.RXDMON flag.

Figure 33.29 Example of Multi-Processor Serial Reception Flowchart (1) (Non-FIFO Mode)

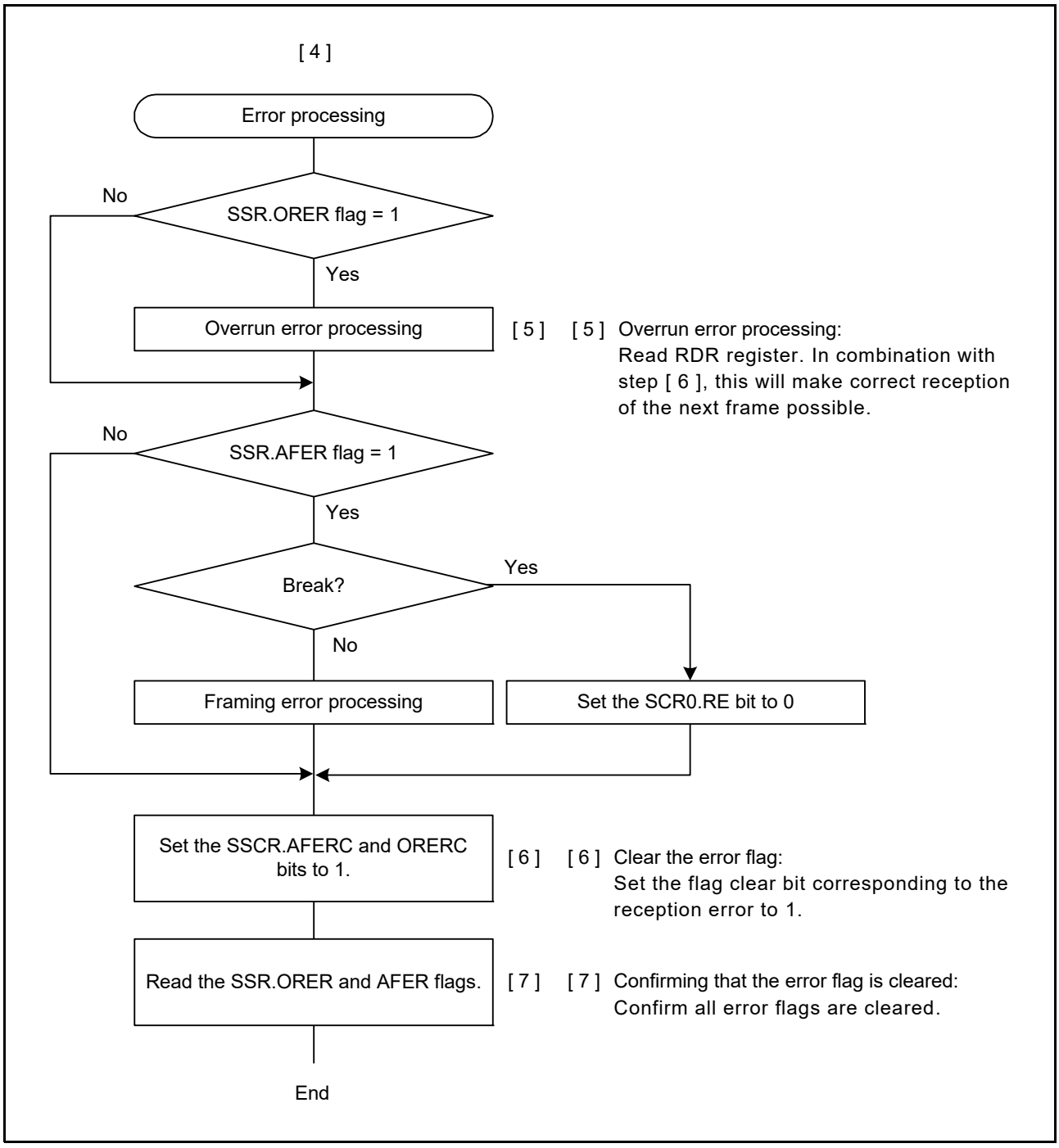


Figure 33.30 Example of Multi-Processor Serial Reception Flowchart (2) (Non-FIFO Mode)

(2) FIFO Mode

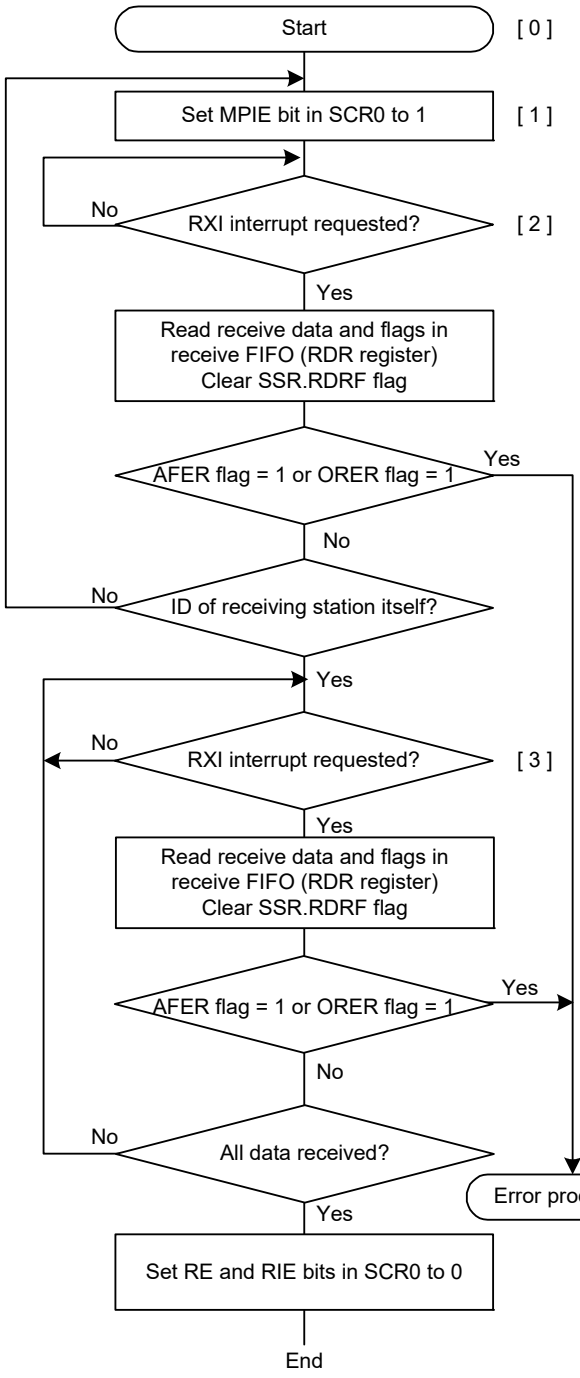
Table 33.34 shows an example of data format that is stored to receive FIFO (RDR register) in multi-processor mode. MPB flag is stored in bit9 of RDR register. 0 is stored to APER and PER flags. Data is stored to receive FIFO RDAT[8:0] bits corresponded to data length. It is stored to 0 for unused bits. Reading the receive FIFO (RDR register) updates the FER, PER, MPB flags and receive data (RDAT[8:0] bits) in the receive FIFO (RDR register) with the next received data. The AFER, APER, and ORER flags in the receive FIFO (RDR register) always reflect the status of the corresponding flags in the SSR and RFSR registers.

Table 33.34 Data Format in Multi-Processor Mode that is Stored to Receive FIFO (FIFO Mode)

Data Length	Register setting		Received Data in RDR Register															
	SCR3	CHR[1:0]	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
7 bits	1	1	0	0	0	FER	PER	DR	MPB	0	0	RDAT[6:0]						
8 bits	1	0	0	0	0	FER	PER	DR	MPB	0	RDAT[7:0]							
9 bits	0	0 or 1	0	0	0	FER	PER	DR	MPB	RDAT[8:0]								
Data Length	**SCR3**	**CHR[1:0]**	**b31**	**b30**	**b29**	**b28**	**b27**	**b26**	**b25**	**b24**	**b23**	**b22**	**b21**	**b20**	**b19**	**b18**	**b17**	**b16**
7 bits	1	1	0	0	0	AFER	APER	0	0	ORER	0	0	0	0	0	0	0	0
8 bits	1	0	0	0	0	AFER	APER	0	0	ORER	0	0	0	0	0	0	0	0
9 bits	0	0 or 1	0	0	0	AFER	APER	0	0	ORER	0	0	0	0	0	0	0	0

Note: When data length is 7bit, it can read always 0 in RDAT[8:7] bits.
When data length is 8bit, it can read always 0 in RDAT[8] bit.

Figure 33.31 shows a sample flowchart for multi-processor data reception in FIFO mode. When the MPIE bit in SCR0 register is set to 1, reading the communication data is skipped until reception of the communication data in which the multiprocessor bit is set to 1. When the communication data in which the multi-processor bit is set to 1 is received, the received data, MPB flag and each errors are transferred to receive FIFO (RDR register), and the MPIE bit in SCR0 register is automatically cleared, thus returning to a normal reception operation. After a framing error occurred and SSR.AFER flag is set to 1, but RSCI continues data reception.
The other operations are the same as the operations in asynchronous mode with FIFO enabled.



[0] Set the SCR3.FM and MP bits to 1.
Also, set the FCR.RTRG[4:0] bits to 00001b and DRES bit to 0 to generate an RXI interrupt request when data with the MPB flag set to 1 is received.

[1] ID reception cycle:
Set SCR0.MPIE bit to 1. And wait for ID reception. If you set this bits at initialization, you can skip this Step.

[2] RSCI status confirmation and reception and comparison of ID:
RSCI skips the received data until it receives the data with the MPB flag set to 1. When receiving data with the MPB flag set to 1, RSCI generates an RXI interrupt request.
In the RXI interrupt routine, read the data from the receive FIFO (RDR register) and compare it with the ID of the receiving station itself. If the ID does not match the ID of the receiving station itself, set the MPIE bit to 1 again and wait for another RXI interrupt request. If it matches the ID of the receiving station itself, set the receive FIFO threshold to the desired value.

[3] Data reception at an RXI interrupt:
Read data in the receive FIFO (RDR register) once in the RXI interrupt routine.

[4] Receive error processing and break detection:
If a receive error occurs, an error is identified by reading ORER and AFER flags in SSR register (There is no problem reading the SSR.ORER and AFER flags again.). After performing the appropriate error processing be sure to clear the ORER and AFER flags. In the case of a framing error, a break can be detected by reading the SSR.RXDMON flag.

Figure 33.31 Example Flowchart of Serial Reception in Multi-Processor Mode (1) (FIFO Mode)

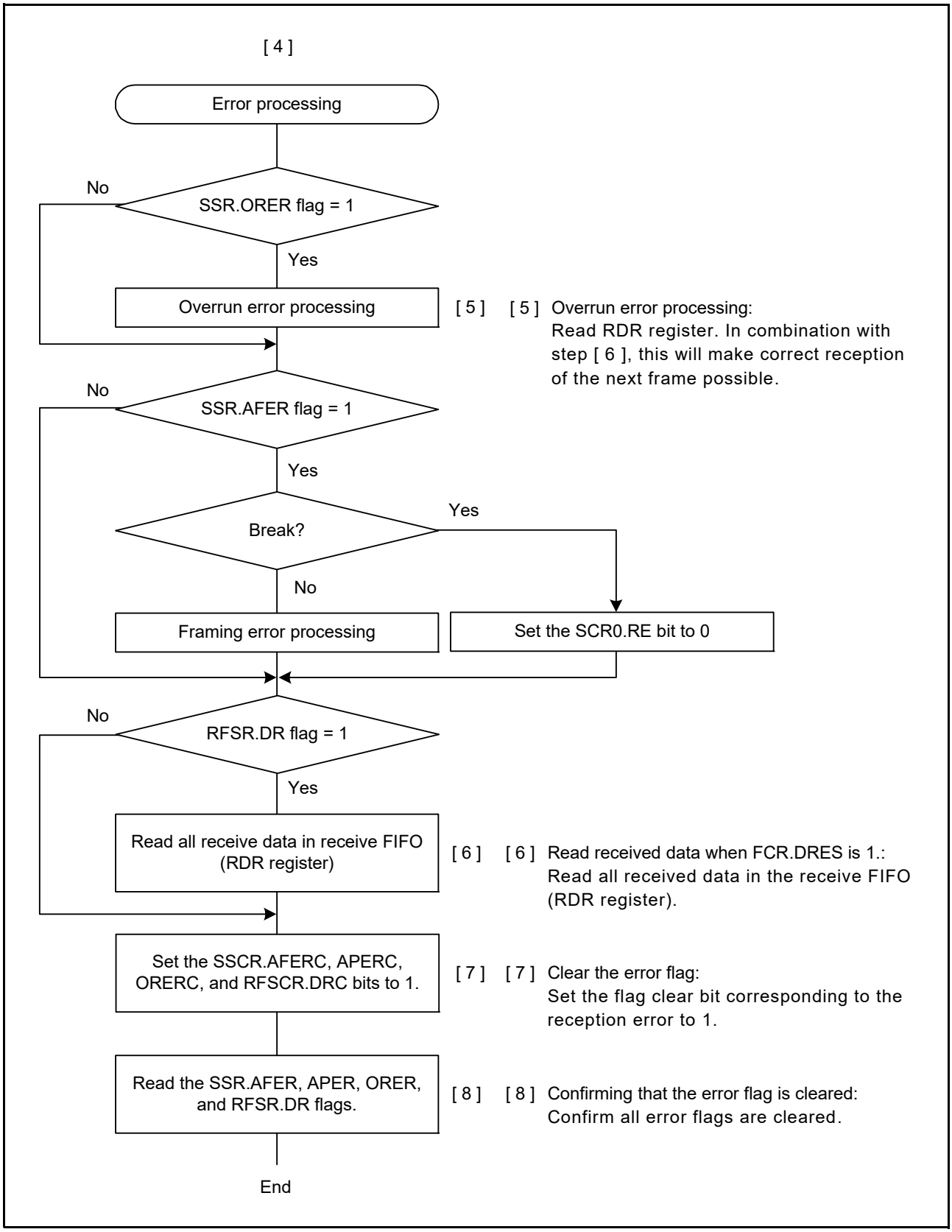


Figure 33.32 Example Flowchart of Serial Reception in Multi-Processor Mode (2) (FIFO Mode)

33.5 Manchester Mode

In manchester mode, the transmit or receive serial data is coded in Manchester encoding.
Figure 33.33 shows the conceptual image of Manchester encoding.

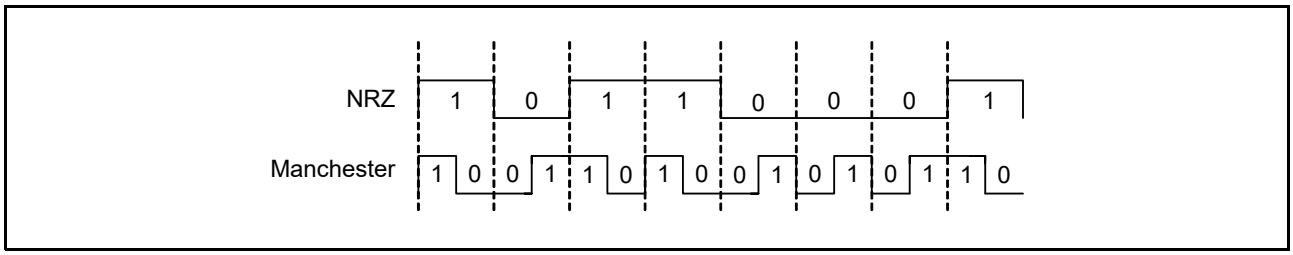


Figure 33.33 Example of Manchester Encoding

In manchester mode, a preface and a start bit area are added to the transmit data in the register to configure a transmit frame. For transmission, data is encoded in Manchester encoding. When data is received, frames having the same format as transmitted frames are detected and Manchester decoding is performed.
For details on the frame format, see section 33.5.1, Frame Format.

33.5.1 Frame Format

Figure 33.34 shows the frame format in manchester mode.
In the upper half of the figure, relevant setting registers are shown.
The preface area and the data area are encoded in Manchester encoding.

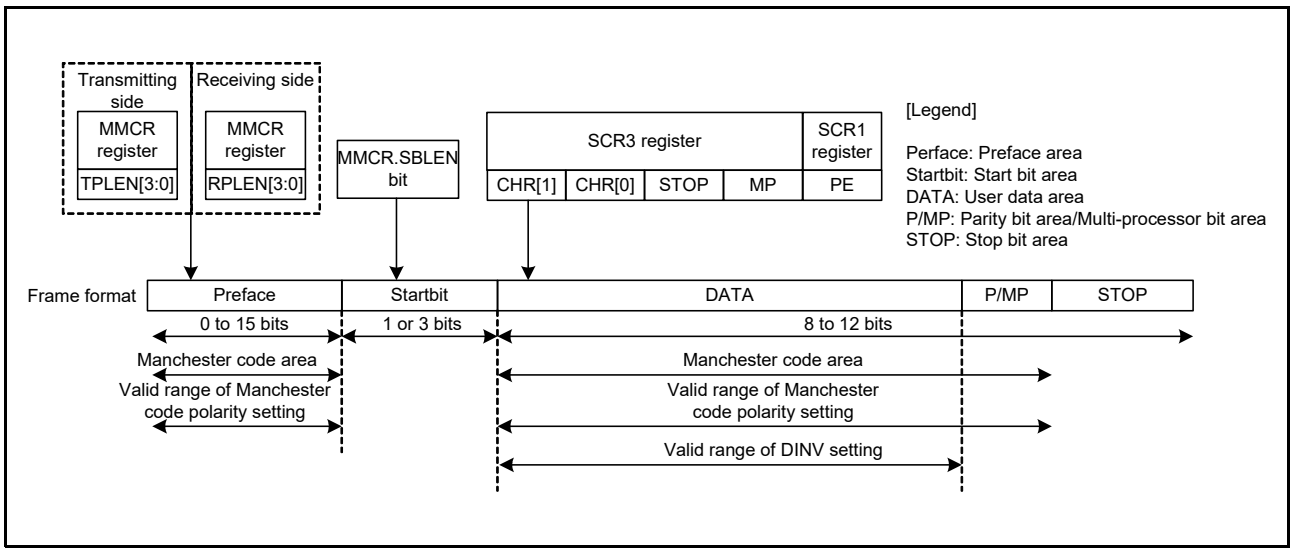


Figure 33.34 Frame Format in Manchester Mode

(1) Preface Area

This is a fixed pattern area located at the beginning of each frame.
Different registers are used to set the preface area for transmission and reception. The preface length is determined by setting MMCR.TPLEN[3:0] bits for transmission. It is determined by setting MMCR.RPLEN[3:0] bits for reception.
If it is set to 0, the transmit preface is disabled and is not added.
If it is set to 1d to 15d, a preface whose length is determined by this setting is added.
(For example, if it is set to 1d, a 1-bit preface is added. If it is set to 15d, a 15-bit preface is added.)
In addition, the preface pattern can be changed by setting, and it can be selected from four types of patterns by setting

MMCR.TPPAT[1:0] bits for transmission and MMCR.RPPAT[1:0] bits for reception.
Figure 33.35 shows how the preface pattern is set. The preface area and the start bit area are added for each communication frame.

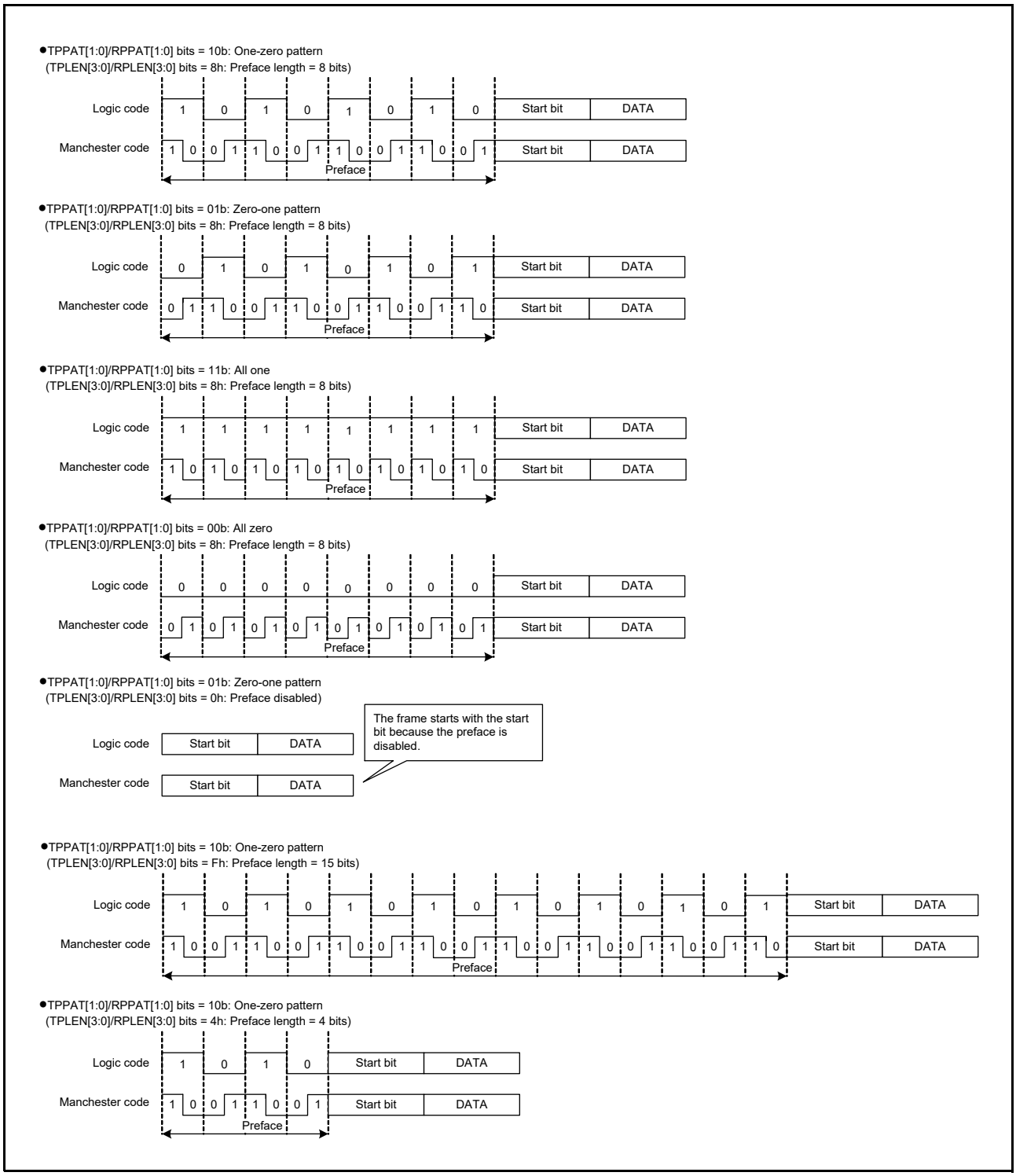
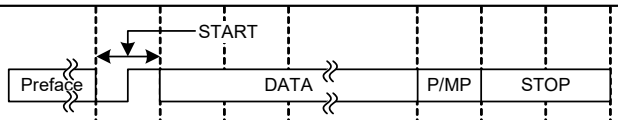
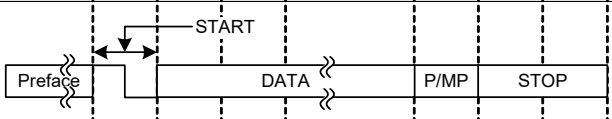
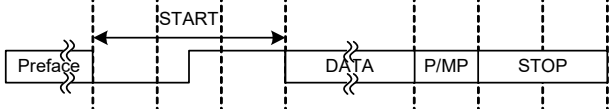
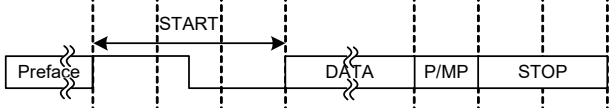
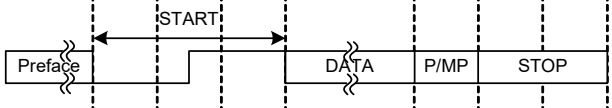
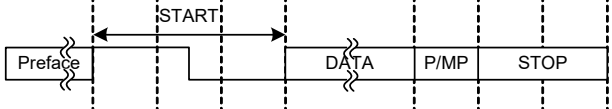


Figure 33.35 Preface Pattern Setting Example

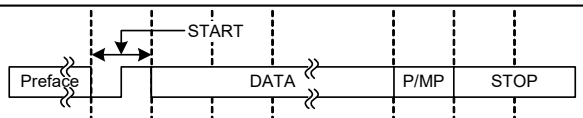
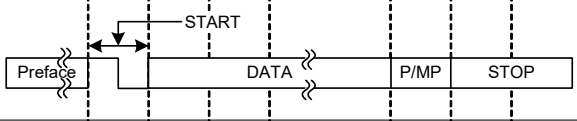
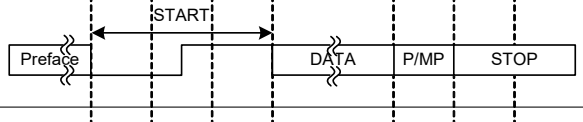
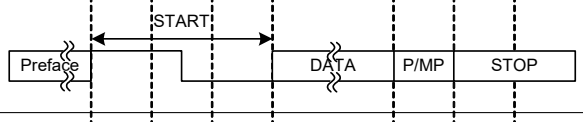
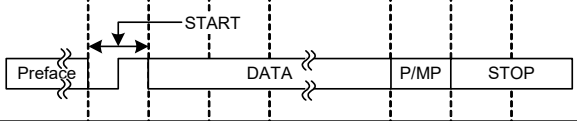
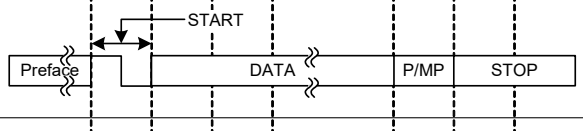
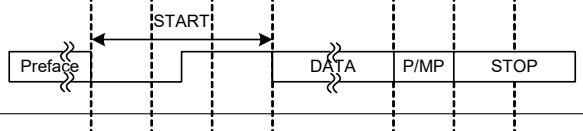
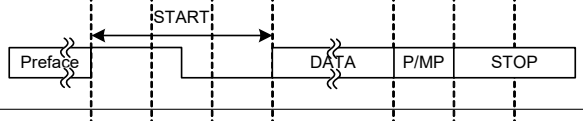
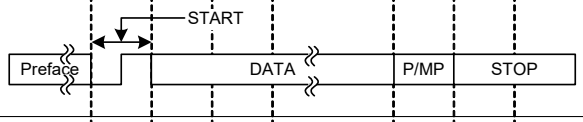
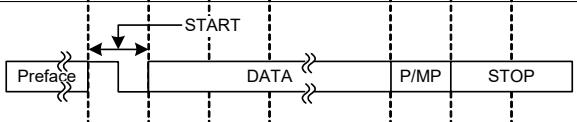
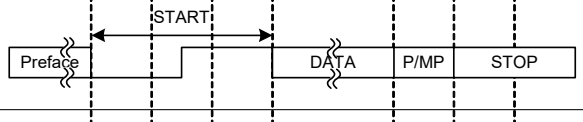
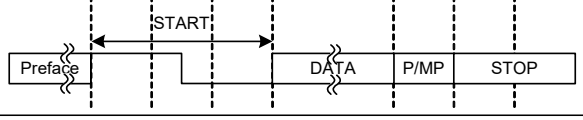
(2) Start Bit Area

This is an area indicating the start of valid data in a frame. It is added after the preface area.
The start bit length is determined by MMCR.SBLEN bit setting. When MMCR.SBLEN bit = 0, the start bit length is 1 bit. When MMCR.SBLEN bit = 1, the start bit length is 3 bits.
When MMCR.SBLEN bit = 1, the Sync type can be selected from command Sync and data Sync.
Command Sync means the three start bits are added as a one-to-zero transition.
Data Sync means the three start bits are added as a zero-to-one transition.
The Sync type is determined by the MMCR.SYNCE, MMCR.SBPTN and TDR.SYNC bits settings. (When receiving, the received result is applied to MMSR.RSYNC bit.)
When MMCR.SBLEN bit = 0, the start bit is added as a zero-to-one or one-to-zero transition. The selection is determined by the MMCR.SBPTN setting.
The MMCR.SYNCE bit specifies the destination to be referred to when setting for transmission. When the MMCR.SYNCE bit is set to 1, the MMCR.SBPTN setting is referred to. When the MMCR.SYNCE bit is set to 0, the TDR.SYNC bit setting is referred to.
Figure 33.36 shows the state of the start bit area according to the settings in the MMCR.SYNCE, MMCR.SBPTN and TDR.SYNC bits in the case of transmission. Figure 33.37 shows that in the case of reception.
The start bit(s) is not affected by the MMCR.ENCS bit or MMCR.DECS bit setting.

Register Setting				Output Signal	Output Start Bit
MMCR			TDR	Output TXD Waveform	
SBLEN	SYNCE	SBPTN	SYNC		
0	d.c.	0	d.c.		1bit 0 to 1 transition
0	d.c.	1	d.c.		1bit 1 to 0 transition
1	0	0	d.c.		3bit 0 to 1 transition
1	0	1	d.c.		3bit 1 to 0 transition
1	1	d.c.	0		3bit 0 to 1 transition
1	1	d.c.	1		3bit 0 to 1 transition

d.c.: don't care
Preface: Preface area
START: Start bit area
DATA: User data area
P: Parity bit area
MP: Multi-processor bit area
STOP: Stop bit area

Figure 33.36 Settings Related to and Format of the Start Bit Area at Transmission

Register Setting				Input Signal	Start Bit Detection Result *1	Register Indication
MMCR			TDR	RXD Input Waveform		MMSR.RSYNC
SBLEN	SYNCE	SBPTN	SYNC			
0	d.c.	0	d.c.		Normal start bit (1 bit: 0 to 1 transition)	0
					Start bit error	0
					Start bit error	0
					Start bit error	0
0	d.c.	1	d.c.		Start bit error	0
					Normal start bit (1 bit: 1 to 0 transition)	0
					Start bit error	0
					Start bit error	0
1	d.c.	d.c.	d.c.		Start bit error	0
					Start bit error	0
					DATA Sync	0
					COMMAND Sync	1

d.c.: don't care
Preface: Preface area
START: Start bit area
DATA: User data area
P: Parity bit area
MP: Multi-processor bit area
STOP: Stop bit area

Note 1. Data other than the start bit is assumed to be normal.

Figure 33.37 Settings Related to and Judgment of the Start Bit Area at Reception

(3) Data

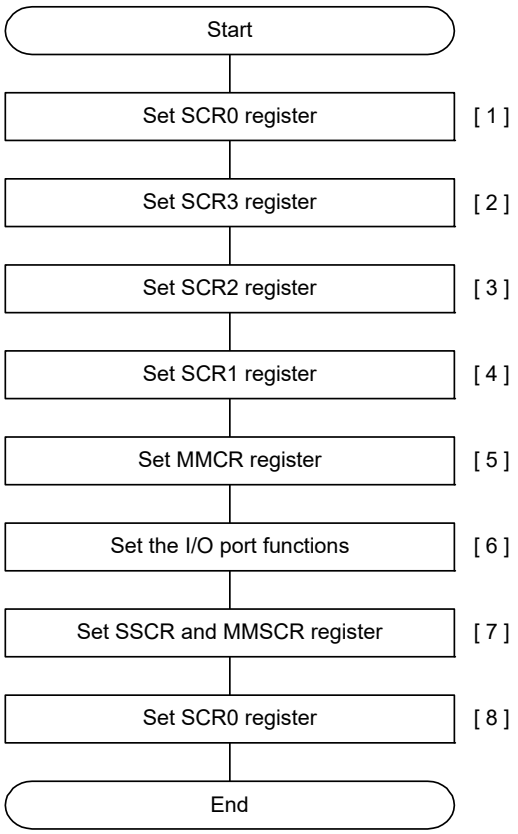
Since the format of the data area is the same as that of the asynchronous mode, see section 33.3.1, Serial Data Transfer Format.
As shown in Figure 33.34, Frame Format in Manchester Mode, the stop bit is not included in the manchester encoding range.

33.5.2 Clock

As the transfer clock in manchester mode, the clock generated by the on-chip baud rate generator is used by setting the SCR2.CKS[1:0] bits.
Also it is possible to set the oversampling (transfer rate of one bit period) by SCR2.ABCS bit. When the SCR2.ABCS bit is set to 0, oversampling ×16 is selected with the one-bit period being 16 cycles of the base clock. When the SCR2.ABCS bit is set to 1, oversampling ×8 is selected with the one-bit period being 8 cycles of the base clock.

33.5.3 RSCI Initialization of Manchester Mode

Before transferring data, write 0 to SCR0.TE and SCR0.RE bits (or write the initial value to SCR0 register), and initialize the RSCI following the example of flowchart shown in Figure 33.38.
Be sure to write 0 to SCR0.TE and SCR0.RE bits before changing the operation mode or communication format.
Note that setting the SCR0.RE bit to 0 initializes none of the SSR.ORER, AFER, APER, RDRF, MMSR.MCER, SYER, PFER and SBER flags, and the RDR registers.
Note also that switching the value of SCR0.TE bit from 0 to 1 when SCR0.TIE bit is 1 generates a TXI interrupt request.



[1] Set the following bits of SCR0 register to 0.
TEIE, TIE, RIE, TE, RE
If you have not changed from the initial settings, you can skip this step.

[2] Set the driver control, multi-processor communication, the communication mode, transmission/reception format, and clock enable.
Leave unused bits such as CPOL and CPHA bits at their initial values.

[3] Set the clock selection, bit rate setting.
Leave unused function bits at their initial values.

[4] Set the noise filter, loopback control, communication terminal status, parity check, and CTS/RTS function.

[5] Set manchester transmission/reception setting.
Start bit setting, retiming setting, manchester polarity setting, transmission preface setting, reception preface setting.

[6] Set the I/O port so that the required pin function is valid among the TXDn, RXDn, and SCKn pins.

[7] Set 1 to the following clear bits in the SSCR and MMSCR registers to clear the target flag.
SSCR.RDRFC, AFERC, APERC, MFFC, ORERC, DFERC, DPERC, DCMFC, ERSC
MMSCR.MCERC, SBERC, SYERC, PFERC
This procedure can be omitted for the initial setting after reset.

[8] Set the SCR0.TE or RE bit to 1. To enable interrupts, set the TE bit and TIE bit, and the RE bit and RIE bit to 1 with one instruction at the same time. Set the multiprocessor function. The TXDn and RXDn pins can be used by setting the TE and RE bits.

Figure 33.38 RSCI Initialization Flow in Manchester Mode

33.5.4 Double Speed Operation

When the ABCS bit in SCR2 register is set to 1 and eight pulses of the base clock for a 1 bit period is selected, the RSCI operates on the bit rate twice that of when ABCS is set to 0.
When the BGDM bit in SCR2 register is set to 1, the cycle of the base clock is reduced to half and the RSCI operates on the bit rate twice that of when BGDM is set to 0.
When the ABCS bit in SCR2 register and the BGDM bit in SCR2 register are set to 1, the RSCI operates on the bit rate four times that of when the ABCS bit in SCR2 register and the BGDM bit in SCR2 register are set to 0.

33.5.5 CTS, RTS Functions

The CTS function controls transmission using the CTSn # pin input. Setting the CTSE bit in SCR1 register to 1 enables the CTS function. The CTSn#/RTSn# pin can be set as a multiplexed pin which allows one pin to be used for either function, or as dedicated pins with each pin for a single function. Use the CRSEP bit in SCR1 register for this setting.
When the CTS function is enabled, transmission starts only when the CTSn# pin is at the low level.
Even if the level of CTSn# pin goes High after transmission starts, does not affect transmission of the current frame, which continues.
The RTS function uses output on the RTSn# pin to request transmission. When the RSCI is ready to receive, it outputs a low level to the RTSn# pin, Conditions for output of the low level and high level are as follows:
[Conditions for low-level output]
When all conditions listed below are satisfied:

- The value of the SCR0.RE bit is 1.
- There are no received data yet to be read and reception is not in progress.
- All of the following flags are set to 0:
  SSR.ORER, AFER, APER and MMSR.MCER, SBER (when SBERIE bit = 1), SYER (when SYERIE bit = 1), PFER (when PFERIE bit = 1)

[Conditions for high-level output]
When the conditions for low output are not satisfied.

33.5.6 Serial Data Transmission in Manchester Mode

The RSCI encodes data in Manchester encoding and sends the resultant data in manchester mode.
When the polarity setting (MMCR.ENCS bit) set to 0, logic 0 is coded as a zero-to-one transition in Manchester code and logic 1 is coded as a one-to-zero transition in Manchester code.
When the polarity setting (MMCR.ENCS bit) set to 1, logic 0 is coded as a one-to-zero transition in Manchester code and logic 1 is coded as a zero-to-one transition in Manchester code.
For this reason, a level transition occurs with the Manchester encoded data in the middle of individual logic data. (See Figure 33.33)
The transmitter constructs transmit frames in a specific format by adding a preface area to data and setting the start bit(s) according to the polarity setting and sends resultant serial data.
For details on the frame format, see section 33.5.1, Frame Format.
Figure 33.39 shows the flowchart in transmission. At transmission starts, set the SCR0.TIE and SCR0.TE bits to 1 simultaneously with one instruction. Then, a TXI interrupt request is generated.
Figure 33.40 to Figure 33.42 show examples of the operation for serial transmission in manchester mode.

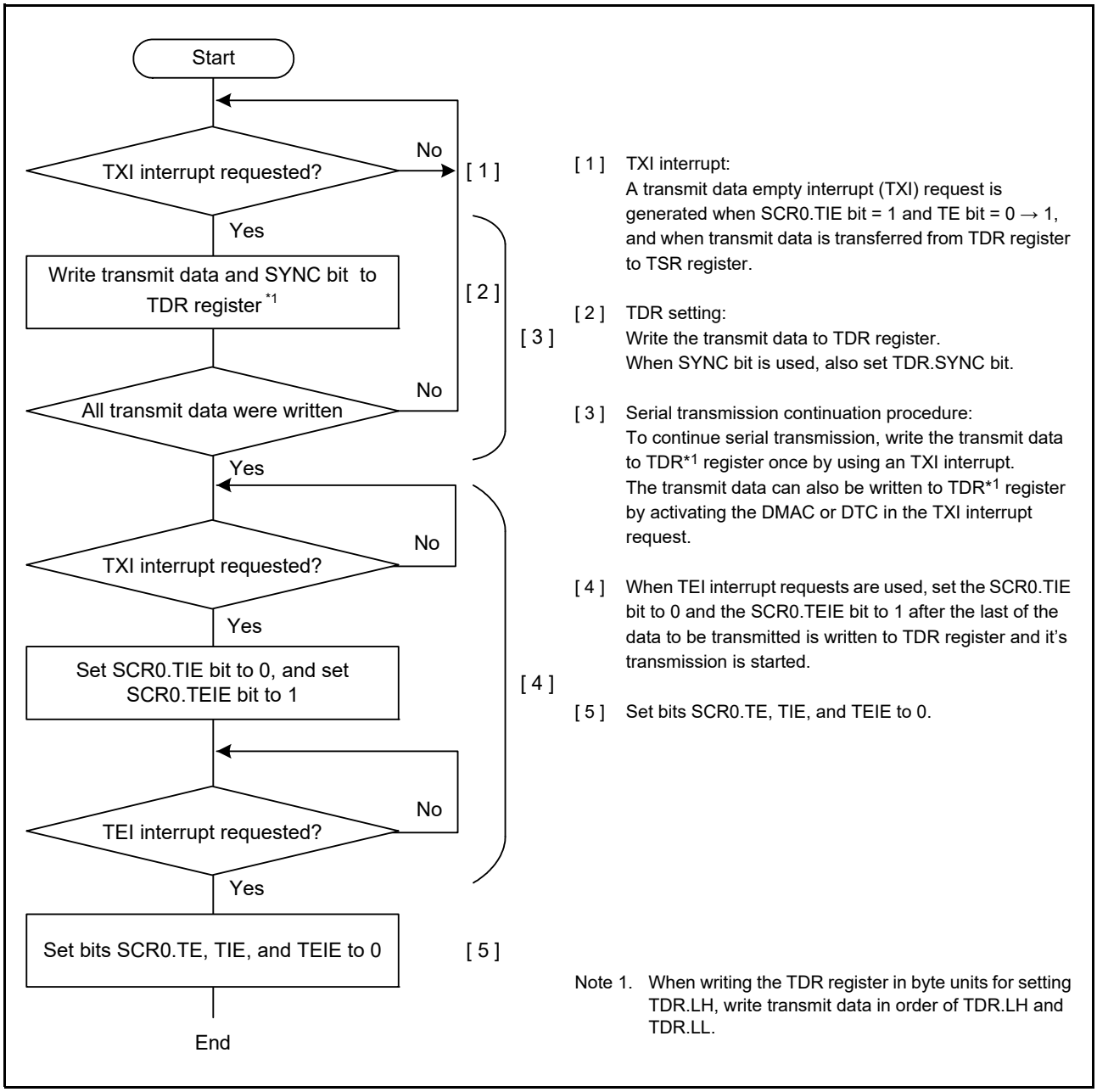
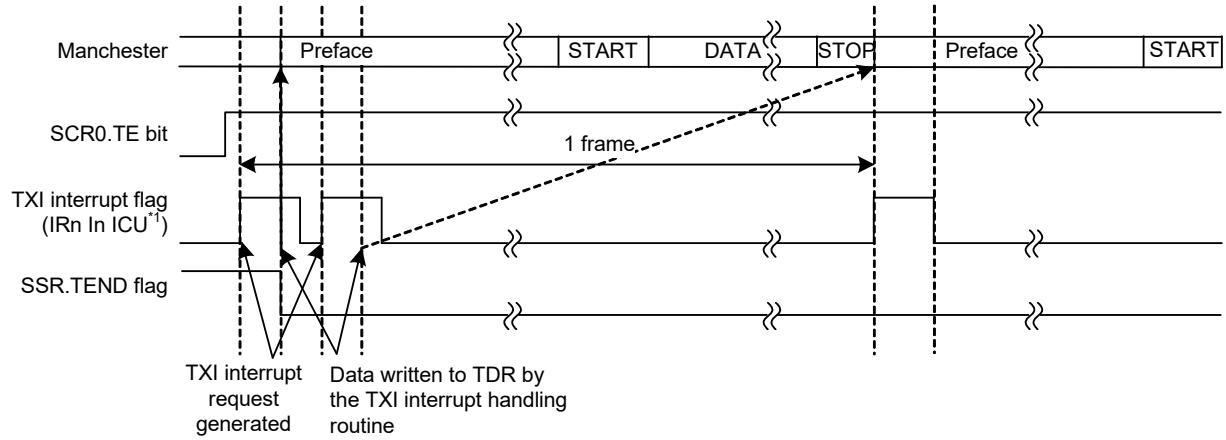
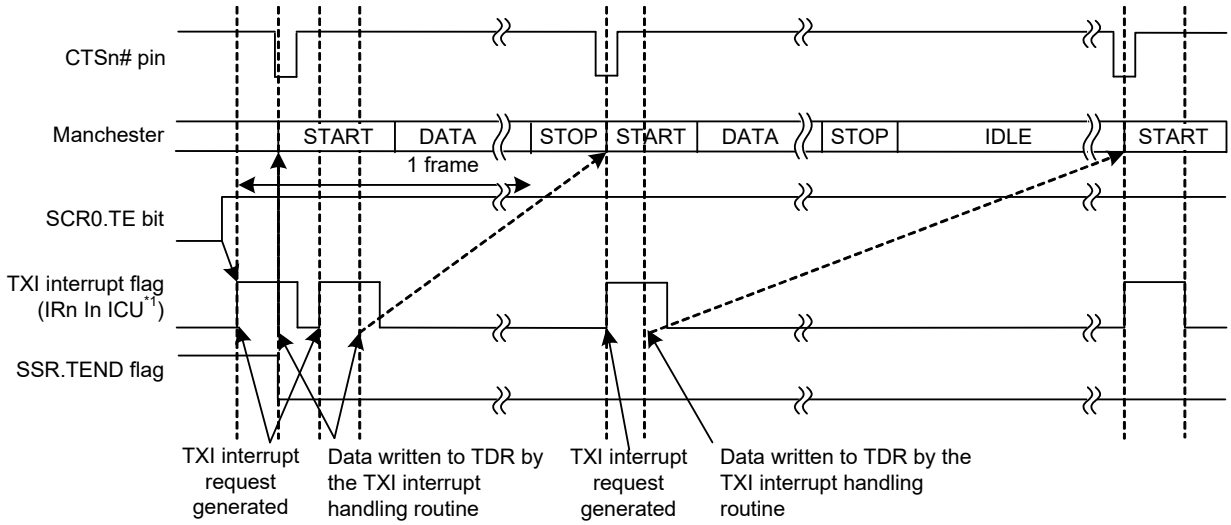


Figure 33.39 Serial Transmission Flowchart in Manchester Mode



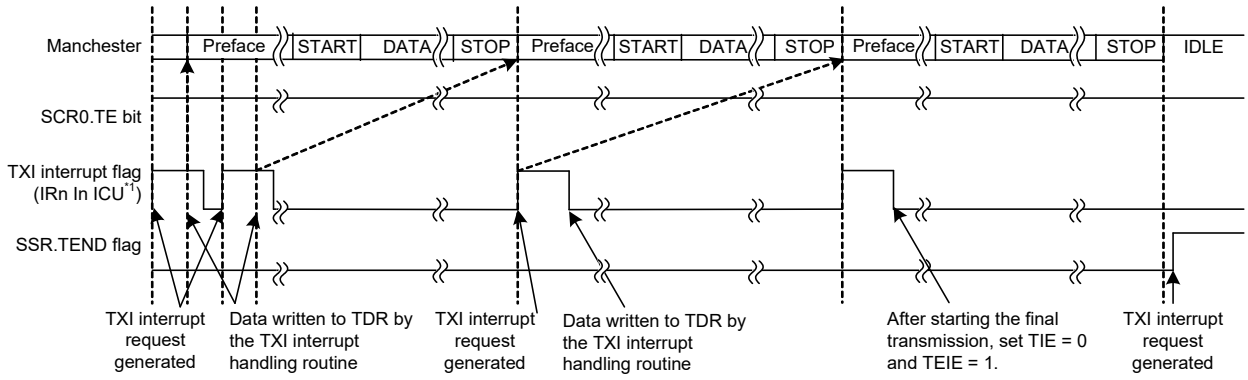
Note 1. Refer to section 14, Interrupt Controller (ICUG) for details on the corresponding interrupt vector number.

Figure 33.40 Example of Start-of-Transmission Operation for Serial Transmission in Manchester Mode (with Preface but without the CTS Function)



Note 1. Refer to section 14, Interrupt Controller (ICUG) for details on the corresponding interrupt vector number.

Figure 33.41 Example of Start-of-Transmission Operation for Serial Transmission in Manchester Mode (without Preface but with the CTS Function)



Note 1. Refer to section 14, Interrupt Controller (ICUG) for details on the corresponding interrupt vector number.

Figure 33.42 Example of End-of-Transmission Operation for Serial Transmission in Manchester Mode (with Preface but without the CTS Function)

33.5.7 Serial Data Reception in Manchester Mode

In manchester mode, the RSCI operates on a base clock with a frequency of 16 times*1 the bit rate. Reception starts by sampling the falling edges of received data at the base clock. As shown in Figure 33.43, reception starts at a falling edge of the received data and it continues if the received data keeps low for the duration of 1/4 bit. If the received data goes high within the duration of 1/4 bit, the RSCI judges it as an error and waits for a falling edge again.
If a high level is expected in the first half of a bit in the received data, the RSCI judges a low level that continues for one base clock cycle as an error and ignores the change to the low level.

Note 1. This is the case when SCR2.ABCS bit = 0. When SCR2.ABCS bit = 1, the RSCI operates on a base clock with a frequency of 8 times the bit rate.

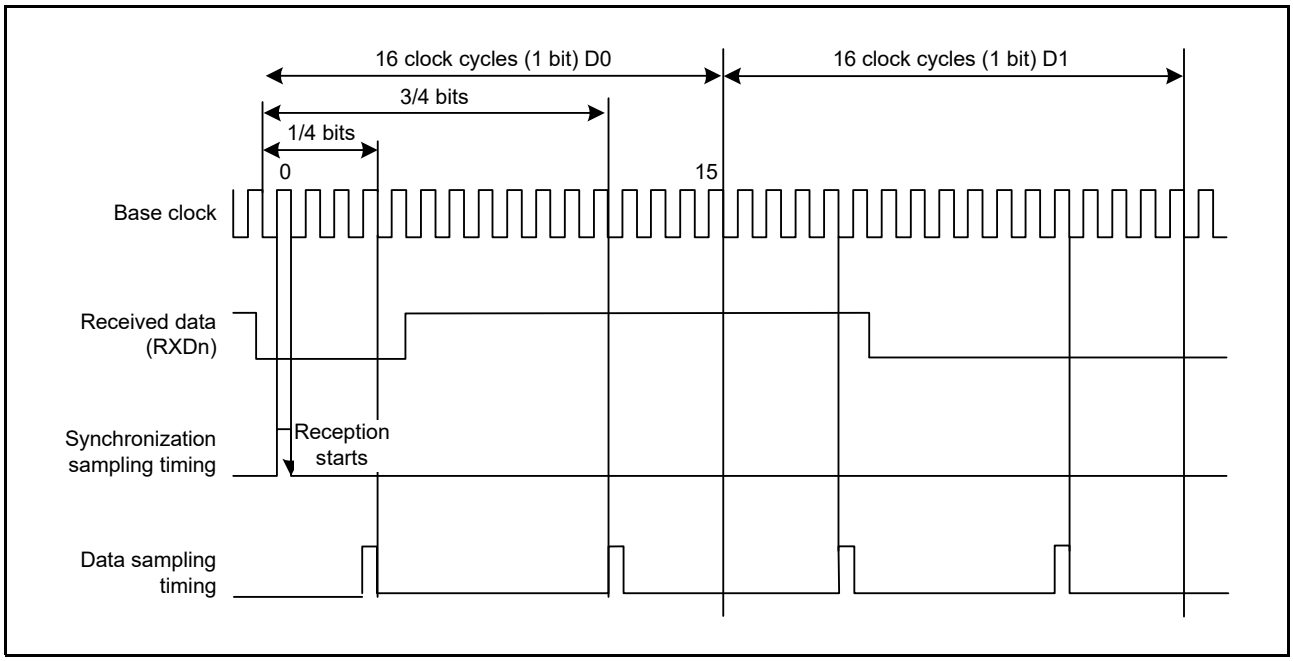
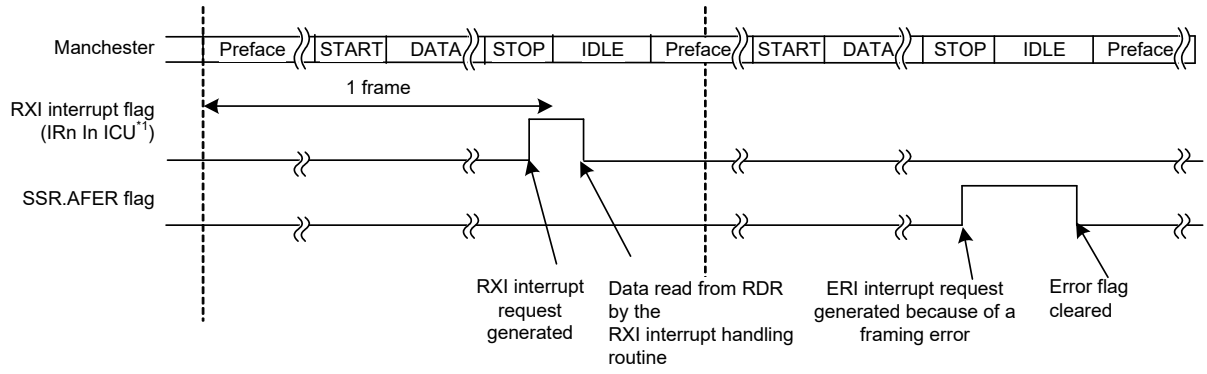


Figure 33.43 Data Reception Sampling Timing in Manchester Mode

In manchester mode, data reception starts with detection of a preface and start bit area.
The RSCI checks the input from the RXDn pin to see whether a preface is added based on the value of MMCR.RPLEN[3:0] bits.
If the preface is disabled (MMCR.RPLEN[3:0] bits = 0), it moves on to the detection of a start bit area without detecting a preface.
When a preface is enabled, it identifies a preface pattern setting according to the set value in MMCR.RPPAT[1:0] bits, and compares it with the RXD input for a pattern match to detect a preface pattern.
Upon detection of a preface pattern match, it judges it as a normal preface and moves on to the detection of a start bit area.
If detecting a preface pattern mismatch or a Manchester code error in the preface area, it judges it as a preface error and asserts a preface error (PFER).
For start bit detection, the RSCI selects an expected value based on the register settings (MMCR.SBLEN and SBPTN bits), compares it with the RXD input for a pattern match to detect a start bit area. Upon detection of a start bit pattern match, it judges it as a normal start bit area and moves on to the data processing.
Only when a preface and a start bit area are detected normally, it moves on to the next phase of data reception.
Upon detection of a start bit pattern mismatch, it asserts a start bit error flag (SBER).
In data processing, the RSCI shifts the data by the expected received data length based on the register settings (SCR3.CHR[1:0] bits) through the RSR register. If two sampling points in a bit of the received data are identical, the RSCI judges this as a Manchester code error. For details, see section 33.5.11, Errors in Manchester Mode (4).
When the parity function is disabled (SCR1.PE bit = 0), the RSCI moves on to the next phase of stop bit detection. When

the parity function is enabled (SCR1.PE bit = 1), the RSCI performs parity checking. If detecting a parity error, it asserts a parity error flag (APER), and then moves on to stop-bit detection.
In stop bit detection, the RSCI checks the following in the stop bit area of the received frame:
It has two sampling points in a bit. If both points are at the high level, the bit is recognized as a normal stop bit and the data is stored in the RDR register. At least one low-level point is judged as an abnormal stop bit, causing a framing error flag (AFER) to be set. Even when an error is detected, the received data is stored in the RDR register as abnormal data.
Figure 33.44 shows an example of the operation for serial data reception in manchester mode.

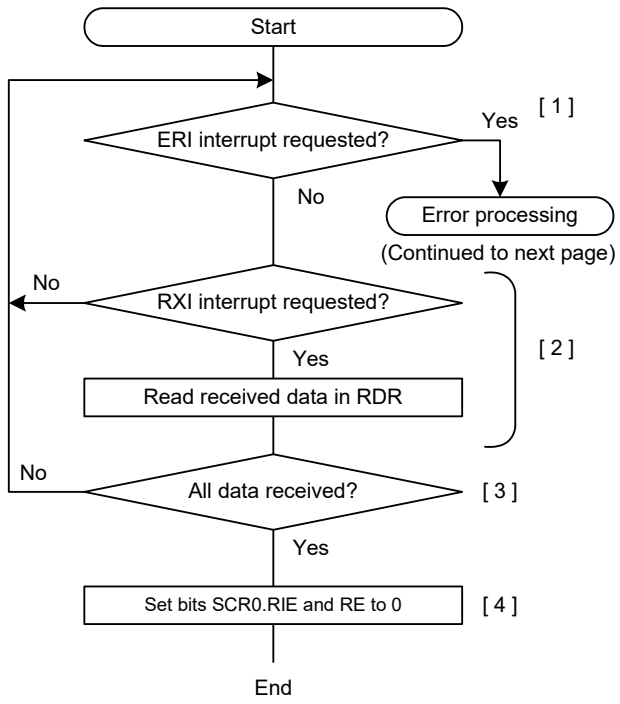


Note 1. Refer to section 14, Interrupt Controller (ICUG) for details on the corresponding interrupt vector number.

Figure 33.44 Example of Operation for Serial Data Reception in Manchester Mode (with a Preface)

For the state of each status flag in the SSR and MMSR registers and RXD input processing when a receive error is detected, see section 33.5.11, Errors in Manchester Mode.
If a receive error is detected, an ERI interrupt request is generated but an RXI interrupt request is not generated.
Data reception cannot be resumed while the receive error flag is 1. Accordingly, set the ORER, AFER, APER, MCER, SYER*2, PFER*2, SBER*2 flags to 0 before resuming reception. Also, be sure to read the RDR register during overrun error processing. When a reception is forcibly terminated by setting the SCR0.RE bit to 0 during operation, read the RDR register because received data which has not yet been read may be left in the RDR register.
Figure 33.45 and Figure 33.46 show examples of serial data reception flowchart in manchester mode.

Note 2. Effective when the corresponding bit is enabled.



[1] Receive error processing:
When a receive error occurs, RSCI generates an ERI interrupt. An error is identified by reading the ORER, APER, AFER in SSR register and MCER, SBER, SYER, PFER in MMSR register. After performing the appropriate error processing, always set the ORER, APER, AFER, MCER, SBER, SYER, PFER flags to 0. Reception cannot be resumed if any of these flags is set to 1.

Note: Each of the PFER, SYER, SBER bits in MMSR register has an enable bit (PFERIE, SYERIE, and SBERIE bits in MMCR). If the corresponding enable bit is set to 0 (disable), the flag value does not affect the communication. Perform error processing as required.

[2] Read the received data in RDR register once in the RXI interrupt handling routine.

[3] Serial reception continuation procedure:
To continue serial reception, before the stop bit of a frame is received, read data from RDR register in the RXI interrupt handling routine. The RDR data can also be read by activating the DMAC or DTC.

[4] Write 0 to SCR0.RIE and RE bits.

Figure 33.45 Example of Serial Data Reception in Manchester Mode (Normal)

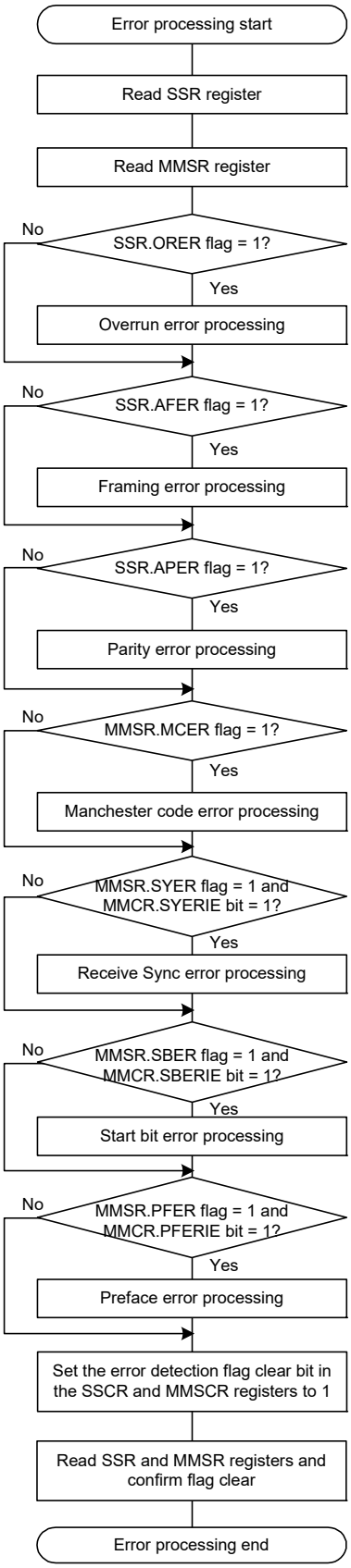


Figure 33.46 Example of Serial Reception Flowchart in Manchester Mode (Error Processing)

33.5.8 Operation When Multi-Processor Bit is Used

See section 33.4, Multi-Processor Communication Function (1) for the operation in manchester mode when using multi-processor mode because the operation is the same.
A preface and a start bit area are added to the frame format in manchester mode.
See Figure 33.46 for error processing in manchester mode for the reception flowchart (Figure 33.29). Refer to Table 33.37 for the operation status when detecting various errors.

33.5.9 Receive Retiming

This function corrects the timing for each central edge of the bit, taking advantage of the fact that each bit has an edge in the center in Manchester code.
The receive retiming function can be turned on or off by setting the SADJE bit in the MMCR register.
When the receive retiming function is turned off (MMCR.SADJE bit = 0), retiming is not performed, causing misalignment between the internal clock and the RXD input to be accumulated and the receive margin to be reduced.
When the receive retiming function is turned on (MMCR.SADJE bit = 1), retiming is performed for the preface area, the start bit area*1, and the data area (excluding the stop bit).

Note 1. Retiming is not performed for the start bit area if the preface length is 0 and the start bit length is 3.

As an example, the receive retiming when oversampling ×16 is selected is shown below.
When detecting an RXD input edge two to four cycles before the expected receive cycle, the receive processing is shortened by one sampling CLK cycle.
When detecting a RXD input edge two to three cycles after the expected receive cycle, the receive processing is extended by one sampling CLK cycle.
(Even if the clock is misaligned with the data by more than two cycles, one cycle is corrected for each bit.)
Figure 33.47 shows the conceptual image of receive retiming range.
When detecting an edge in the tolerance area in the figure, data is received as is without making correction.
When detecting an edge in the SyncJump area in the figure, data is corrected for reception.
When detecting an edge in the SyncError area in the figure, data is received as abnormal data with no correction made.
For a Manchester code error (data matches at the 1/4-phase and 3/4-phase sampling points), the RSCI reports a code error.

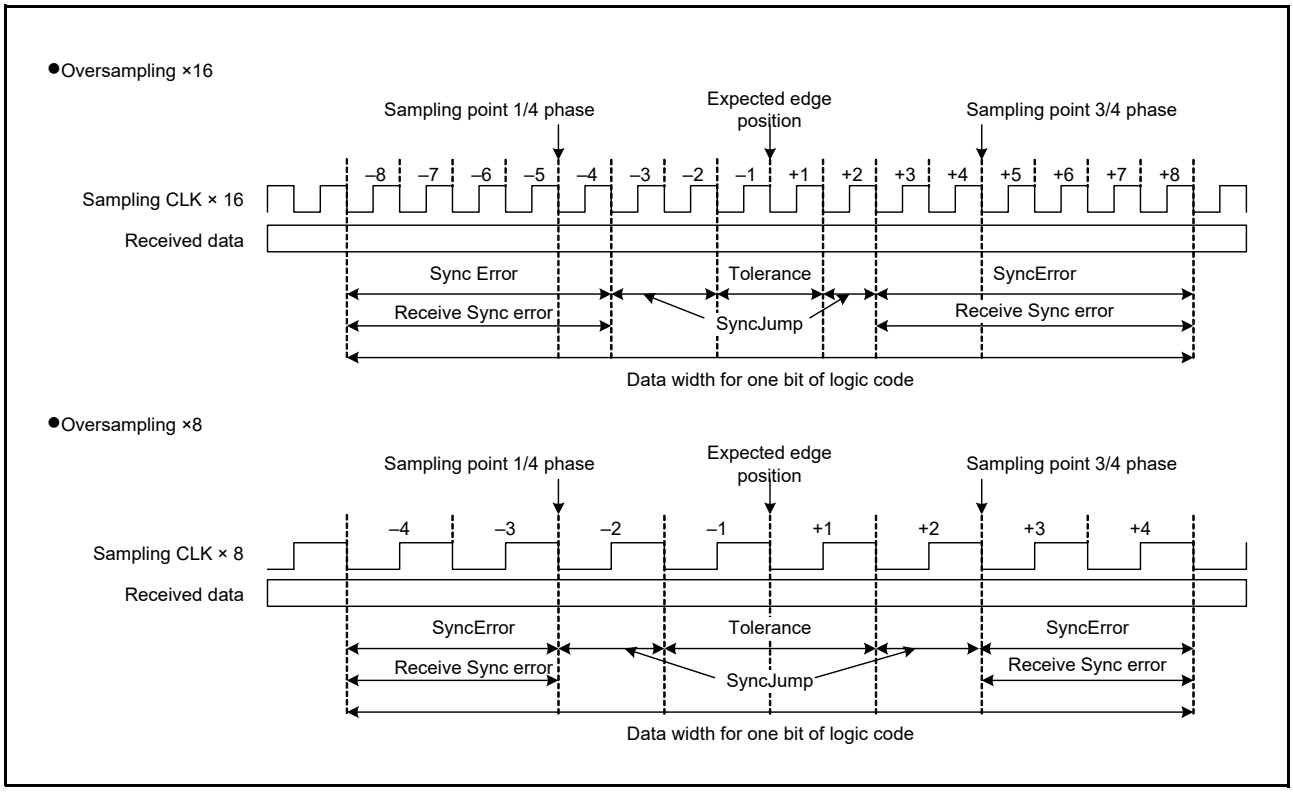


Figure 33.47 Conceptual Image of Reception Retiming Range

33.5.10 Polarity Setting for Manchester Code

The polarity of the Manchester code can be set with the Manchester Control Register (MMCR).

It can be set separately for transmission and reception. Use the MMCR.ENCS bit to set the polarity for transmission and the MMCR.DECS bit to set the polarity for reception.

The Manchester code polarity setting is valid for the preface area, the data area, and the parity or multi-processor area.

When the initial settings (ENCS/DECS bit = 0) are used for the polarity of Manchester code, logic 0 is encoded as a zero-to-one transition in Manchester code and logic 1 is encoded as a one-to-zero transition in Manchester code.

If the settings are changed to ENCS/DECS bit = 1, logic 0 is encoded as a one-to-zero transition in Manchester code and logic 1 is encoded as a zero-to-one transition in Manchester code. Figure 33.48 shows the conceptual image of the settings and operation.

Separately from the function above, the transmitted and received data in the data area can be inverted by the transmitted/received data inversion function (SCR3.DINV bit). Since the polarity of Manchester code (MMCR.ENCS/DECS bit) can be set separately from the transmitted/received data invert function (SCR3.DINV bit), if both are set to inversion (MMCR.ENCS/DECS bit = 1 and SCR3.DINV bit = 1), the transmitted and received data are set to initial state (inversion + inversion = normal).

The polarity of the start bit area can be set by a register different from the ones mentioned above.

Since a different register is used, the polarity of the start bit area is not affected by the polarity setting for Manchester code mentioned above.

For details on the setting for the start bit area, see section 33.5.1, Frame Format (2).

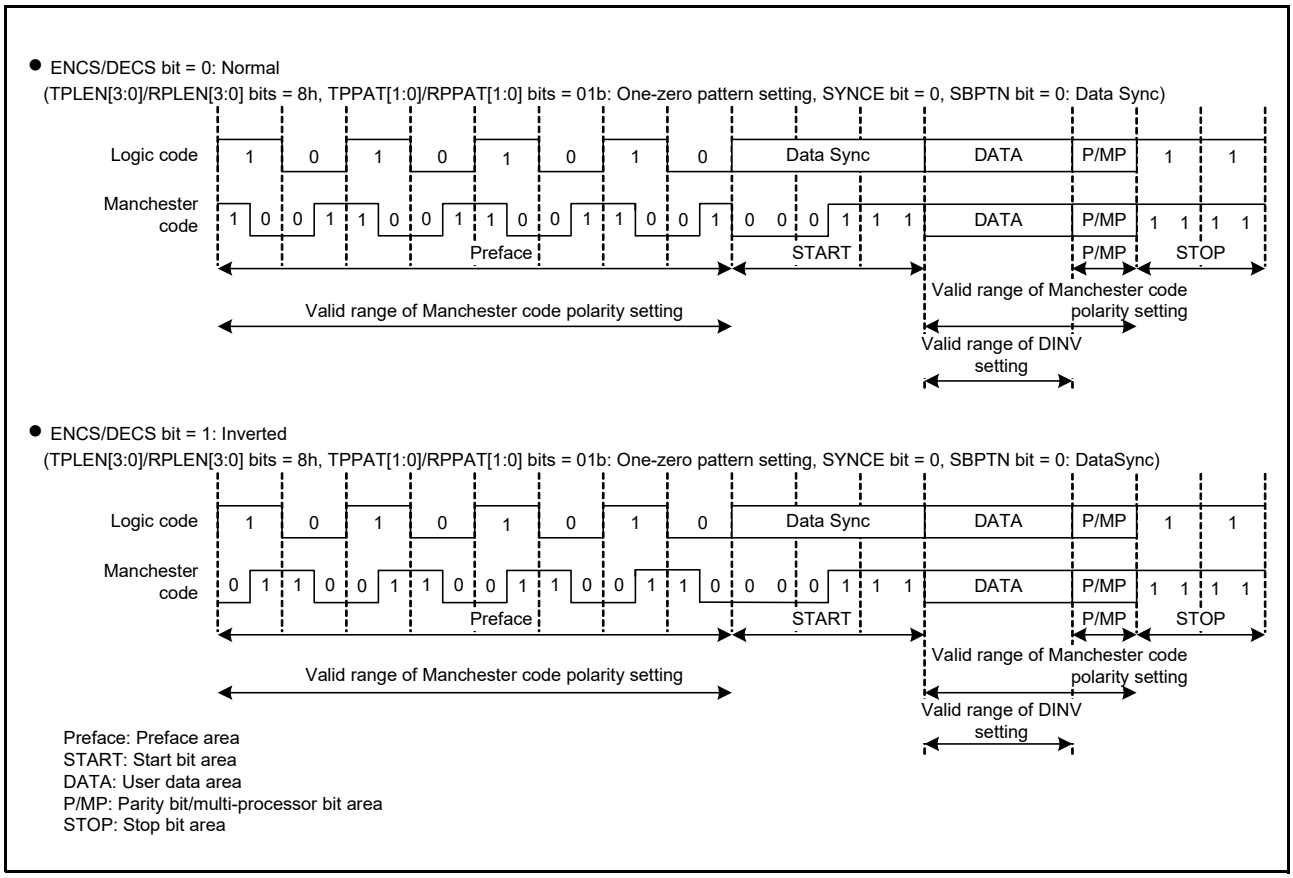


Figure 33.48 Valid Range of the Manchester Code Polarity Setting

33.5.11 Errors in Manchester Mode

There are the following errors in manchester mode:

(1) Parity error
(2) Over run error
(3) Framing error
(4) Manchester code error
(5) Preface error
(6) Start bit error
(7) Receive Sync error

For errors (1) to (3), see section 33.3.9, Serial Data Reception (Asynchronous Mode) (1) because they are the same as in asynchronous mode.
Each error is judged in each area, but they are reflected on flags and operations at the timing of 3/4-bit sampling of the STOP bit area. If a preface error or start bit error is detected, subsequent data will not be received. Therefore, no other error detection is performed, and the error flag holds the previous information.
Table 33.35 lists the states of the serial status register when detecting errors and judgment about whether to store data in the RDR register.
Table 33.36 lists the errors that can be detected in each area of a Manchester frame. If a Preface error or Start bit error is detected, subsequent data will not be received. Therefore, no other error detection is performed, and the error flag holds the result of the previous frame reception. Also, if an error is detected in the previous frame, data will not be received, but errors in the preface area and start bit area will update that flag. Table 33.37 shows the flags and actions in this case.

(4) Manchester Code Error

A Manchester code error is generated when a Manchester code error is detected.
In Manchester code, there must be an edge (transition) in the center of the bit.
In the data area of a received frame (including the parity/multi-processor bit), the values of the 1/4-bit and 3/4-bit sampling points are checked in each received 1-bit data, and a Manchester code error is determined if these two values matches.
If a Manchester code error is detected, the Manchester code error flag (MMSR.MCER flag) is asserted.
If a Manchester code error occurs, it is handled as an interrupt source and event source. If a Manchester error is detected, the next reception is not performed until the corresponding error flag is cleared.

(5) Preface Error

A preface error is generated when the preface pattern does not match or a Manchester code error is detected in the preface area. If a preface error is detected, the preface error flag (MMSR.PFER flag) is asserted.
It is possible to set whether to use this error flag as an interrupt source with the setting of the MMCR register.
When MMCR.PFERIE bit = 1, a preface error is handled as an interrupt source or event source. If a preface error is detected, the next reception is not performed until the corresponding error flag is cleared.
When MMCR.PFERIE bit = 0, a preface error is not handled as an interrupt source or event source, and the next reception is not halted. However, a preface error is notified to MMSR.PFER flag.

(6) Star Bit Error

A start bit error is generated when a mismatch is detected between the start bit area in the received frame and the preset start bit pattern. Upon detection of a start bit error, a start bit error flag (MMSR.SBER) is asserted.
It is possible to set whether to use the start bit error as an interrupt source with the setting of the MMCR register.
When MMCR.SBERIE bit = 1, a start bit error is handled as an interrupt source or event source. If a start bit error is detected, the next reception is not performed until the corresponding error flag is cleared.

When MMCR.SBERIE bit = 0, a start bit error is not handled as an interrupt source or event source, and the next reception is not halted. However, a start bit error is notified to MMSR.SBER flag.

(7) Receive Sync Error (SyncError)

When the receive retiming function described in section 33.5.9, Receive Retiming is enabled, the receive retiming operation is performed.
If no edges are detected within the receive retiming range (SyncError area in Figure 33.47) when receive timing operation is being performed, a receive Sync error is generated. Upon detection of a receive Sync error, a receive Sync error flag (MMSR.SYER) is asserted. In areas not subject to receive retiming, receive Sync errors are not detected.
The preface area *1, the start bit area*1, *2, and the data area (excluding the stop bit) for which receive retiming operation is performed are checked.
It is possible to set whether to use the receive Sync error as an interrupt source with the setting of the MMCR register.
When MMCR.SYERIE bit = 1, a receive Sync error is handled as an interrupt source or event source. If a receive Sync error is detected, the next reception is not performed until the corresponding error flag is cleared.
When MMCR.SYERIE bit = 0, a receive Sync error is not handled as an interrupt source or event source, and the next reception is not halted. However, a receive Sync error is notified to MMSR.SYER flag.

Note 1. In the case of a frame that starts with a pattern that expects the first half of the bit to be High, it is excluded from retiming.
Note 2. In the start bit area, when there is no preface length and 3 bit start bit is set, it is not subject to retiming. Also, the 1st bit and the 2nd bit in the start bit area when 3 bit start bit is set are not subject to retiming.

Table 33.35 Flags in the Status Register and Receive Data Handling in Manchester

Flags in the SSR Register			Flags in the MMSR Register					
ORER	AFER	APER	MCER	SBER *1	PFER *1	SYER	Received Data	Received Error Status (ERI Interrupt/Event Generation)
0	0	0	0	0	0	0	Transfer to RDR	No error
0	1	0	0	0	0	0	Transfer to RDR	Framing error
0	0	1	0	0	0	0	Transfer to RDR	Parity error
0	1	1	0	0	0	0	Transfer to RDR	Framing error + Parity error
0	0	0	1	0	0	0	Transfer to RDR	Manchester code error
0	1	0	1	0	0	0	Transfer to RDR	Framing error + Manchester code error
0	0	1	1	0	0	0	Transfer to RDR	Parity error + Manchester code error
0	1	1	1	0	0	0	Transfer to RDR	Framing error + Parity error + Manchester code error
1	0	0	0	0	0	0	Lost	Overrun error
1	1	0	0	0	0	0	Lost	Overrun + Framing error
1	0	1	0	0	0	0	Lost	Overrun + Parity error
1	1	1	0	0	0	0	Lost	Overrun + Framing error + Parity error
1	0	0	1	0	0	0	Lost	Overrun + Manchester code error
1	1	0	1	0	0	0	Lost	Overrun + Framing error + Manchester code error
1	0	1	1	0	0	0	Lost	Overrun + Parity error + Manchester code error
1	1	1	1	0	0	0	Lost	Overrun + Framing error + Parity error + Manchester code error
0	Combination of the above errors			0	0	1	Transfer to RDR	Error above + Receive Sync error*2
1				0	0	1	Lost	Error above + Receive Sync error*2
hold	hold	hold	hold	0	1	0	Lost	Preface error*3
hold	hold	hold	hold	1	0	0	Lost	Start bit error*3
hold	hold	hold	hold	0	1	1	Lost	Preface error*3 + Receive Sync error*2
hold	hold	hold	hold	1	0	1	Lost	Start bit error*3 + Receive Sync error*2

Note 1. Start bit error and preface error never become 1 at the same time.
Note 2. When MMCR.SYERIE bit = 1, ERI interrupt/event is generated by SYER factor.
Note 3. If MMCR.PFERIE bit = 1 or MMCR.SBERIE bit = 1, an ERI interrupt/event is generated when the corresponding flag is set.

Table 33.36 Errors Detectable in Each Area

	Preface Error (PFER)	Start Bit Error (SBER)	Manchester Code Error (MCER)	Receive Sync Error (SYER)	Parity Error (APER)	Framing Error (AFER)
Preface area	✓	—	—*1	✓*2	—	—
Start bit area	—	✓	—	✓*2	—	—
Data area	—	—	✓	✓	—	—
Parity area	—	—	✓	✓	✓	—
Multi-processor area	—	—	✓	✓	—	—
Stop bit area	—	—	—	—	—	✓

✓: Detected, —: Not detected
Note 1. When an Manchester code error occurs in the preface area, it is defined as a preface error.
Note 2. It may not be subject to receive Sync error detection. For details see section 33.5.11, Errors in Manchester Mode (7)

Table 33.37 Operation Status due to Presence/Absence of Error in Previous Frame and Operation Status List in Multiprocessor Mode (SCR0.MPIE Bit = 0)

Previous Frame	Each Area of the Frame					PFE RIE	SBE RIE	SYE RIE	Received Data	Error Flag	Interrupt Request	Event Signal
	Preface	Start Bit	Data	Parity Bit	Stop Bit							
No error	PFER, No SYER*1	No error	—	—	—	0	—	—	Lost	Set PFER*1	Not output	Not output
						1					Output	Output
	No error	SBER, No SYER*1	—	—	—	—	0	—	Lost	Set SBER*1	Not output	Not output
							1				Output	Output
	SYER, No PFER	No error	—	—	—	—	—	0	Transfer to RDR	Set SYER	Not output	Not output
								1	Lost		Output	Output
	No error	SYER, No SBER	—	—	—	—	—	0	Transfer to RDR	Set SYER	Not output	Not output
								1	Lost		Output	Output
	No error	No error	SYER		No error	—	—	0	Transfer to RDR	Set SYER	Not output	Not output
								1			Output	Output
	No error	No error	MCER		No error	—	—	—	Transfer to RDR	Set MCER	Output	Output
	No error	No error	—	APER	No error	—	—	—	Transfer to RDR	Set APER	Output	Output
	No error	No error	—	—	AFER	—	—	—	Transfer to RDR	Set AFER	Output	Output
	There is some error ORER					—	—	—	Lost	Set some flags*2	Output	Output
	No error	No error	No error	No error	No error, ORER	—	—	—	Lost	Set ORER	Output	Output
Some error*3, *6	PFER, No SYER*1	No error	—	—	—	0	—	—	Lost	Set PFER*1	Output*4	Not output*5
						1						
	No error	SBER, No SYER*1	—	—	—	—	0	—		Set SBER*1		
							1					
	SYER, No PFER	No error	—	—	—	—	—	0		Set SYER		
								1				
	No error	SYER, No SBER	—	—	—	—	—	0		Set SYER		
								1				
	No error	No error	SYER		No error	—	—	0		don't set any flags		
								1				
	No error	No error	MCER		No error	—	—	—				
	No error	No error	—	APER	No error	—	—	—				
	No error	No error	—	—	AFER	—	—	—				
	There is some error ORER					—	—	—				
	No error	No error	No error	No error	No error, ORER	—	—	—				

Note 1. If SYER is detected, the SYER flag is also set. Other operations are as shown in this table.
Note 2. Other detected error flags including ORER are also set.

Note 3. If all the error flags are cleared before the STOP bit is judged, the operation will be the same as the case where there is no error in the previous frame of this table.
Note 4. Since the ERI interrupt request is level output, it remains active due to errors in the previous frame regardless of the presence or absence of errors in the relevant frame.
Note 5. Since the error cause is continuously detected, the ERI event is not newly output regardless of the presence of errors in the relevant frame.
Note 6. For MMSR.PFER, SBER, and SYER, when each enable bit is set to disable, it is treated as no error.

Table 33.38 Operation Status List in Multiprocessor Mode (SCR0.MPIE Bit = 1)

MPB*1	Each Area of the Frame					PFERIE	SBERIE	SYERIE	Received Data	Error Flag	Interrupt Request	Event Signal
	Preface	Start Bit	Data	Parity Bit	Stop Bit							
1	No error	No error	—	—	—	—	—	—	Transfer to RDR	Set some flags	Output	Output
	No PFER, SYER	No SBER, SYER	—	—	—	—	—	0				
								1	Lost	don't set any flags	Not output	Not output
	PFER	No error	—	—	—	—	—	—				
	No error	SBER	—	—	—	—	—					

Note 1. If the received MPB flag is 0, it is not received the frame, and the operation is the same as lost of the reception data of this table.
Note 2. If no error is detected, RXI interrupt request or event is output, and if it is detected, ERI interrupt request or event is output.
Note 3. When SYER is detected in the preface area or the start bit area, the behavior of handling as an error depending on the MMCR.SYERIE bit changes.

33.6 HBS Support Mode

Setting the HBSCR.HBSE bit to 1 supports the negative logic RZI coding to generate waveforms (AMI, 50% duty cycle, negative logic) required by the home bus system. Since this function operates only in the asynchronous mode, refer to the asynchronous mode for the setting, transmission flow, and reception flow.

33.6.1 Reception in HBS Support Mode

When receiving in HBS support mode, the falling edge of the input from the RXDn pin is detected and the signal after the start bit is recognized is received. One frame is sampled according to the set bit rate, and if the stop bit is correctly received without error, the data value is stored in the receive data register RDR.

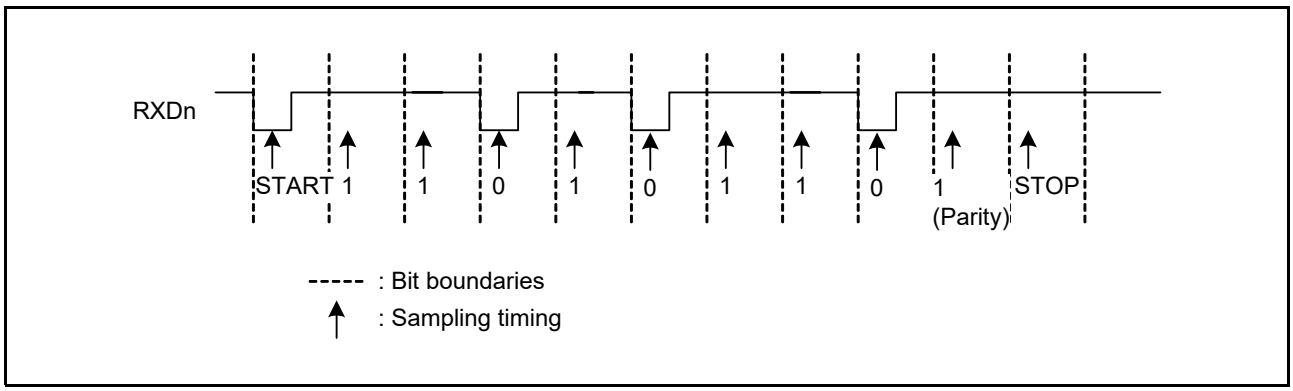


Figure 33.49 HBS Support Mode Reception Timing Chart

To receive in the HBS support mode, it is necessary to sample at the timing of 1/4 of 1 bit in order to capture the pulse in the first half of the 1 bit period. Sampling operates with a frequency 16 times the bit rate*1 as the base clock, similar to the asynchronous mode. The start bit is detected by detecting the Low level from the falling edge of RXD four times continuously with the base clock. When High is detected on the way, it is regarded as noise and waits for the next fall edge.

To set the sampling timing to 1/4 of the 1-bit period, enable the reception sampling timing adjustment function with the SCR4.RTADJ bit, set the SCR4.RTMG[3:0] bits to 1100b, and adjust Adjust from the center of the bit, which is the previous sampling timing, four clocks ahead of the base clock.

Since the sampling timing can be adjusted backward and forward using the reception sampling timing adjustment function, this timing can be adjusted according to the reception status. Increasing the SCR4.RTMG[3:0] bits value from 1100b will move the sampling timing forward, and decreasing it will move it backward. Refer to section 33.3.2, Receive Data Sampling Timing and Reception Margin in Asynchronous Mode for the details of adjustment.

After recognizing the start bit, sampling is performed at the timing according to the set bit rate, but the low width and high width of the waveform are not checked. Therefore, it is possible to receive even a normal asynchronous waveform.

Note 1. HBS support mode supports only following setting: SCR2.ABCS bit = 0 and SCR2.ABCSE bit = 0.

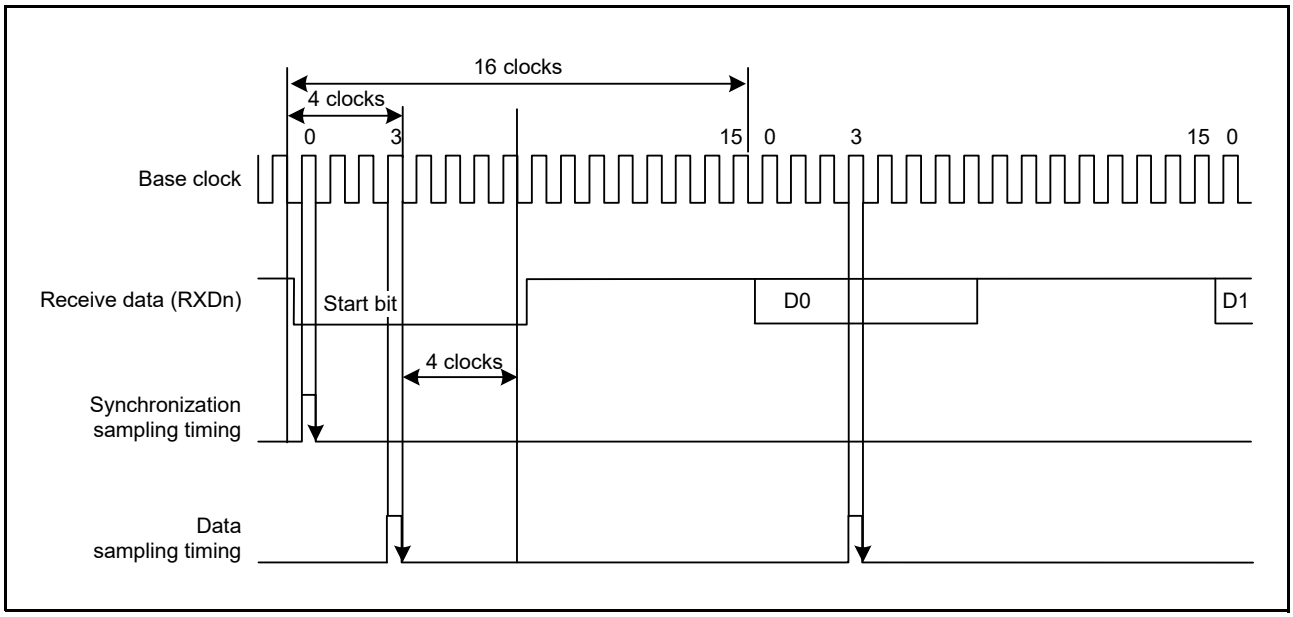


Figure 33.50 Details of Reception Sampling Timing in HBS Support Mode

33.6.2 Transmission in HBS Support Mode

Transmission in HBS support mode, data 0 is output as a low pulse for the first half of the 1-bit period.

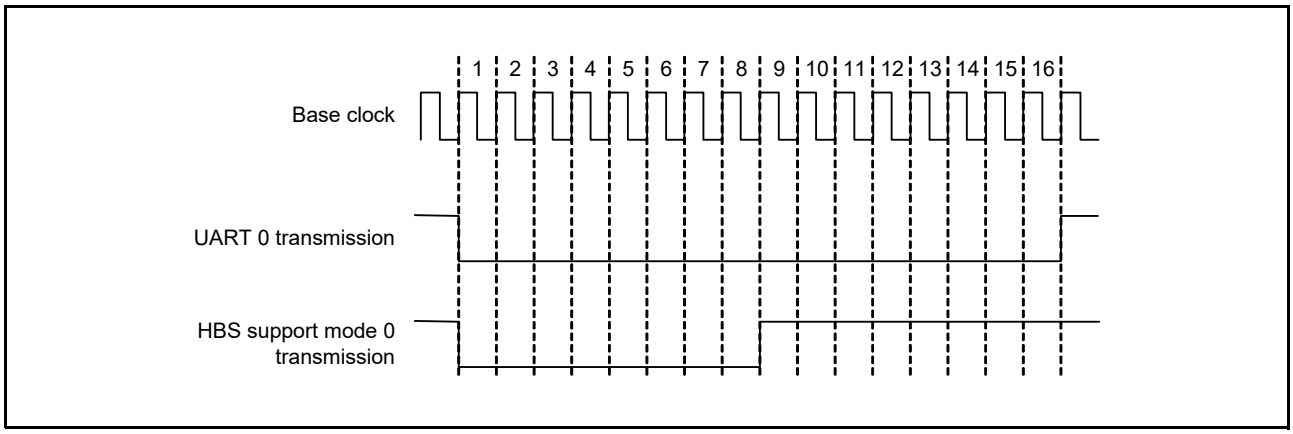


Figure 33.51 Transmission Waveform of HBS Support Mode

When HBSCR.AOE bit = 0, the all bits are output from TXDn pins, and when HBSCR.AOE bit = 1, data 0 is output alternately from TXDAn and TXDBn pins. Use the HBSCR.LPS bit to select which transmission pin starts output of the start bit.

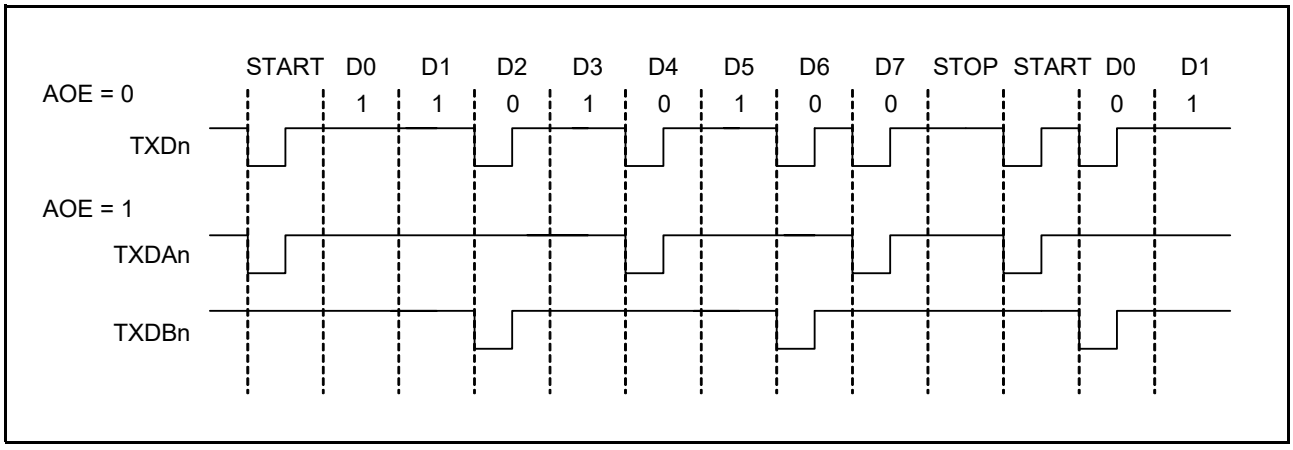


Figure 33.52 Difference in Transmission Waveform Depending on AOE Bit (When LPS Bit = 0)

Figure 33.52 shows an example of the transmission waveform for each HBSCR.AOE bit value. When the AOE bit is set to 0, the waveform is output from the TXDn pin, but when the AOE bit is set to 1, data 0 including start bit is output alternately from the TXDAn pin and the TXDBn pin.

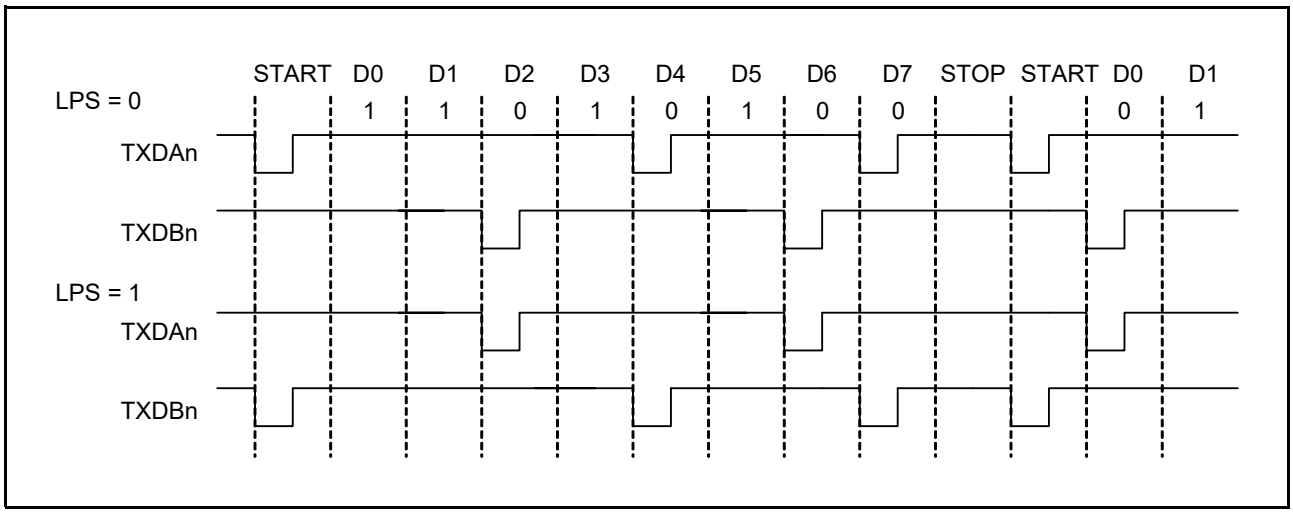


Figure 33.53 Difference in Transmission Waveform Depending on LPS Bit (When AOE Bit = 1)

Figure 33.53 shows an example of the transmission waveform for each HBSCR.LPS bit.
When the HBSCR.LPS bit = 0, the start bit is output from the TXDAn pin, and when the HBSCR.LPS bit is 1, the start bit is output from the TXDBn pin, and data 0 is output to each pin alternately. The start bit of the next frame starts output again from the pin specified by the HBSCR.LPS bit.
If the HBSCR.HBSE bit = 0, the TXDBn pin becomes High regardless of the settings of other bits.
When SCR0.TE bit = 0, both TXDn/TXDAn/TXDBn pins become high impedance, but can be controlled by SCR1.SPB2IO bit and SCR1.SPB2DT bit. At this time, the same output is applied to the TXDn/TXDAn/TXDBn pins.

33.6.3 Register Setting for HBS Support Mode

The HBS support mode is a part of asynchronous mode function, but there are some settings that are not supported when using this function. Set each bit of the control register as shown in Table 33.39 before use. Register bits not described can be set in the same way as the asynchronous mode.

Table 33.39 Control Register Setting Value for HBS Support Mode

Register Bit Name	Value	Remarks
SCR0.DCME	0	Use it when the data match detection function is disabled.
SCR1.NFCS[2:0]	000b	Use this setting when using the noise filter.
SCR1.HDSEL	0	Half-duplex communication with the TXDn pin cannot be used.
SCR1.CTSE	0	Please use it without the CTS function.
SCR2.BRME	0	Bit rate modulation function cannot be used.
SCR2.ABCSE	0	The setting that 6 cycles of the base clock becomes 1 bit cannot be used.
SCR2.ABCS	0	Only the setting that 16 cycles of the base clock becomes 1 bit can be used.
SCR3.CKE[1:0]	00b	Use with internal clock and without clock output.
SCR3.DEEN	0	Use without the RS-485 driver function.
SCR3.FM	0	Use without FIFO function.
SCR3.MOD[2:0]	000b	Set to asynchronous mode.
SCR3.RXDESEL	1	Detect the start bit at the falling edge of the RXDn pin input.
SCR3.STOP	0	Use with stop bit 1.
SCR3.DINV	0	Use without data inversion.
SCR3.DDIR	1	Use with LSB first.
SCR3.CHR[1:0]	10b	Use with 8 bit length.
SCR4.RTMG[3:0]	1100b	Use this setting when receiving in HBS support mode.*1
SCR4.TTADJ	0	Use without adjust transmit timing function.
SCR4.RTADJ	1	Use this setting when receiving in HBS support mode.

Note 1. This is the timing to sample at the center of the effective pulse. It can be adjusted if needed.

33.7 Smart Card Interface Mode

The RSCI supports smart card (IC card) interfaces conforming to ISO/IEC 7816-3 (standard for Identification Cards), as an extended function.
Smart card interface mode can be selected using the appropriate register.

33.7.1 Sample Connection

Figure 33.54 shows a sample connection between a smart card (IC card) and this MCU.
As in the figure, since this MCU communicates with an IC card using a single transmission line, interconnect the TXDn and RXDn pins and pull up the data transmission line to VCC using a resistor.
Setting the TE and RE bits in the SCR0 register to 1 with an IC card disconnected enables closed transmission/reception allowing self-diagnosis.
To supply an IC card with the clock pulses generated by the RSCI, input the SCKn pin output to the CLK pin of an IC card.
The output port can be used to output the reset signal.

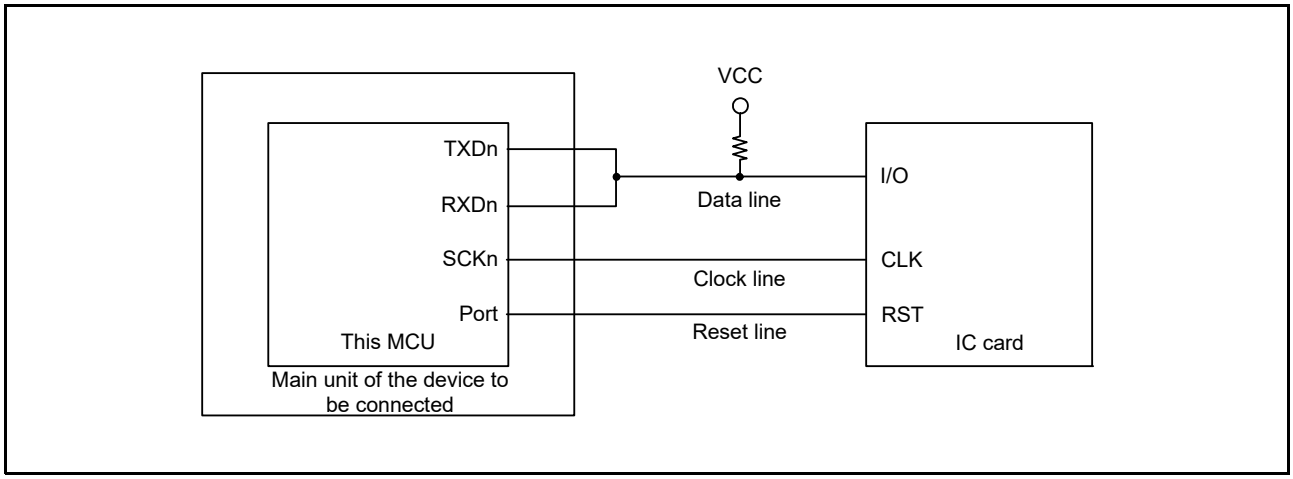


Figure 33.54 Sample Connection with a Smart Card (IC Card)

33.7.2 Data Format (Except in Block Transfer Mode)

Figure 33.55 shows the data transfer formats in smart card interface mode.

- One frame consists of 8-bit data and a parity bit in asynchronous mode.
- During transmission, at least 2 etu (elementary time unit: time required for transferring 1 bit) is secured as a guard time from the end of the parity bit until the start of the next frame.
- If a parity error is detected during reception, a low-level error signal is output for 1 etu after 10.5 etu has passed from the start bit.
- If an error signal is sampled during transmission, the same data is automatically retransmitted after at least 2 etu.

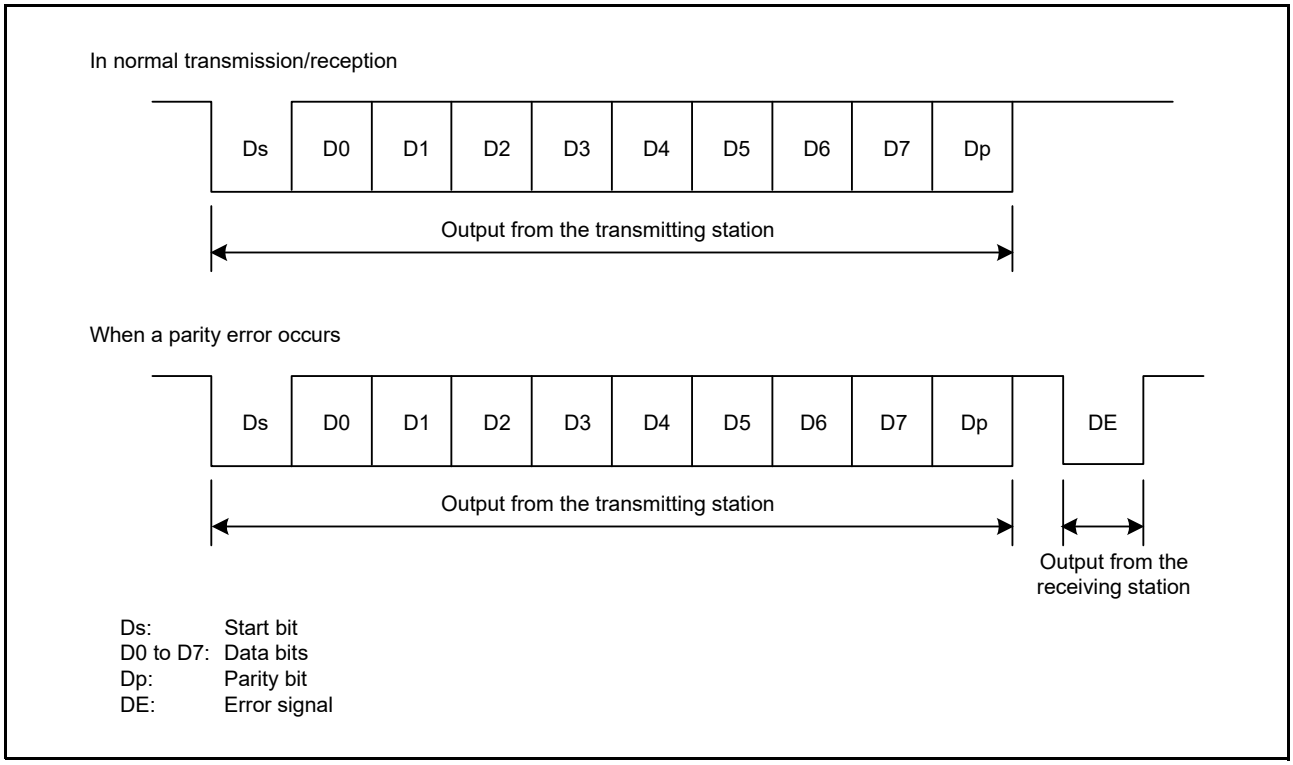


Figure 33.55 Data Formats in Smart Card Interface Mode

For communications with IC cards of the direct convention type and inverse convention type, follow the procedure below.

(1) Direct Convention Type

For the direct convention type, logic levels 1 and 0 correspond to states Z and A, respectively, and data is transferred with LSB first as the start character, as shown in Figure 33.56. Therefore, data in the start character in the figure is 3Bh. When using the direct convention type, write 1 to the SCR3.DDIR bit and 0 to the SCR3.DINV bit. Write 0 to the PM bit in the SCR1 register in order to use even parity, which is prescribed by the smart card standard.

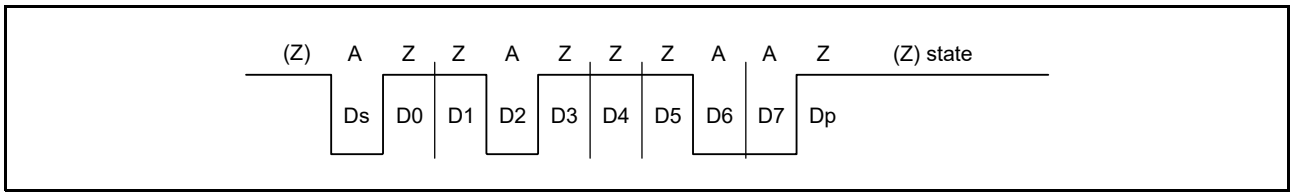


Figure 33.56 Direct Convention (SCR3.DDIR Bit = 1, SCR3.DINV Bit = 0, SCR1.PM Bit = 0)

(2) Inverse Convention Type

For the inverse convention type, logic levels 1 and 0 correspond to states A and Z, respectively and data is transferred with MSB first as the start character, as shown in Figure 33.57. Therefore, data in the start character in the figure is 3Fh. When using the inverse convention type, write 0 to the SCR3.DDIR bit and 1 to the SCR3.DINV bit. The parity bit is logic level 0 to produce even parity, which is prescribed by the smart card standard, and corresponds to state Z. Since the DINV bit of the this MCU only inverts data bits D7 to D0, write 1 to the PM bit in the SCR1 register to invert the parity bit for both transmission and reception.

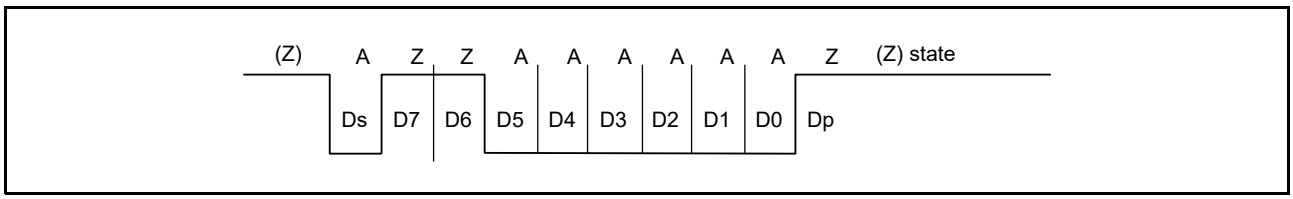


Figure 33.57 Inverse Convention (SCR3.DDIR Bit = 0, SCR3.DINV Bit = 1, SCR1.PM Bit = 1)

33.7.3 Block Transfer Mode

Block transfer mode is different from non-block transfer mode in the following respects.

- Even if a parity error is detected during reception, no error signal is output. Since the APER flag in the SSR register is set by error detection, clear the APER flag before receiving the parity bit of the next frame.
- During transmission, at least 1 etu is secured as a guard time from the end of the parity bit until the start of the next frame.
- Since the same data is not retransmitted during transmission, the TEND flag in the SSR register is set 11.5 etu after transmission start.
- In block transfer mode, the ERS flag in the SSR register indicates the error signal status as in non-block transfer mode, but the flag is read as 0 because no error signal is transferred.

33.7.4 Receive Data Sampling Timing and Reception Margin

Only the base clock generated by the on-chip baud rate generator can be used as a transmit/receive clock in smart card interface mode.

In this mode, the RSCI can operate on a base clock with a frequency of 32, 64, 372, 256, 93, 128, 186, or 512 times the bit rate according to the settings of the SCR2.BCP[2:0] bits.

For data reception, the falling edge of the start bit is sampled with the base clock to perform synchronization. Receive data is sampled on the 16th, 32nd, 186th, 128th, 46th, 64th, 93rd, and 256th rising edges of the base clock so that it can be latched at the middle of each bit as shown in Figure 33.58. The reception margin here is determined by the following formula.

$$M = \left| \left(0.5 - \frac{1}{2N}\right) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\ (\%)$$

M: Reception margin (%)
N: Ratio of bit rate to clock (N = 32, 64, 372, 256)
D: Duty cycle of clock (D = 0 to 1.0)
L: Frame length (L = 10)
F: Absolute value of clock frequency deviation

Assuming values of F = 0, D = 0.5, and N = 372 in the above formula, the reception margin is determined by the formula below.

M = {0.5 – 1/(2 × 372)} × 100 (%) = 49.866 (%)

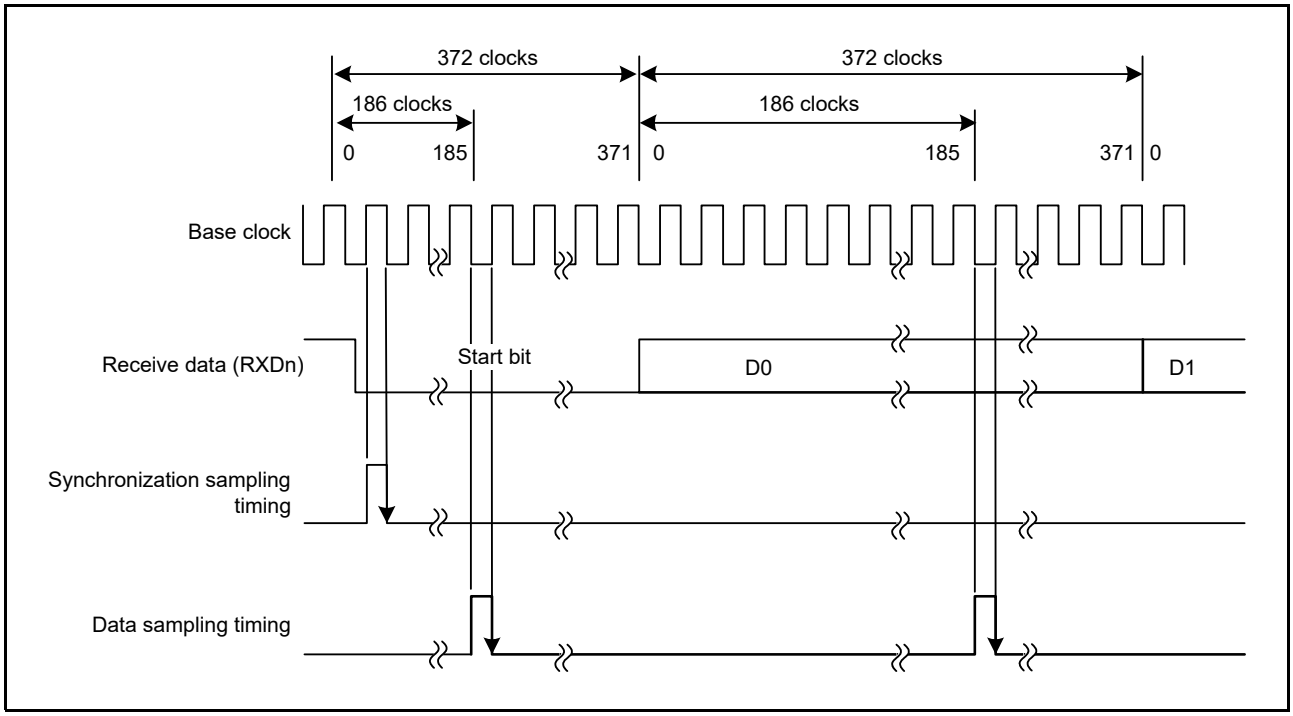


Figure 33.58 Receive Data Sampling Timing in Smart Card Interface Mode (When Clock Frequency is 372 Times the Bit Rate)

33.7.5 RSCI Initialization (Smart Card Interface Mode)

Before transmitting and receiving data, write 0 to SCR0.TE bit and SCR0.RE bit (or write the initial value to SCR0 register). And initialize RSCI following example flowchart in Figure 33.59.
Be sure to set the initial value in the TIE, RIE, TE, RE, and TEIE bits in SCR0 register before switching from transmission mode to reception mode and vice versa. Even if the RE bit is set to 0, the RDR register is not initialized. In transmission mode, set 1 to the TE bit and TIE bit simultaneously, then the TXI interrupt request is generated. To change reception mode to transmission mode, first check that reception has completed, and then initialize RSCI. At the end of initialization, set TE bit = 1 and RE bit = 0. Reception completion can be verified by RXI interrupt request, SSR.ORER, or SSR.APER flag. To change transmission mode to reception mode, first check that transmission has completed, and then initialize RSCI. At the end of initialization, set TE bit = 0 and RE bit = 1. Transmission completion can be verified by reading SSR.TEND flag.

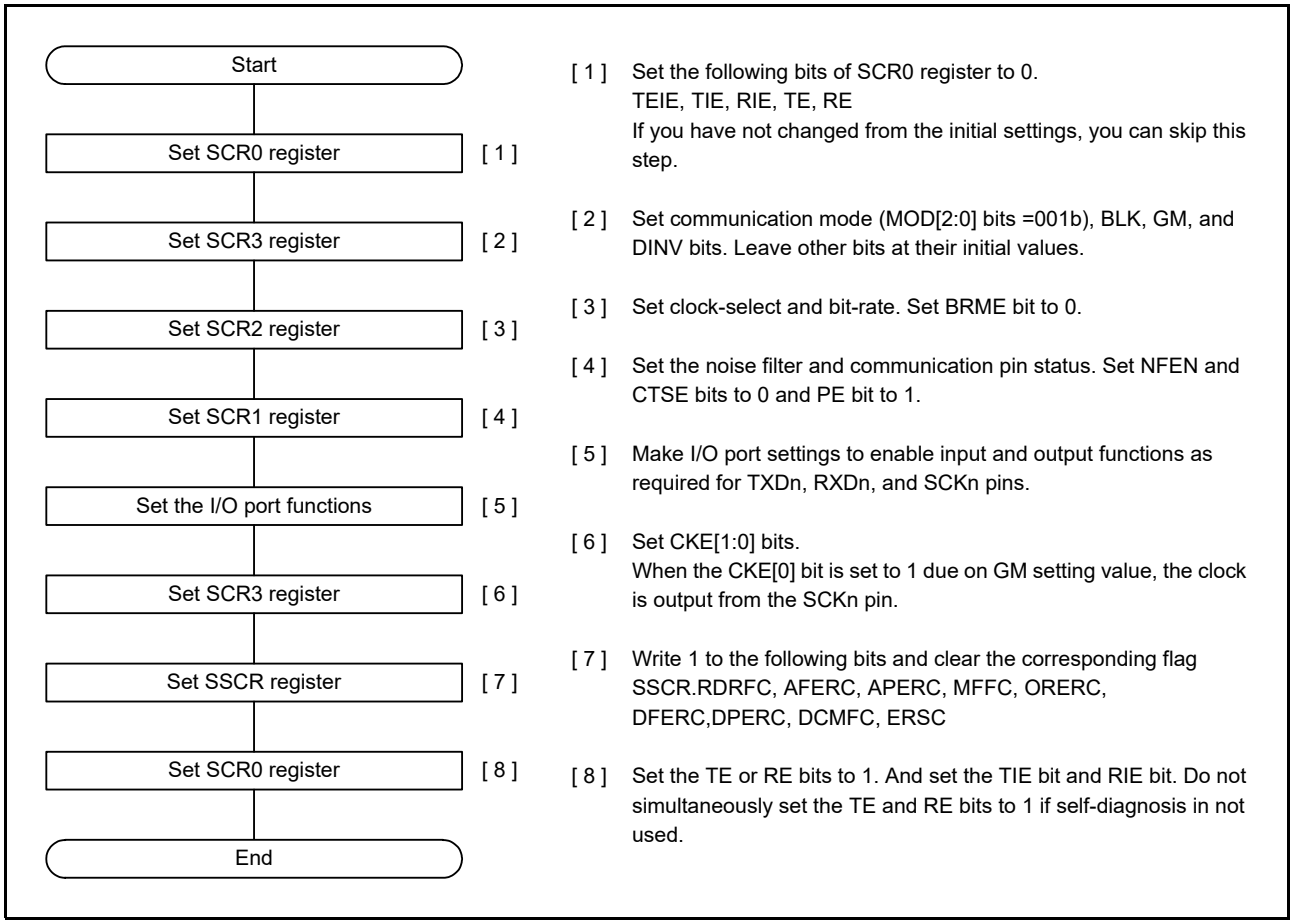


Figure 33.59 Example of RSCI Initialization Flowchart (Smart Card Interface Mode)

Figure 33.60 is a timing chart when data transmission is performed by making transition to the Smart Card Interface mode according to the above flow chart. The figure shows the case when SCR3.GM bit is 0. As shown in the figure, when the pin function is set to the SCKn pin, the SCKn pin is high impedance because the SCR3.CKE[0] bit is 0. When the TXDn pin is set, the TXDn pin is high impedance because the SCR0.TE bit is 0. Start clock output to the SCKn pin with the clock output setting SCR3.CKE[0] bit to 1, start data transmission by writing transmit data after setting SCR0.TE bit to 1.
In the smart card interface mode, even if not communicating at SCR0.TE bit = 0 and SCR0.RE bit = 0, the clock is continuously output if the clock output setting is used.

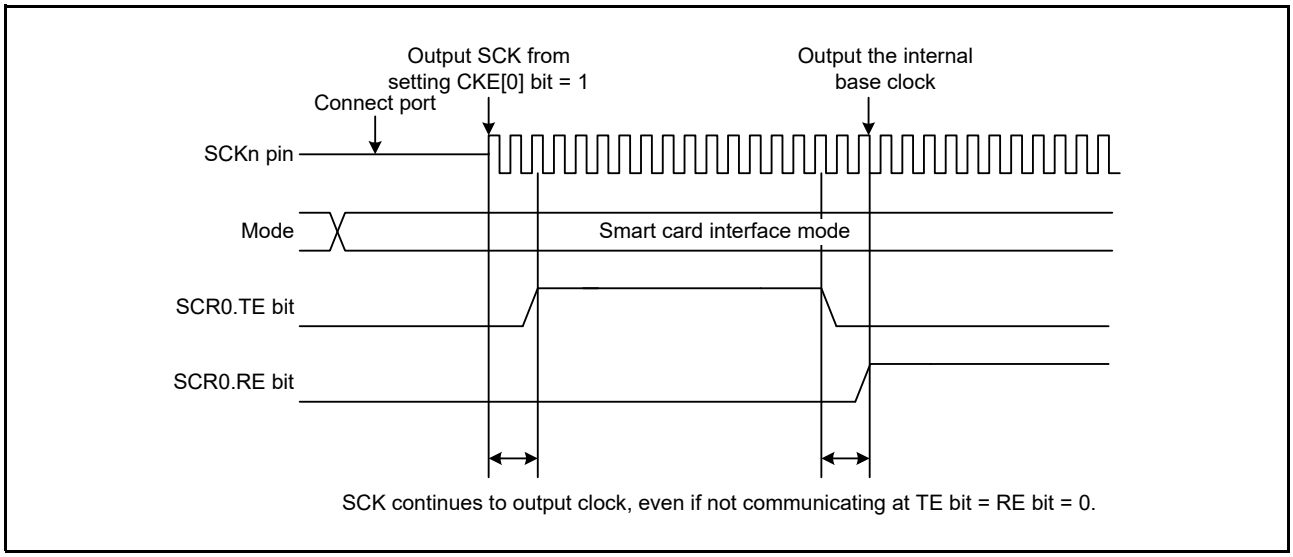
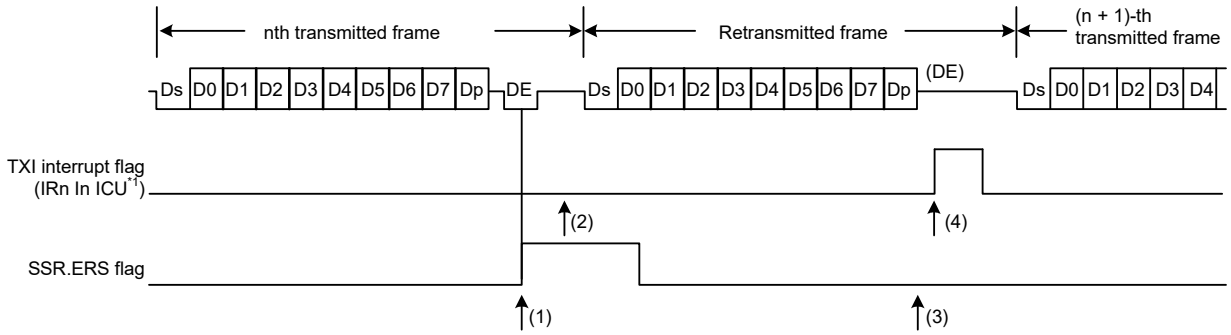


Figure 33.60 Example of Data Transmission Timing in Smart Card Interface Mode

33.7.6 Serial Data Transmission (Except in Block Transfer Mode)

Serial data transmission in smart card interface mode (except in block transfer mode), in that an error signal is sampled and data can be retransmitted, is different from that in non-smart card interface mode. Figure 33.61 shows the data retransmit operation during transmission.

(1) When an error signal from the receiver end is sampled after one-frame data has been transmitted, the ERS flag in the SSR register is set to 1. If the RIE bit in the SCR0 register is 1 at this time, an ERI interrupt request is generated. Clear the ERS flag to 0 before the next parity bit is sampled.
(2) For a frame in which an error signal is received, the TEND flag in the SSR register is not set. Data is retransferred from the TDR register to the TSR register allowing automatic data retransmission.
(3) If no error signal is returned from the receiver, the ERS flag is not set to 1.
(4) In this case, the RSCI judges that transmission of one-frame data (including retransmission) has been completed, and the TEND flag is set. If the TIE bit in the SCR0 register is 1 at this time, a TXI interrupt request is generated. Writing transmit data to the TDR register starts transmission of the next data.



Note 1. Refer to section 14, Interrupt Controller (ICUG) for details on the corresponding interrupt vector number.

Figure 33.61 Data Retransmit Operation in RSCI Transmit Mode

Figure 33.63 shows a sample flowchart of serial transmission. All the processing steps are automatically performed by using a TXI interrupt request to activate the DTC or DMAC. When SSR.TEND flag is set to 1 in transmission, if the SCR0.TIE bit is 1, a TXI interrupt request is generated. The DTC or DMAC is activated by a TXI interrupt request if the TXI interrupt request is specified as a source of DTC or DMAC activation beforehand, allowing transfer of transmit data. The TEND flag is automatically set to 0 when the DTC or DMAC transfers the data. If an error occurs, the RSCI automatically retransmits the same data. During this retransmission, the TEND flag is kept to 0 and the DTC or DMAC is not activated. Therefore, the RSCI and DTC or DMAC automatically transmit the specified number of bytes, including retransmission in the case of error occurrence. However, since the ERS flag is not automatically cleared, set the RIE bit to 1 beforehand to enable an ERI interrupt request to be generated at error occurrence, and clear the ERS flag to 0.
When transmitting/receiving data using the DTC or DMAC, be sure to make settings to enable the DTC or DMAC before making RSCI settings. For DTC or DMAC settings, refer to section 17, DMA Controller (DMACAa) and section 18, Data Transfer Controller (DTCb).
Note that SSR.TEND flag is set in different timings depending on the SCR3.GM bit setting. Figure 33.62 shows the TEND flag generation timing.

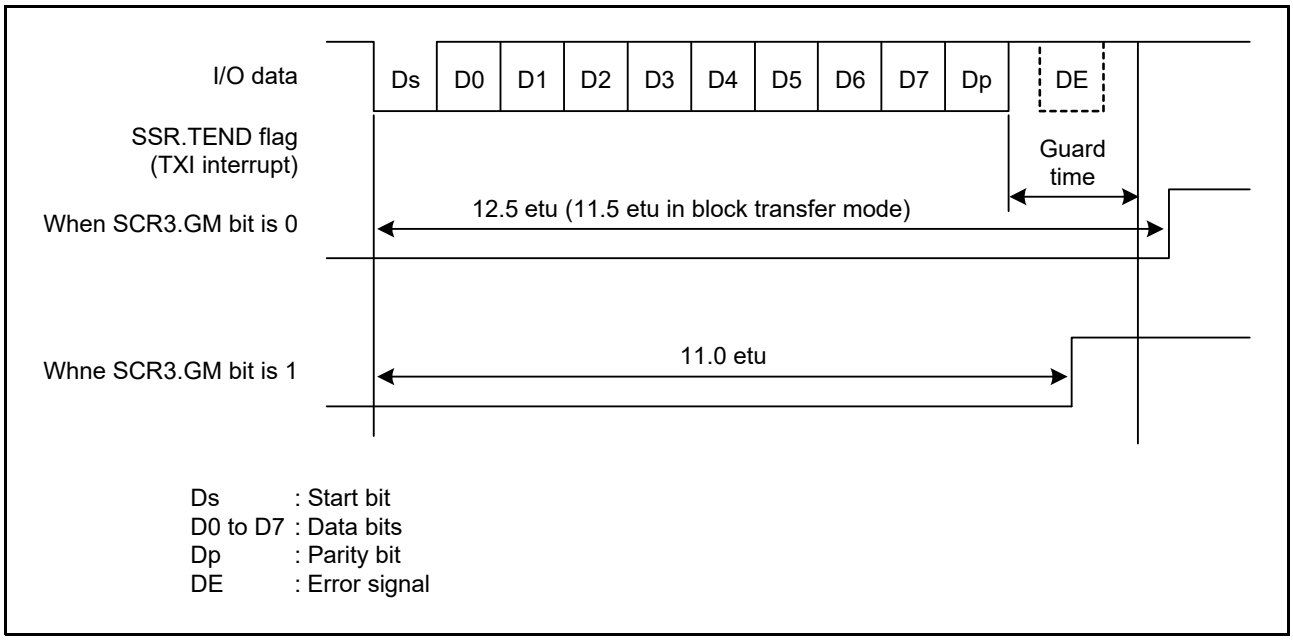


Figure 33.62 SSR.TEND Flag Generation Timing during Transmission

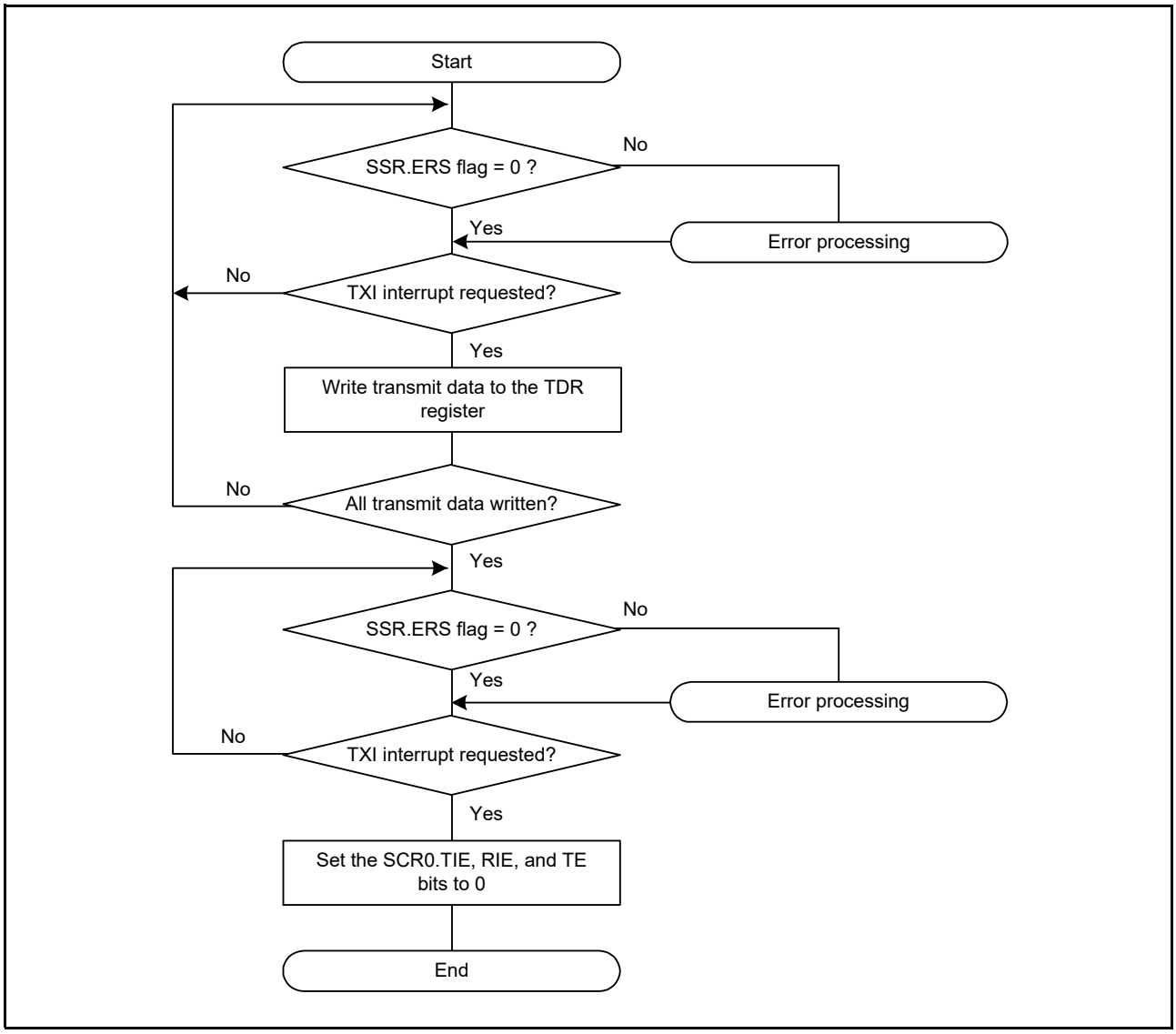
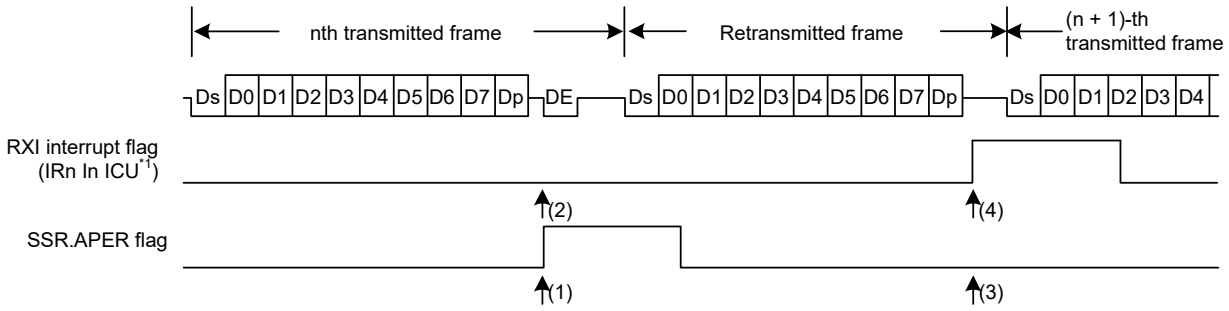


Figure 33.63 Sample Smart Card Interface Transmission Flowchart

33.7.7 Serial Data Reception (Except in Block Transfer Mode)

Serial data reception in smart card interface mode is similar to that in non-smart card interface mode. Figure 33.64 shows the data retransmit operation in receive mode.

(1) If a parity error is detected in receive data, the APER flag in the SSR register is set to 1. When the RIE bit in the SCR0 register is 1 at this time, an ERI interrupt request is generated. Clear the APER flag to 0 before the next parity bit is sampled.
(2) For a frame in which a parity error is detected, no RXI interrupt is generated.
(3) When no parity error is detected, the APER flag in the SSR register is not set to 1.
(4) In this case, data is determined to have been received successfully. When the RIE bit in the SCR0 register is 1, an RXI interrupt request is generated.



Note 1. Refer to section 14, Interrupt Controller (ICUG) for details on the corresponding interrupt vector number.

Figure 33.64 Data Retransmit Operation in RSCI Receive Mode (Data Retransmit Operation during Reception)

Figure 33.65 shows a sample flowchart for serial data reception. All the processing steps are automatically performed using an RXI interrupt request to activate the DTC or DMAC. In reception, setting the RIE bit to 1 allows an RXI interrupt request to be generated. The DTC or DMAC is activated by an RXI interrupt request if the RXI interrupt request is specified as a source of DTC or DMAC activation beforehand, allowing transfer of receive data. If an error occurs during reception and either the SSR.ORER or APER flag is set to 1, a receive error interrupt (ERI) request is generated. Clear the error flag after the error occurrence. If an error occurs, the DTC or DMAC is not activated and receive data is skipped. Therefore, the number of bytes of receive data specified in the DTC or DMAC is transferred. Even if a parity error occurs and the APER flag is set to 1 during reception, receive data is transferred to RDR register, thus allowing the data to be read. When a reception is forcibly terminated by setting the SCR0.RE bit to 0 during operation, read the RDR register because the received data which has not yet been read may be left in RDR register.

Note: For operations in block transfer mode, refer to section 33.3, Operation in Asynchronous Mode.

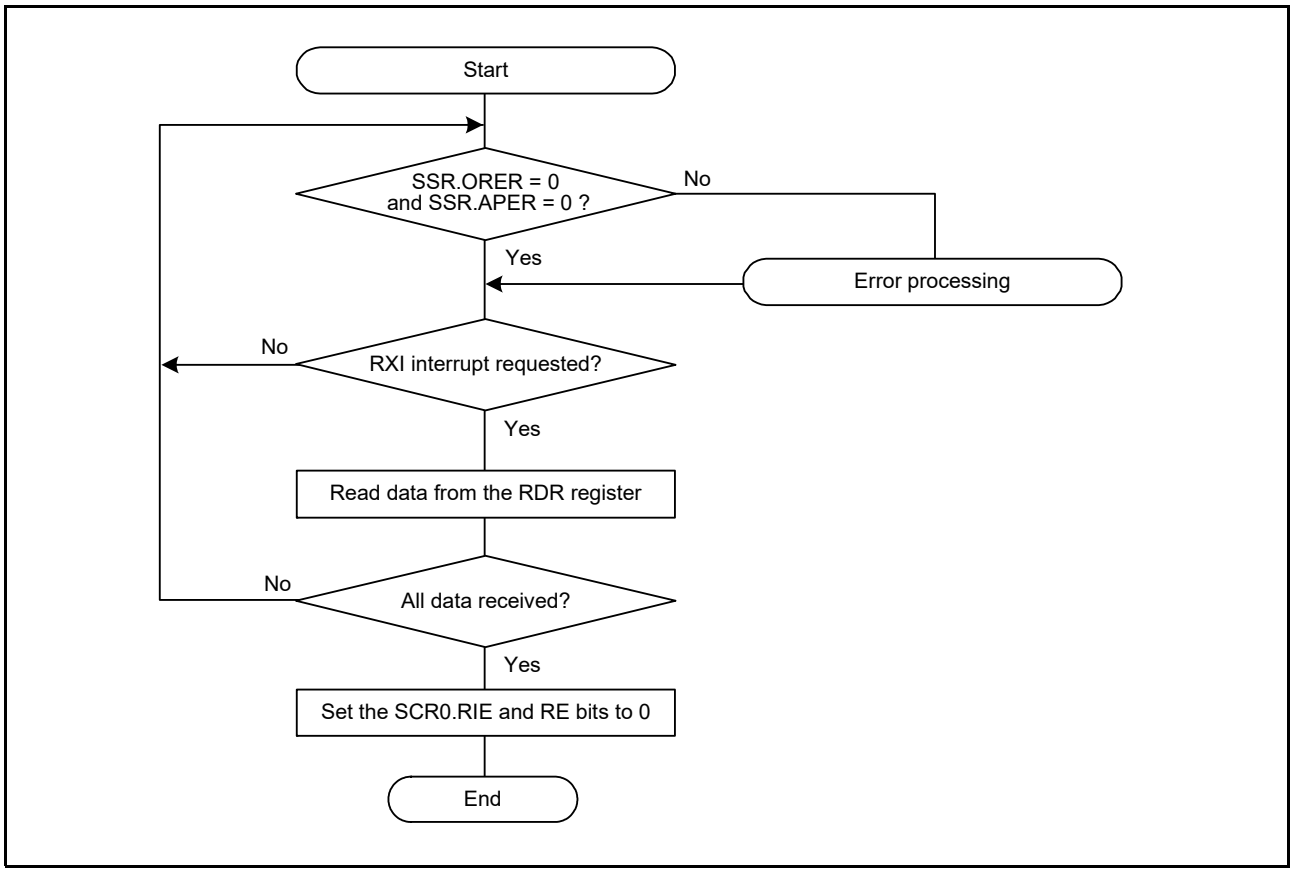


Figure 33.65 Sample Smart Card Interface Reception Flowchart

33.7.8 Clock Output Control

When the SCR3.GM bit is set to 1, the clock output can be controlled by the SCR3.CKE[1:0] bits. Refer to the description of the SCR3.CKE[1:0] bits in section 33.2.8, Control Register 3 (SCR3). When setting the clock output, the base clock described in section 33.7.4, Receive Data Sampling Timing and Reception Margin is output, so the width of the clock pulse can be kept to the width specified by setting the bit rate. It is described in section 33.2.7, Control Register 2 (SCR2), the bit rate is set by SCR2.CKS[1:0] bits, SCR2.BCP[2:0] bits and BRR[7:0] bits.
Figure 33.66 shows the timing chart for explaining clock output control. This is an example when the SCR3.CKE[1] bit is set to 0 and the SCR3.CKE[0] bit is controlled.
When the SCR3.GM bit is 0, output control by the SCR3.CKE[0] bit is immediately reflected on the SCKn pin, so there is a possibility that pulses with an unintended width may be output from the SCKn pin.
When the SCR3.GM bit is 1, the output pulse control by the SCR3.CKE[0] bit controls the pulse width set to be based on the state of the base clock.

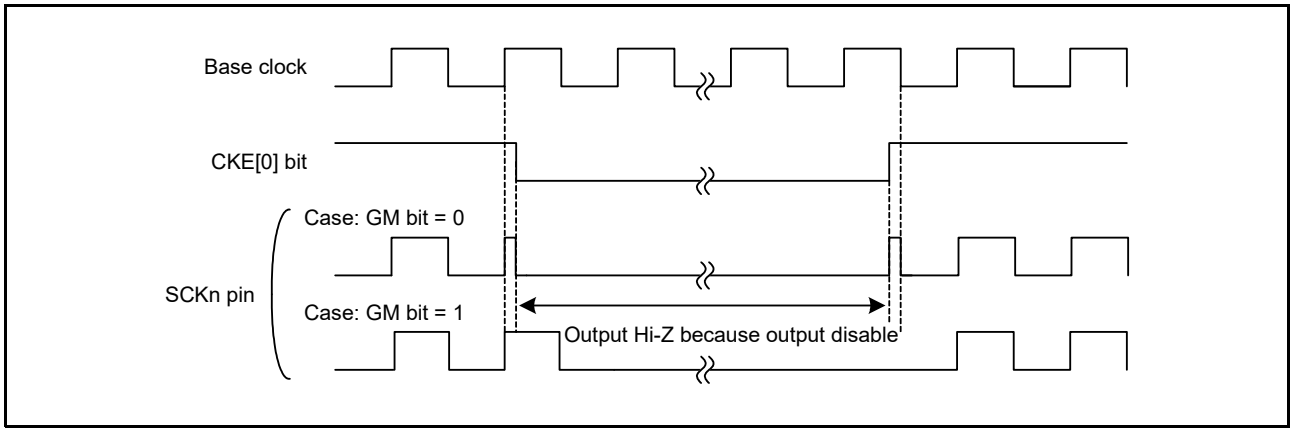


Figure 33.66 Clock Control Timing Chart by SCR3.GM Bit

33.8 Extended Serial Mode

33.8.1 Serial Transfer Protocol

As an extended function of the RSCI, the RSCI supports the serial communication protocol (Figure 33.67) consisting of a Start Frame and an Information Frame. Extended serial mode is enabled by the SCR3.MOD[2:0] bits = 110b. Since the extended serial mode uses the same circuit as the asynchronous mode for transmission/reception control other than Break Field, the basic communication settings required for the asynchronous mode are also required for the extended serial mode (however, set the SCR3.RXDESEL bit to 1).
The Start Frame consists of a Break Field, Control Field 0, and Control Field 1. The Information Frame can be configured with some Data Fields, a CRC16 Upper Field, and a CRC16 Lower Field.

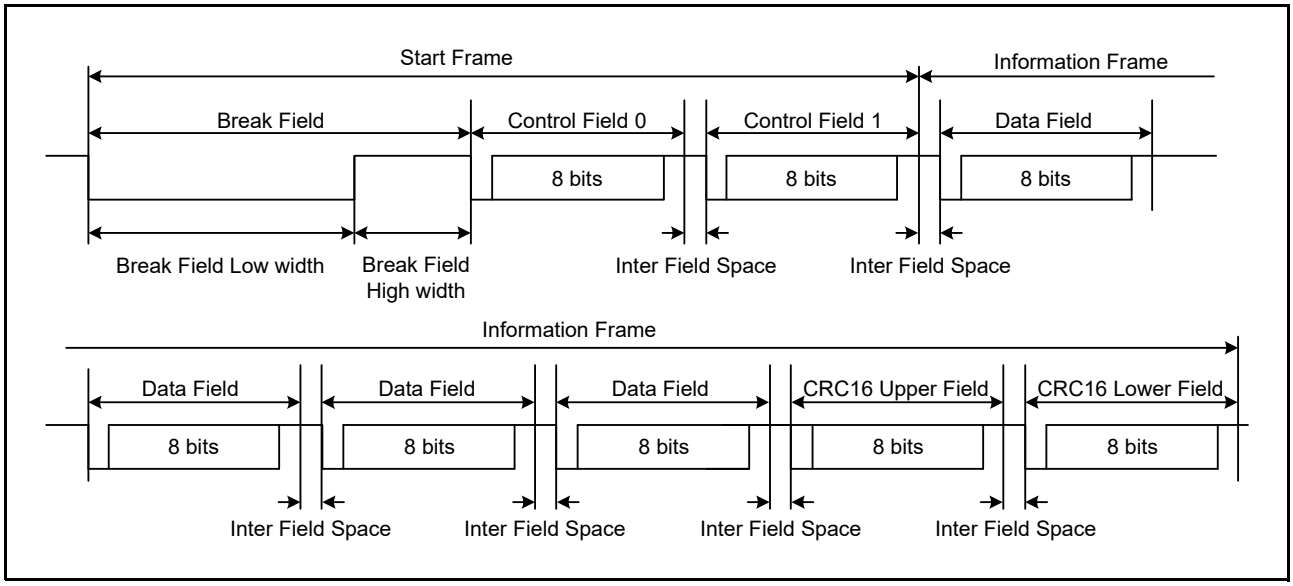


Figure 33.67 Protocol for Serial Transfer by the Extended Serial Mode

The following describes operations in extended serial mode. In this section, operations are described with the following conditions:
Communication pin (RXDn/TXDn) level inversion function: OFF (RINV bit = TINV bit = 0)
When using the communication pin (RXDn/TXDn) level inversion function enabled, replace the RXD and TXD signal levels with their inverted levels.

33.8.2 Transmitting a Start Frame

Figure 33.68 shows an example of transmission of the Start Frame consisting of a Break Field, Control Field 0, and Control Filed 1. (Omit Break Field and Control Field 0 according to the Start Frame configuration.)
Figure 33.69 shows a flowchart for Start Frame transmission.
The RSCI operates as follows during Start Frame transmission.

(1) Make the initial settings for the RSCI according to the RSCI initialization flow (Figure 33.8) in asynchronous mode. In extended serial mode, do not set SCR0.TE and TIE bits to 1 at the same time to avoid TXI output before the Break Field. Therefore, perform the following two steps sequentially to set the RSCI initialization flow (asynchronous mode) procedure [10].

- Set the bits except SCR0.TIE bit. (SCR0.TIE bit = 0, SCR0.TE bit = 1, and SCR0.RE bit = 0)
- Set SCR0.TIE bit to 1.

(2) When 1 is written to TCST, the timer in the extended serial module starts counting and outputs a low level (Break Field) from the TXDn pin for the period set in XCR2.BFLW[15:0] bits. A timer count clock source can be selected by XCR0.TCSS[1:0] bits.
Writing 0 to XCR1.TCST bit suspends output of the Break Field. After the suspension, set SCR0.TE bit = 0 and turn off the transmission.

(3) When the extended serial module timer count value matches the set XCR2.BFLW[15:0] bits value, the timer stops counting and inverts the TXDn pin output level, and the XSR0.BFOF flag is set to 1*1. Furthermore, if XCR0.BFOIE bit has been set to 1 at this time, a TXI interrupt is generated.

(4) After confirming the BFOF flag is set to 1, send the Control Field 0 data.*2

(5) After the Control Field 0 data has been transmitted, write the Control Field 1 data to TDR. And it is transmitted.

(6) After the Control Field 1 data has been transmitted, the Information Frame data is transmitted.

Note 1. After XSR0.BFOF flag is set to 1, if 1 is written to XCR1.TCST bit without clearing it, no TXI interrupt is output at the end of Break Field transmission. Clear XSR0.BFOF flag before writing 1 to XCR1.TCST bit.

Note 2. LIN communication requires a Break delimiter (IDLE period) of 1 bit or more from the end of Break Field transmission until the next data transmission starts. For this reason, the Break delimiter length is counted upon completion of Break Field transmission. If transmit data is written while the Break delimiter length is being counted, transmission does not start until the Break delimiter length counting is completed. When transmit data is written after the Break delimiter length has been counted, transmission starts at the same timing as normal data transmission.
Break delimiter length count time after Break Field transmission: 1-bit to 2-bit length (SCR3.STOP bit = 0) 2-bit to 3-bit length (SCR3.STOP bit = 1)

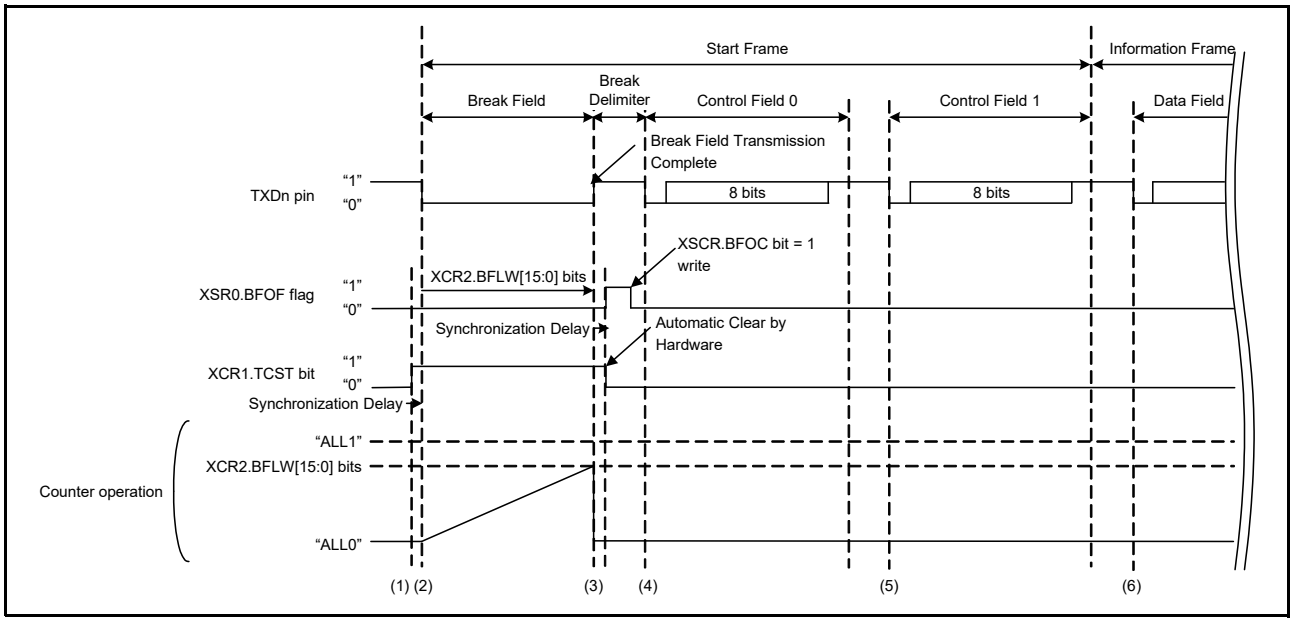


Figure 33.68 Example of Operations When Transmitting a Start Frame

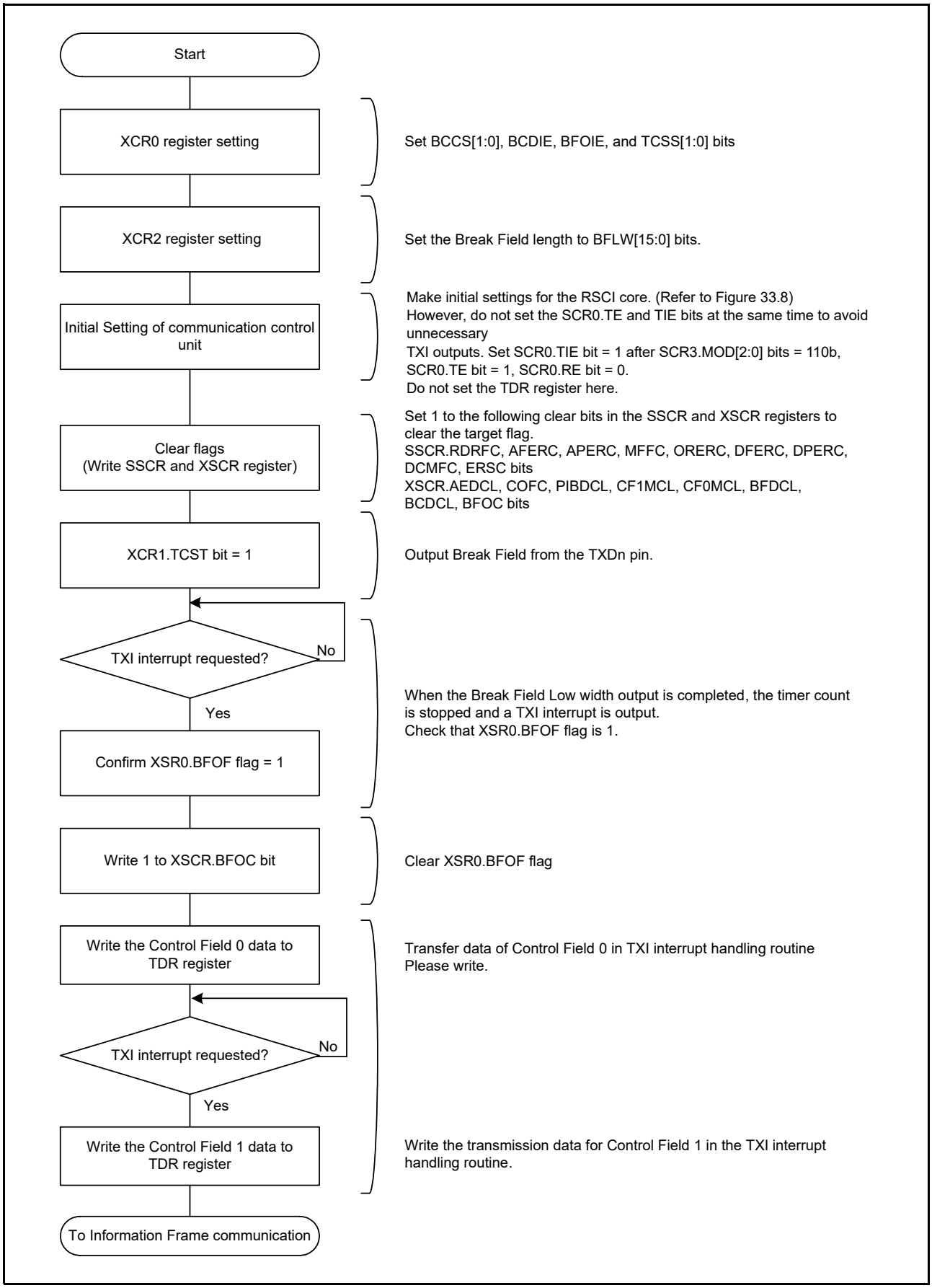


Figure 33.69 Example of Start Frame Transmission

33.8.3 Receiving a Start Frame

The extended serial mode control section is capable of receiving Start Frames with the structures listed in Table 33.40.

Table 33.40 Structures of Start Frames

XCR0		Start Frame Configuration
BFE	CF0RE	
0	0	Control Field1 (8 bits) → Information Frame
0	1	Control Field0 (8 bits) → Control Field1 (8 bits) → Information Frame
1	0	Break Field (Low width) → Control Field1 (8 bits) → Information Frame
1	1	Break Field (Low width) → Control Field0 (8 bits) → Control Field1 (8 bits) → Information Frame

33.8.3.1 Extended Serial Normal Reception of Start Frame (PIB not Used)

Figure 33.70 shows an example of normal reception of the Start Frame consisting of a Break Field, Control Field0, and Control Field1. Figure 33.71 shows an example of reception to detect the Break Field during Control Field 1. Figure 33.72 shows a flowchart to receive the Start Frame, and Figure 33.73 shows a state transition diagram.
When receiving the Start Frame, the RSCI operates as follows. Omit the processing of Break Field and Control Field0 according to the Start Frame configuration.

(1) Writing 1 to XCR1.SDST bit makes it possible to detect the Start Frame. When XCR0.BFE bit = 1, RXD input to the RSCI core is disabled until the Break Field is detected (because XSR0.RXDSF flag is set to 1). Once the Break Field is detected, RXD input can be received to the RSCI core (XSR0.RXDSF flag = 0).
(2) When a low level is input from the RXDn pin, the Break Field detection count starts. A timer count clock source can be selected by XCR0.TCSS[1:0] bits.
(3) When a low level that is equal to or longer than the period set in XCR2.BFLW[15:0] bits is input from the RXDn pin, it is determined as Break Field. At this time, XSR0.BFDF flag is set to 1. If XCR0.BFDIE bit has been set to 1 at this time, a BFD interrupt is generated.
The timer continues counting until the RXD rising edge or counter overflow.
(4) After the Break Field is detected, when the input level from the RXDn pin becomes high, the count value is captured to XSR1.CCV[15:0] bits when XCR1.BRME bit = 0. At this time, XSR0.RXDSF flag is cleared to 0 and the RSCI core starts receiving the RXD input.
(5) The RSCI core starts receiving Control Field 0. Because the extended serial continuously counts the edge interval, it determines a low level that is equal to or longer than the period set in XCR2.BFLW[15:0] bits as detection of the Break Field. When the Break Field is detected in the Control Field 0 phase, the RSCI core waits for reception of Control Field 0 again (Figure 33.71).
(6) When Control Field 0 has been received, an RXI interrupt is generated and the Control Field 0 data is stored in XSR0.CF0RD[7:0] bits. When the received data matches the set XCR2.CF0D[7:0] bits value, XSR0.CF0MF flag is set to 1. If the received data differs from the set XCR2.CF0D[7:0] bits value, the RSCI transitions to the state before the Break Field is detected.
(7) The RSCI core starts receiving Control Field 1. When BFE bit = 1, the Break Field detection function is continuously enabled while SDST bit = 1 as in the case of Control Field 0. When the Break Field is detected in the Control Field 1 phase, the RSCI core waits for reception of Control Field 0 again.
(8) When Control Field 1 has been received, an RXI interrupt is generated and the Control Field 1 data is stored in XSR0.CF1RD[7:0] bits. When the received data matches the set XCR1.PCF1D[7:0] bits value or the set

XCR1.SCF1D[7:0] bits value, XSR0.CF1MF flag is set to 1. If the received Control Field 1 data matches neither the set XCR1.PCF1D[7:0] bits value nor the set XCR1.SCF1D[7:0] bits value, the RSCI transitions to the state before the Break Field is detected.

(9) The RSCI core performs Information Frame communication.

(10) When communication is completed, write 0 to XCR1.SDST bit and 0 to SCR0.RE bit to stop reception.

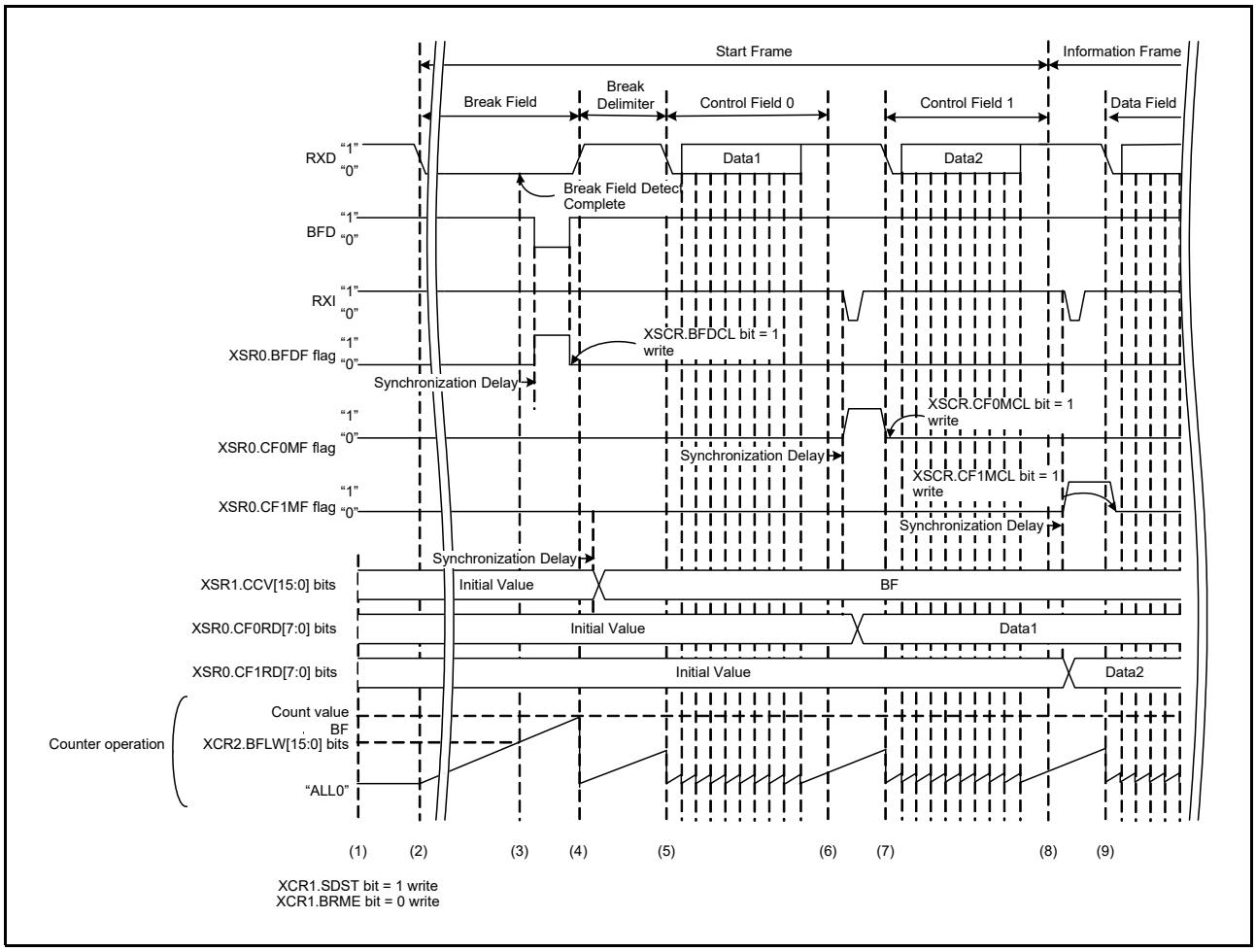


Figure 33.70 Normal Reception Example of Start Frame (PIB Not Used)

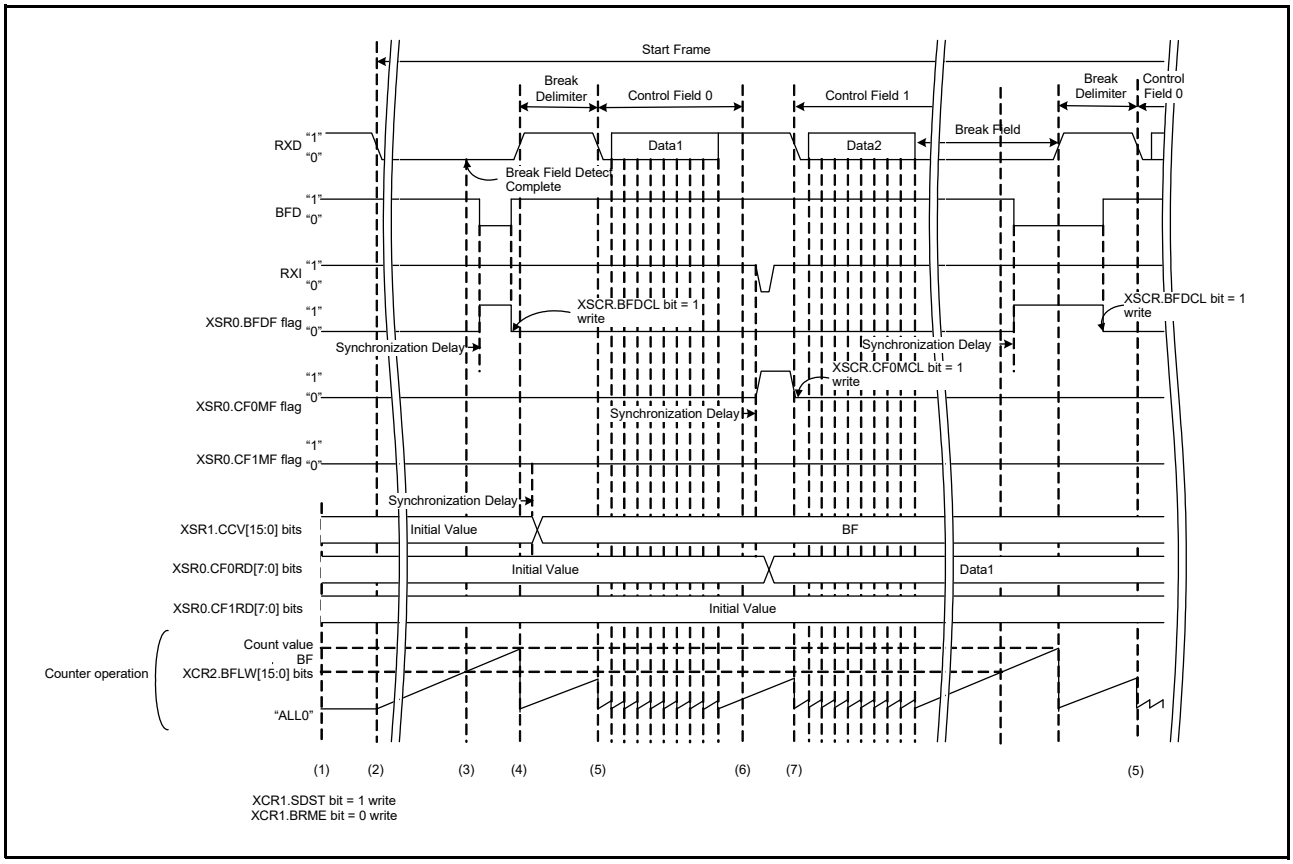


Figure 33.71 Start Frame Reception Example (PIB Not Used) Break Field Detected during Control Field 1

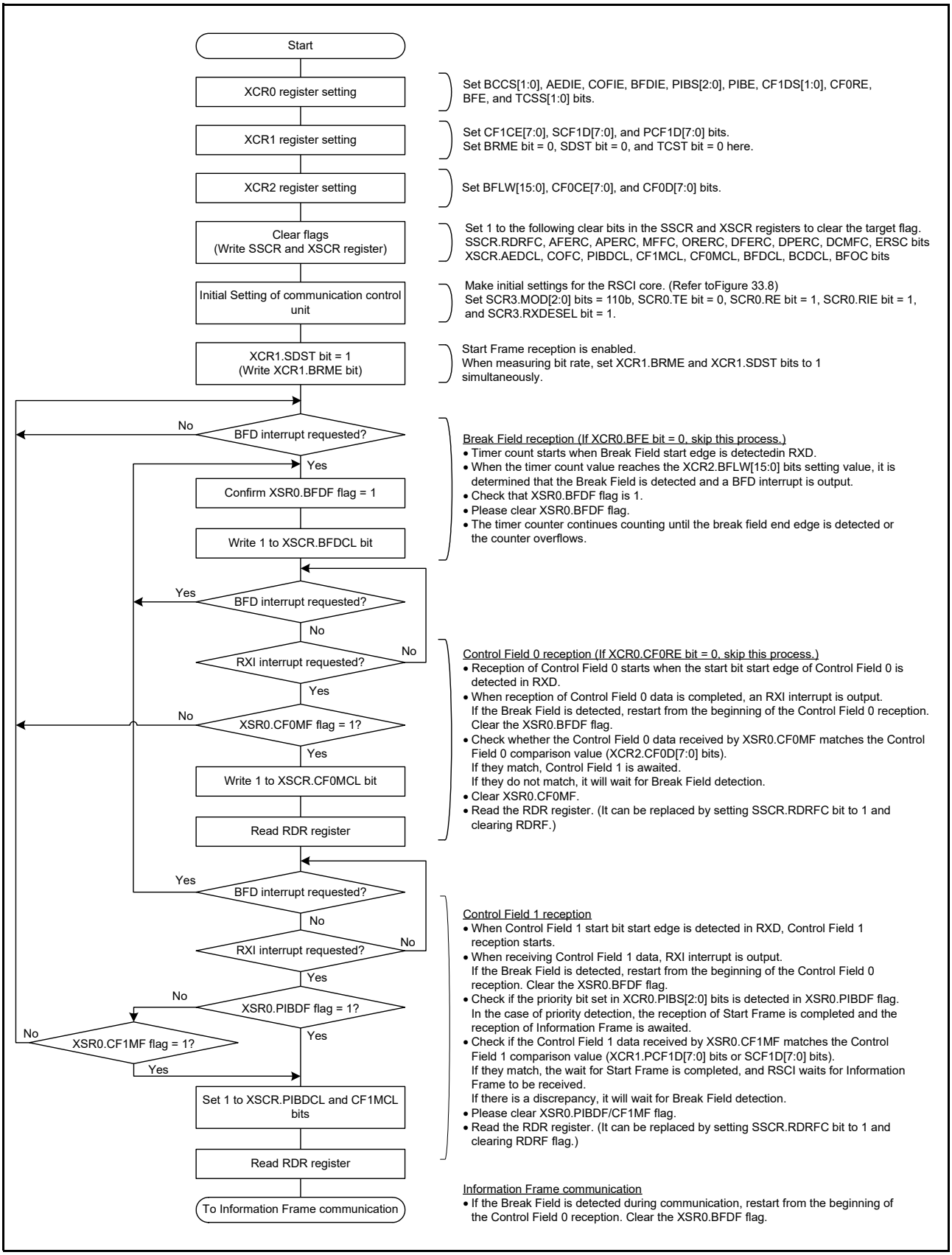


Figure 33.72 Sample Flowchart for Reception of a Start Frame

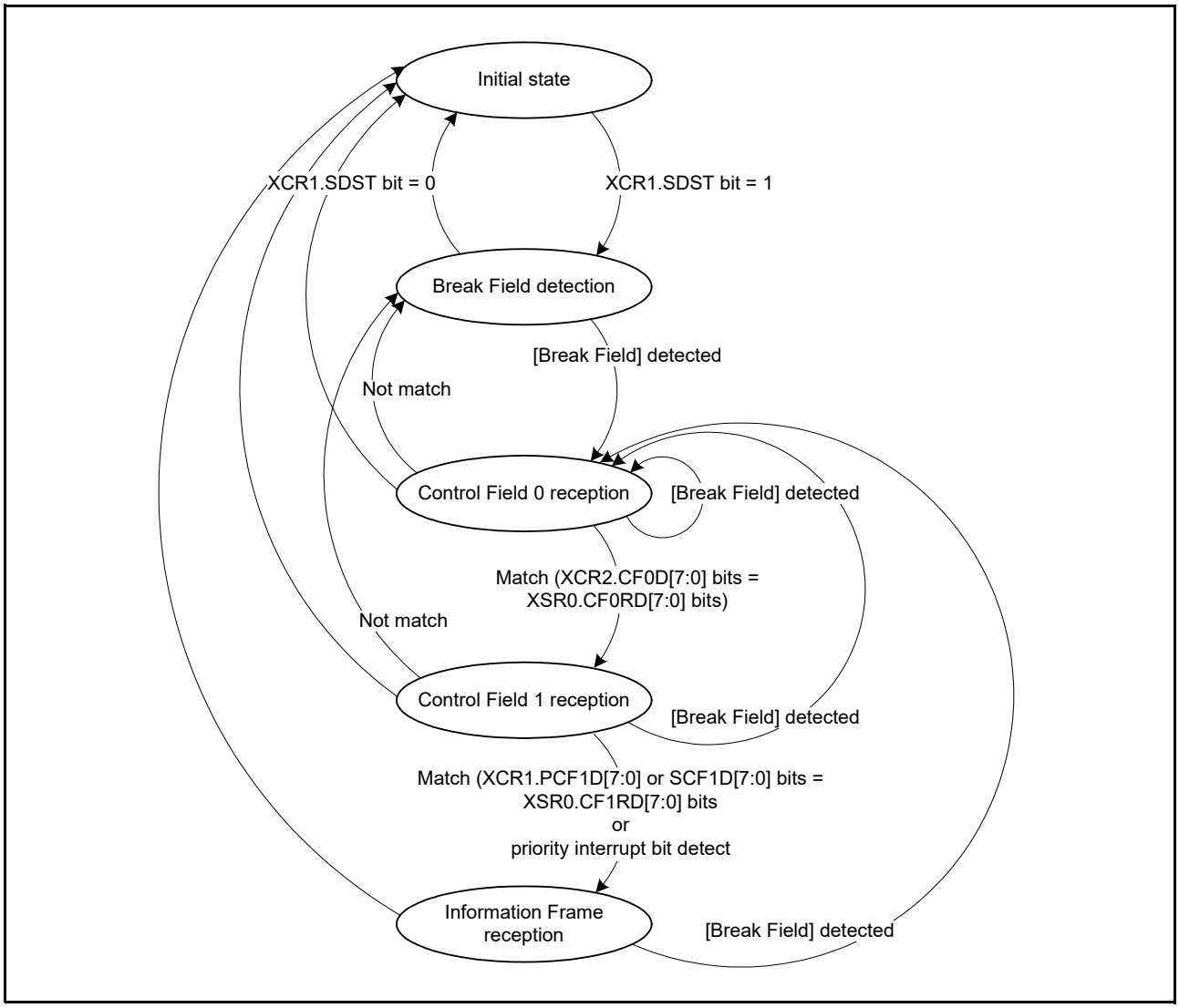


Figure 33.73 State Transition Diagram of Start Frame Reception

33.8.3.2 Priority Interrupt Bit

Figure 33.74 shows an example of Start Frame reception using the priority interrupt bit. The priority interrupt bit is enabled by setting XCR0.PIBE bit to 1.

The RSCI operates as follows during Start Frame reception using the priority interrupt bit.

Steps (1) to (7) are the same as in Figure 33.70, for Start Frame reception.

(8) When the value specified in the XCR0.PIBS[2:0] bits matches the set XCR1.PCF1D[7:0] bits value, XSR0.PIBDF flag is set to 1and the RSCI core performs communication of the Information Frame. If the data received in Control Field 1 matches neither the set XCR1.PCF1D[7:0] bits value nor the set XCR1.SCF1D[7:0] bits value and the priority interrupt bit is not detected, the RSCI transitions to the state before the Break Field is detected.

(9) The RSCI core performs Information Frame communication.

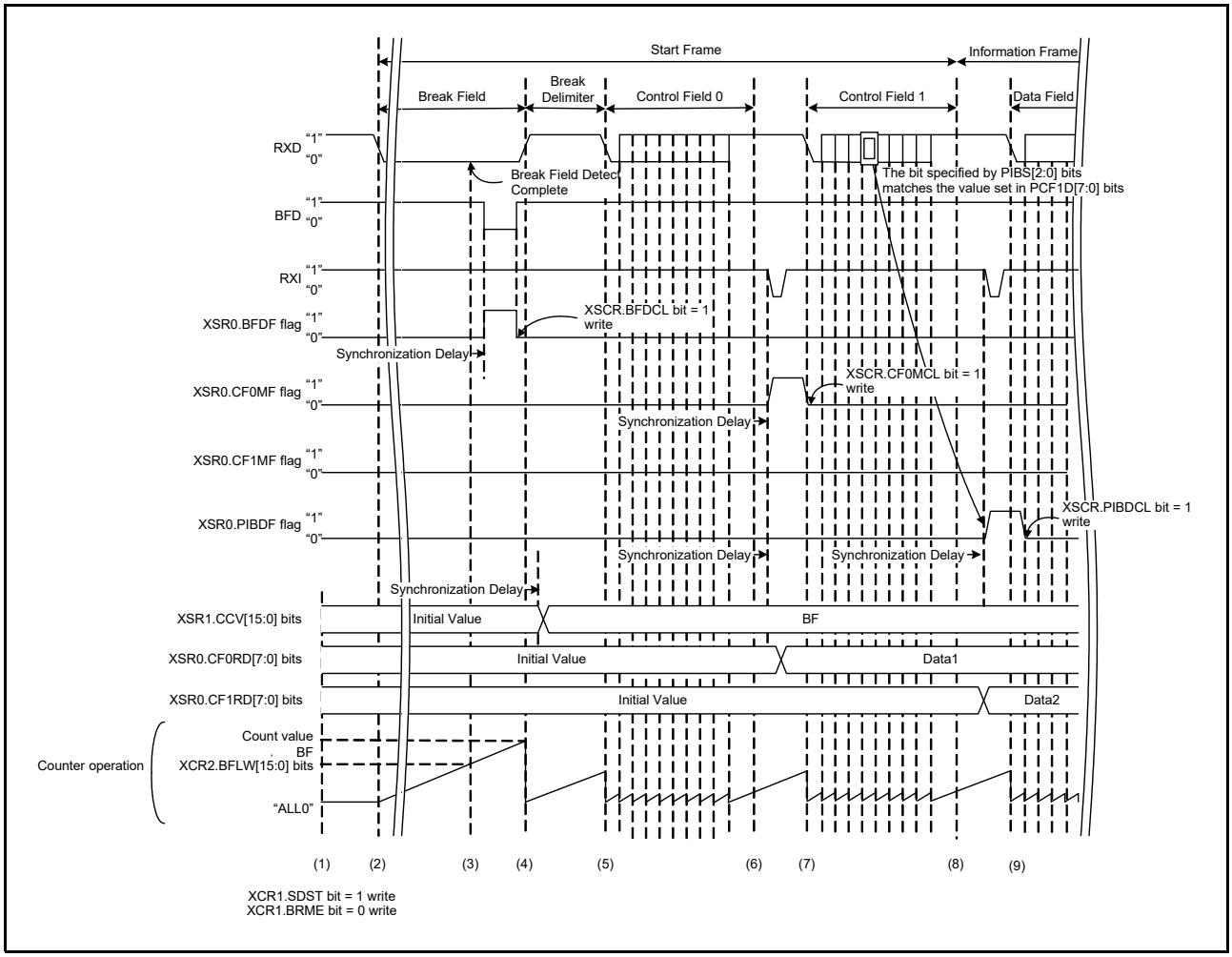


Figure 33.74 Start Frame Reception Example (Priority Interrupt Bit Used)

33.8.4 Detection of Bus Collisions

In extended serial mode (SCR3MOD[2:0] bits = 110b) when TE bit = 1, the bus conflict detection function works during Break Field transmission and during data transmission.

Figure 33.75 shows an operation example of the bus conflict detection function. The TXDn pin output and the RXDn pin input are sampled by the bus conflict detection clock set in XCR0.BCCS[1:0] bits. When a mismatch occurs three times in a row, XSR0.BCDF flag is set to 1, and if XCR0.BCDIE bit has been set to 1 at this time, an ERI interrupt is generated.

When an ERI interrupt is generated, stop transmission according to Figure 33.76. Check the bus state to decide whether to resume transmission.

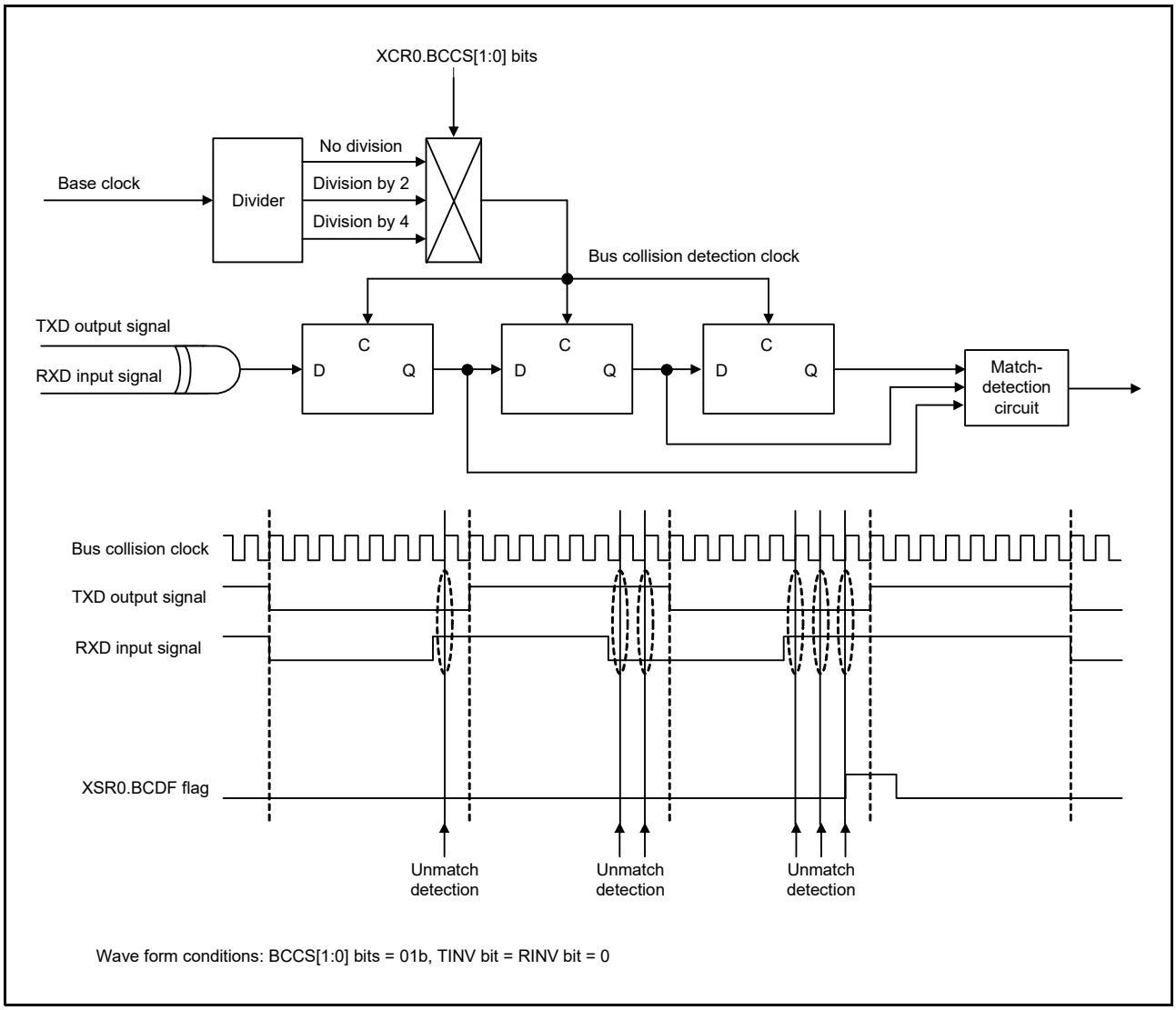


Figure 33.75 Example of Operations with Bus Collision Detection

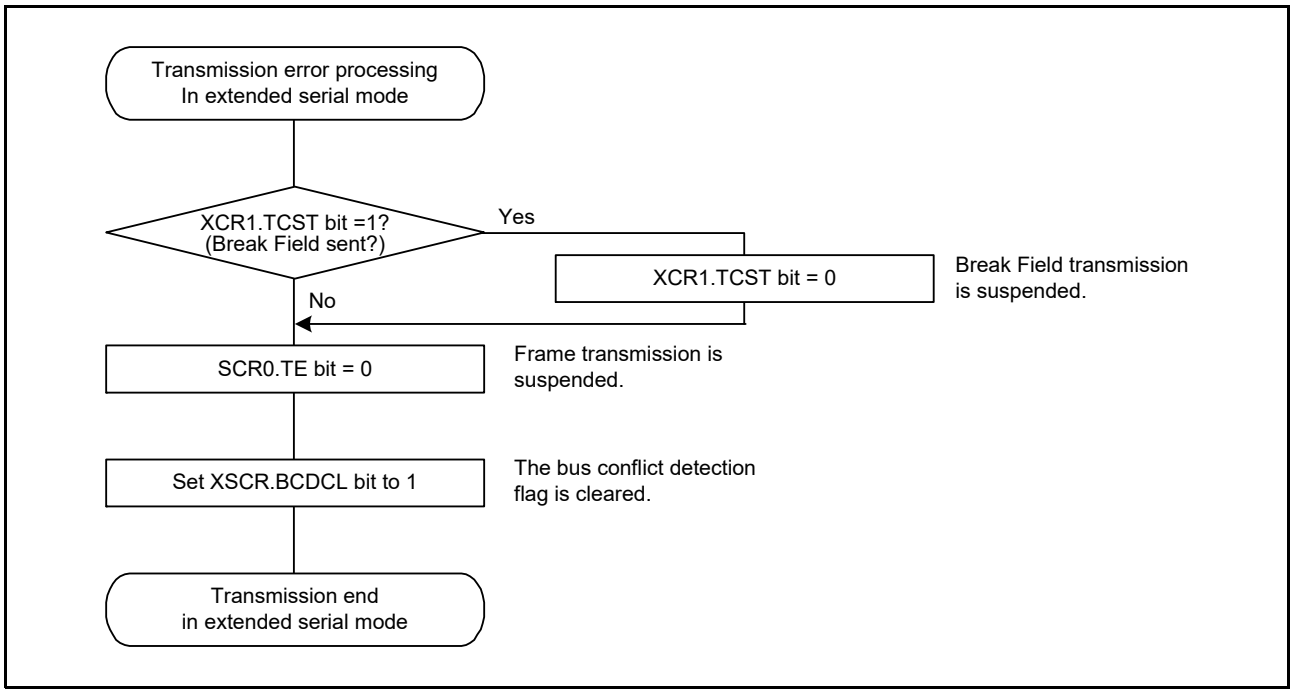


Figure 33.76 ERI Interrupt Handling Flow at Transmission in Extended Serial Mode

33.8.5 Bit Rate Measurement

This function measures a bit rate between the effective edges of the input signal from the RXDn pin. Figure 33.77 shows an operation example of the bit rate measurement function.

(1) Writing 1 to XCR1.SDST and XCR1.BRME bits enables bit rate measurement. When this bit is set to 1, the valid edge interval of Control Field 0 and Control Field 1 data is measured. However, bit rate is not measured between the Break Field and the Break Delimiter. Set XCR1.BRME and XCR1.SDST bits to 1 simultaneously, only when measuring bit rate.
(2) Because bit rate is not measured in the Break Field, the effective edge detection flag is not set to 1 at the rising edge at the end of the Break Field, and the counter capture value is not stored in XSR1.CCV[15:0] bits.
(3) The counter starts counting from the falling edge of the start bit in Control Field 0. The Break Delimiter count value is not captured in XSR1.CCV[15:0] bits.
(4) The rising edge of the start bit is detected as an effective edge, and then the XSR0.AEDF flag is set to 1. If XCR0.AEDIE bit has been set to 1 at this time, an AED interrupt is output. The start bit count value is stored in XSR1.CCV[15:0] bits. The XSR1.CCV[15:0] value is retained until the effective capture value is read.
(5) Even if an effective edge is input from the RXD input pin, the count value of this effective edge timing is not captured because the XSR1.CCV[15:0] bits value has not been read and retention has not been released. In this case, an AED interrupt is not output.
(6) The XSR1.CCV[15:0] value is read. Then the retention of XSR1.CCV[15:0] bits is released and the XSR0.AEDF flag is cleared by hardware.
(7) Because the retention of XSR1.CCV[15:0] bits has been released, the count value is captured at the effective edge and is retained. At the same time, the XSR0.AEDF flag is set to 1, and if XCR0.AEDIE bit has been set to 1, an AED interrupt is output. The bit rate can be adjusted by calculating it from the count value between effective edges by software and by changing the RSCI settings.
(8) To disable bit rate measurement, write 0 to XCR1.BRME bit.
(9) The XSR0.AEDF value and the XSR1.CCV[15:0] value remain unchanged at the effective edge timing because the bit rate measurement function is disabled.

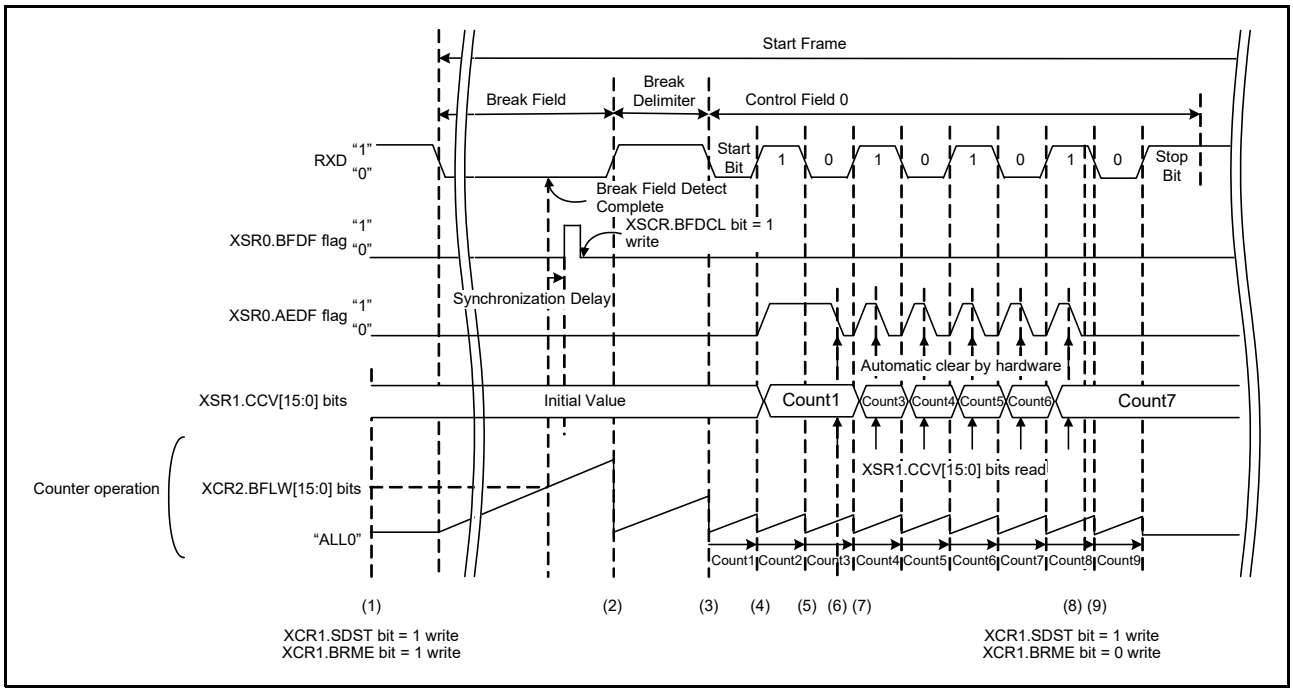


Figure 33.77 Operation Example of the Bit Rate Measurement Function

33.9 Operation in Simple I2C Mode

Simple I2C-bus format is composed of 8 data bits and an acknowledge bit. The frame following the start condition or restart condition is the slave-address frame, which is used by the master device to specify a slave device with which it is communicating. The currently specified slave device remains valid until a new slave device is specified or a stop condition is satisfied. The 8-bit data of each frame is transmitted from MSB.
Figure 33.78 shows I2C bus format, and Figure 33.79 shows the timing of I2C.

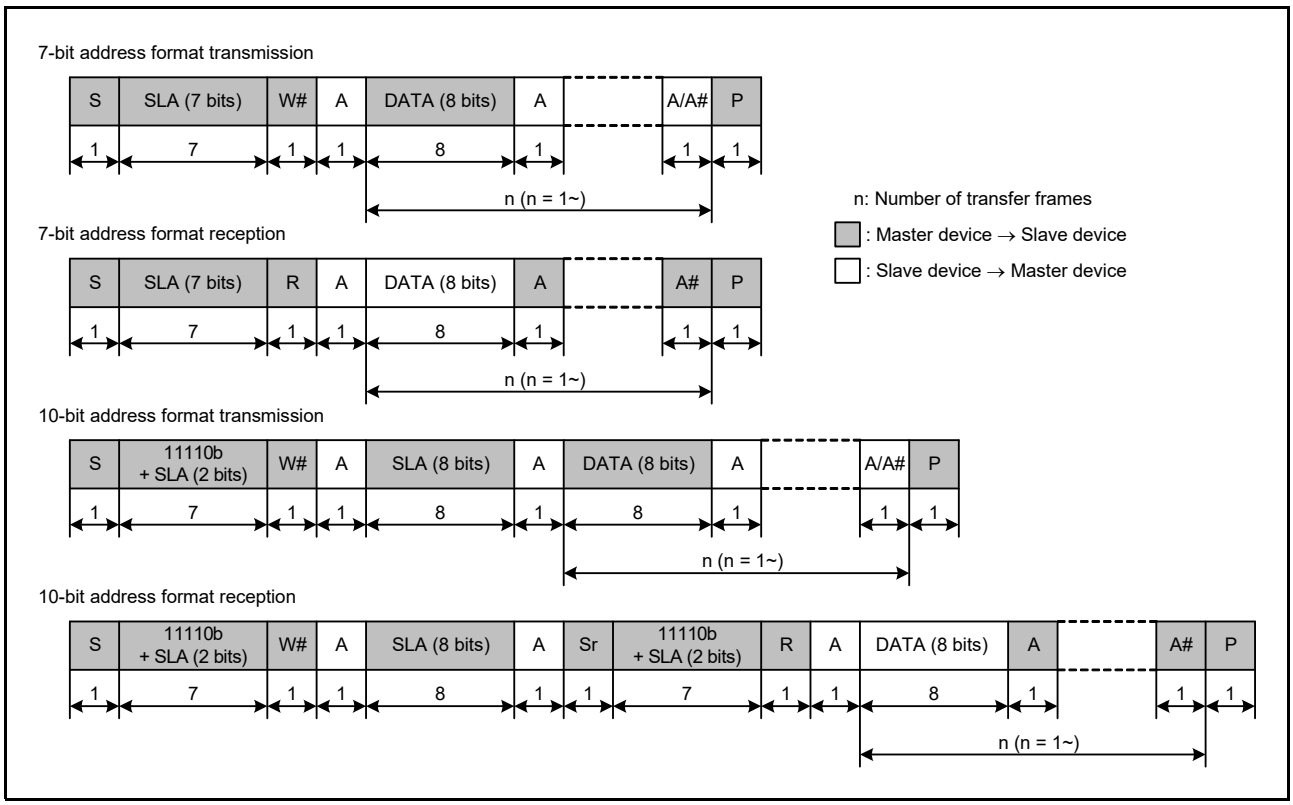


Figure 33.78 I2C-Bus Format

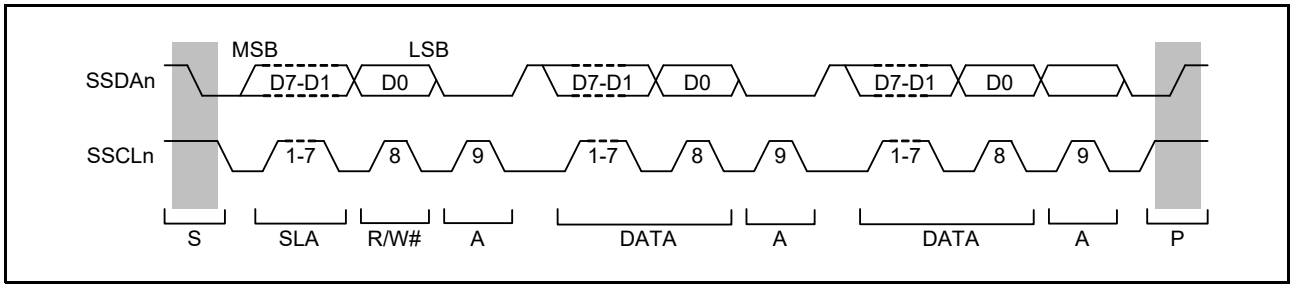


Figure 33.79 I2C-Bus Timing (When SLA is 7 Bits)

S: Indicates a start condition. The master device changing the level on the SSDAn line from the high to the low level when the SSCLn line is at the high level.
SLA: Indicates a slave address. The master device selects a slave device.
R/W#: Indicates the direction of transfer (reception or transmission). When R/W# is high, the transfer direction is from the slave device to the master device. When R/W# is low, the transfer direction is from the master device to the slave device.
A/A#: Indicates an acknowledge. This is returned by the slave device for master transmission and by the master device for master reception. The low level indicates ACK and the high level indicates NACK.
Sr: Indicates a restart condition. The master device changing the level on the SSDAn line from the high to the low level when the SSCLn line is at the high level.
DATA: Indicates the received or transmitted data.
P: Indicates a stop condition. The master device changing the level on the SSDAn line from the low to the high level when the SSCLn line is at the high level.

33.9.1 Generation of Start, Restart, and Stop Conditions

Writing 1 to SIMR.IICSTAREQ bit causes the generation of a start condition. The following operations are done at the generation of a start condition.

- The level on the SSDAn line falls (from the high level to the low level) and the SSCLn line is kept in the released state.
- The hold time for the start condition is secured as half of a bit period at the bit rate determined by the setting of the SCR2.BRR[7:0] bits.
- The level on the SSCLn line falls (from the high level to the low level), the SIMR.IICSTAREQ bit is set (to 0), and a start-condition generated interrupt is output.

Writing 1 to SIMR.IICRSTAREQ bit causes the generation of a restart condition. The following operations are done at the generation of a restart condition.

- The SSDAn line is released and the SSCLn line is kept at the low level.
- The period at low level for the SSCLn line is secured as half of a bit period at the bit rate determined by the setting of the SCR2.BRR[7:0] bits.
- The SSCLn line is released (transition from the low to the high level).
- Once the high level on the SSCLn line is detected, the setup time for the restart condition is secured as half of a bit period at the bit rate determined by the setting of the SCR2.BRR[7:0] bits.
- The level on the SSDAn line falls (from the high level to the low level).
- The hold time for the restart condition is secured as half of a bit period at the bit rate determined by the setting of the SCR2.BRR[7:0] bits.
- The level on the SSCLn line falls (from the high level to the low level), the SIMR.IICRSTAREQ bit is set (to 0), and a restart-condition generated interrupt is output.

Writing 1 to SIMR.IICSTPREQ bit causes the generation of a stop condition. The following operations are done at the generation of a stop condition.

- The level on the SSDAn line falls (from the high level to the low level) and the SSCLn line is kept at the low level.
- The period at low level for the SSCLn line is secured as half of a bit period at the bit rate determined by the setting of the SCR2.BRR[7:0] bits.
- The SSCLn line is released (transition from the low to the high level).
- Once the high level on the SSCLn line is detected, the setup time for the stop condition is secured as half of a bit period at the bit rate determined by the setting of the SCR2.BRR[7:0] bits.
- The SSDAn is released (transition from the low to the high level), the SIMR.IICSTPREQ bit is set (to 0), and a stop condition generated interrupt is output.

Figure 33.80 shows the timing of operations in the generation of start, restart and stop conditions.

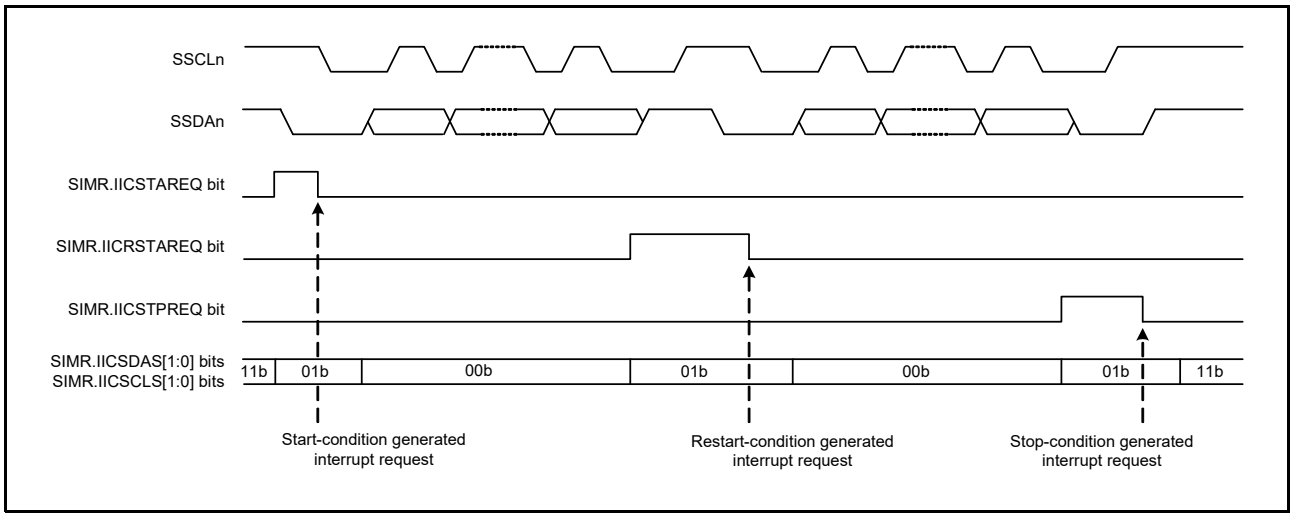


Figure 33.80 Timing of Operations in the Generation of Start, Restart, and Stop Conditions

33.9.2 Clock Synchronization

The slave device of the communication partner may make SSCLn line Low-level with a view to insert a wait. Setting the SIMR.IICCSC bit to 1, applies control to obtain synchronization when the levels of the internal SCL signal and the level being input on the SSCLn pin differ.

When the SIMR.IICCSC bit is set to 1, the level of the internal SCL signal changes from low to high, counting to determine the period at high level is stopped while the low level is being input on the SSCLn pin, and counting to determine the period at high level starts after the transition of the input on the SSCLn pin to the high level. The interval until counting to determine the period at high level starts on the transition of the SSCLn pin to the high level is the total time which contains the SSCLn input delay, the noise filtering delay of the SSCLn pin (2 or 3 cycles of the filtering clock), and the internal processing delay (1 or 2 cycles of PCLK). The period at high level of the internal SCL signal is extended even if other devices are not placing the low level on the SSCLn line.

If the SIMR.IICCSC bit is 1, synchronization is obtained for the transmission and reception of data by taking the logical AND of the input on the SSCLn pin and the internal SCL signal. If the SIMR.IICCSC bit is 0, synchronization with the internal SCL signal is obtained for the transmission and reception of data.

If a slave device inserts a period of waiting into the interval until the transition of the internal SCL signal from the low to the high level after a request for the generation of a start, restart, or stop condition is issued, the time until generation is prolonged by that period.

If a slave device inserts a period of waiting after the transition of the internal SCL signal from the low to the high level, although the generation-completed interrupt is issued without stopping the period of waiting, generation of the condition itself is not guaranteed.

Figure 33.81 shows an example of operations to synchronize the clocks.

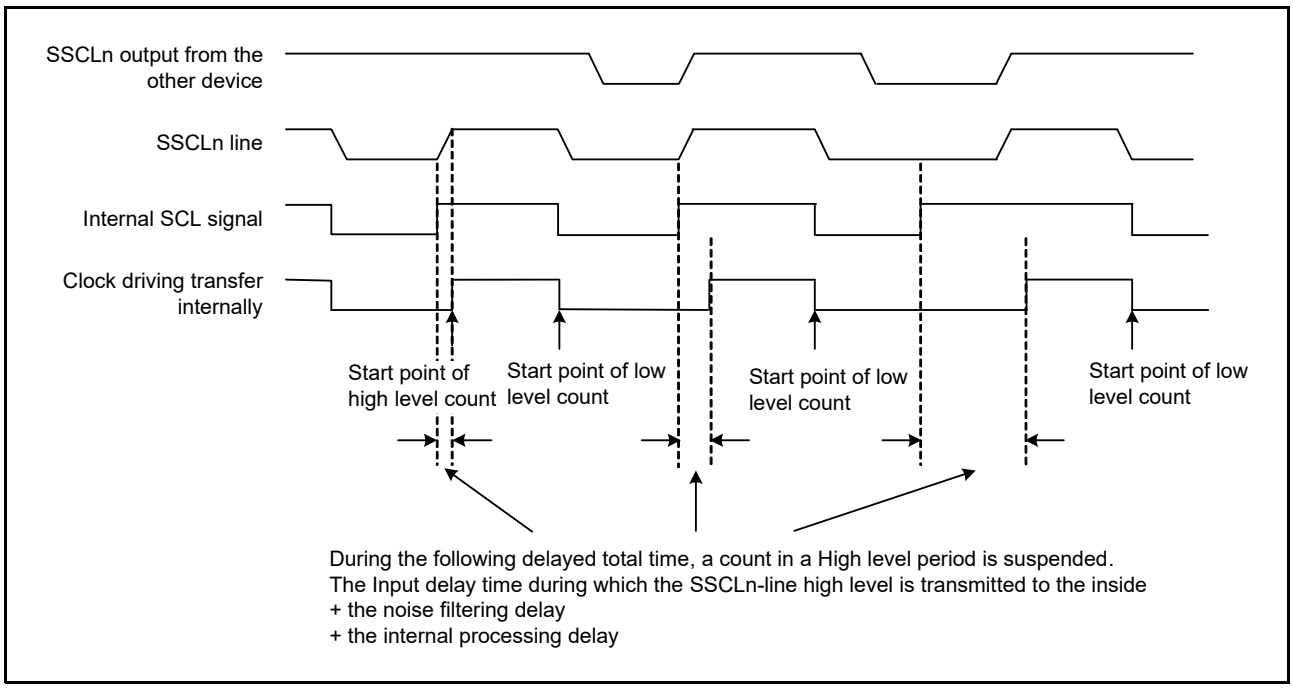


Figure 33.81 Example of Operations for Clock Synchronization

33.9.3 SDA Output Delay

The SIMR.IICDL[4:0] bits can be used to set a delay for output on the SSDAn pin relative to falling edges of output on the SSCLn pin. The delay-time settings are selectable from 0 to 31 cycles of the clock signal from the on-chip baud rate generator (The base is PCLK and selected the divided clock by the SCR2.CKS[1:0] bits). About Start/Restart/Stop conditions, 8bit-transmission data and acknowledge, the SSDAn pin output can be delayed.
If the SDA output delay is shorter than the falling time of the SSCLn output pin, the change of the SSDAn output pin will start while the SSCLn output pin level is falling, then there is a possibility of erroneous operation for slave devices. Ensure setting the SDA output delay greater than the SSCLn maximum falling time. (300 ns for I^2C normal/fast mode.)
Figure 33.82 shows the timing of delays in SDA output.

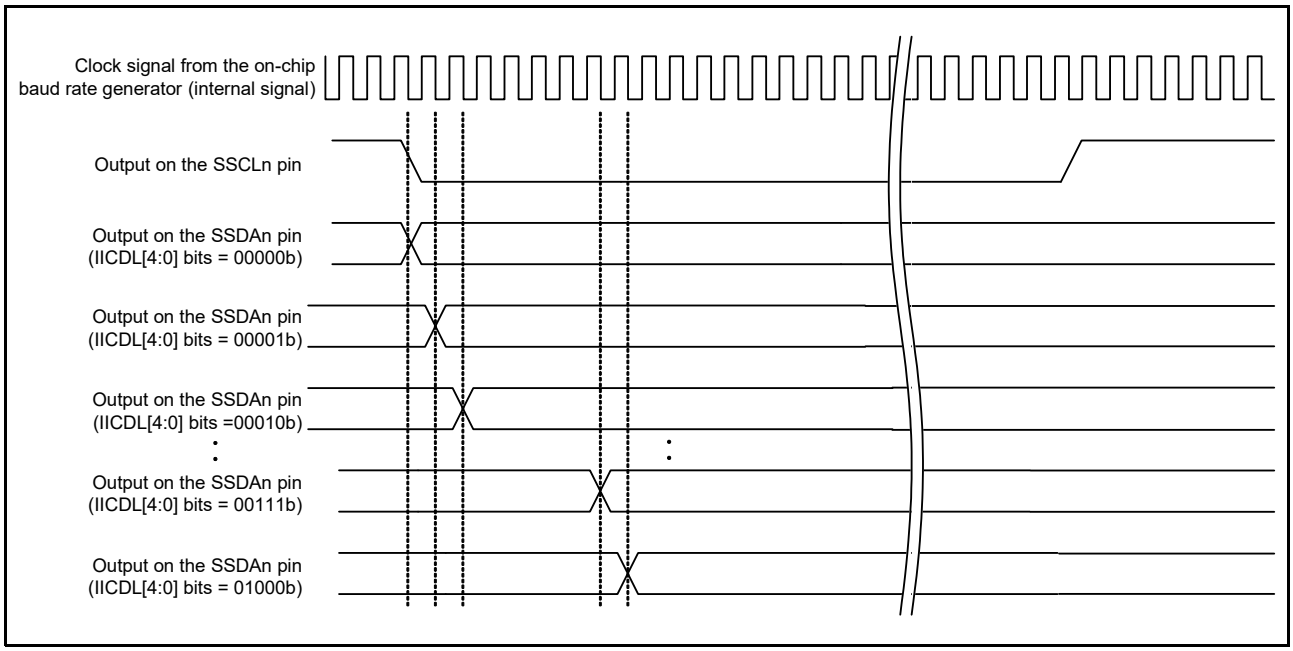
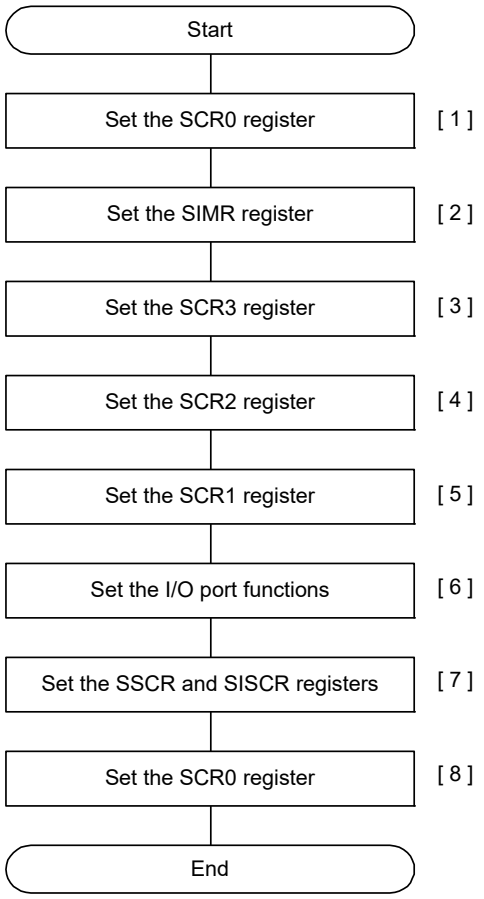


Figure 33.82 Timing of Delays in SDA Output

33.9.4 RSCI Initialization (Simple I2C Mode)

Write initial value (0000_0000h) to SCR0, then initialize RSCI according to Figure 33.83.
When changing the operating mode, transfer format, and so on, be sure to set 0 to SCR0.TE bit and SCR0.RE bit before proceeding with the changes. (Or write initial value to SCR0 again.) In simple I2C mode, the open-drain setting for the communication ports should be made on the port side.



[1] Set the following bits of SCR0 register to 0.
TEIE, TIE, RIE, TE, RE
If you have not changed from the initial settings, you can skip this step.

[2] Set the IICSDAS[1:0] and IICSCLS[1:0] bits to 11b.
Set the IICDL[4:0] and IICINTM bits as required.
Set the IICACKT and IICCSC bits to 1.

[3] Set the transmission/reception format as the communication mode (MOD[2:0] bits = 100b) and CKE[1:0] bits = 00b.

[4] Set the bit rate modulation function*1, the clock selection, and the bit rate.

[5] Set noise filter, communication pin status, parity check, CTS/RTS function.

[6] Set I/O port settings that allow use (on NMOS opendrain output pins and Hi-Z) of the SSCLn and SSDAn pin functions.

[7] Write 1 to the following bits and clear the corresponding flag.
SSCR.RDRFC, AFERC, APERC, MFFC, ORERC, DFERC, DPERC, DCMFC, ERSC
SISCR.IICSTIFC

[8] Set the TE and RE bits to 1.
To enable interrupts, set the TE bit and TIE bit, and the RE bit and RIE bit to 1 with one instruction at the same time.
(for transmission and when the SIMR.IICINTM bit is 1, clear the RIE bit.) Setting the TE and RE bits to 1 makes the SSCLn and SSDAn pin functions available.

Note 1. If you do not use the bit rate modulation unction, you do not need to set it.

Note: Set the SCR0.TE and RE bits to 0 or 1 at the same time.

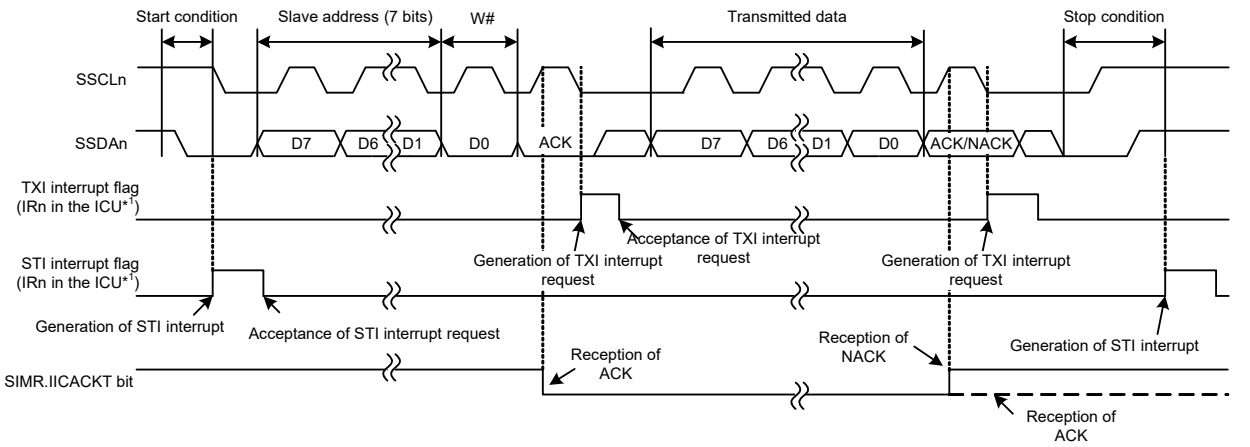
Figure 33.83 Example of the Flowchart of RSCI Initialization (for Simple I2C Mode)

33.9.5 Operation in Master Transmission (Simple I2C Mode)

Figure 33.84 and Figure 33.85 show examples of operations in master transmission, Figure 33.86 to Figure 33.88 show the example flowcharts. See Table 33.47 about the STI interrupt.

Figure 33.84 shows the operation example when SIMR.IICINTM bit is 1 (Reception/Transmission interrupt are in use). In this case, you can start DMAC or DTC by TXI interrupt. However, if use DMAC or DTC, ACK/NACK can not be confirmed. So, if you want to confirm ACK/NACK, prepare transmit data by CPU. In simple I2C mode, TXI interrupt is generated when communication of one frame is completed. And it isn't used reception interrupt in master transmission, so the SCR0.RIE bit set to 0.

Figure 33.86 shows a flow chart in the case of SIMR.IICINTM bit is 1 and address transmission by CPU and data transmission by DTC or DMAC. Figure 33.87 shows a flow chart of address and data transmission by CPU. When 10-bit slave addresses are in use, steps [3] and [4] are repeated twice.



Note 1. Refer to section 14, Interrupt Controller (ICUG) for details on the corresponding interrupt vector number.

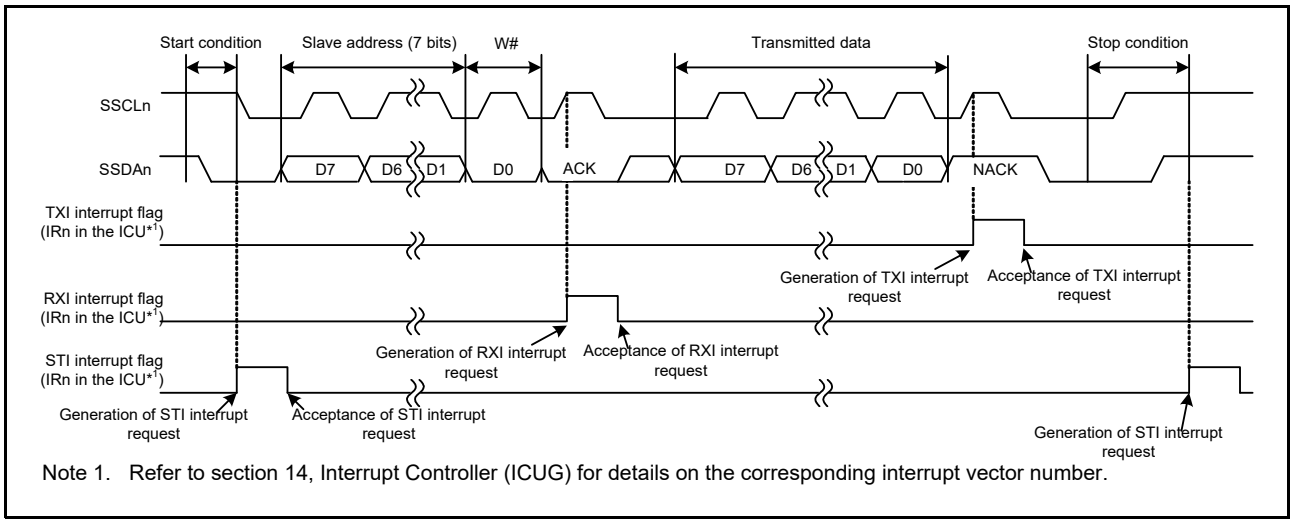
Figure 33.84 Example 1 of Operations for Master Transmission in Simple I2C Mode (with 7-Bit Slave Addresses, Transmission Interrupts, and Reception Interrupts in Use (SIMR.IICINTM Bit = 1))

Figure 33.85 shows an example of operations when SIMR.IICINTM bit is 0 (ACK and NACK interrupt in use). In this case, DTC or DMAC is activated by the ACK interrupt, and necessary number of data bytes are transmitted. When the NACK is received, error processing, such as transmission stop and retransmission, is performed by the NACK interrupt as the trigger.

Figure 33.88 shows a flow chart of SIMR.IICINTM bit is 0

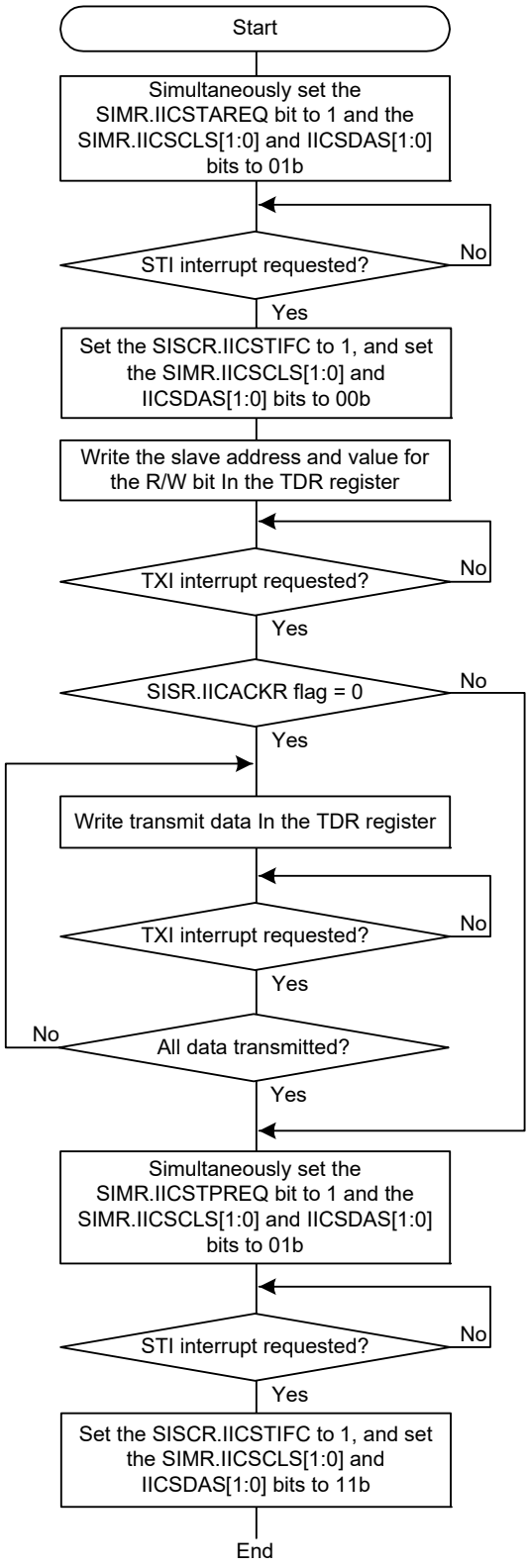
To resume communication after interrupting communication for some reason after writing transmit data to TDR, follow the procedure below.

1. Set the SCR0.TE and SCR0.RE bits to 0 to stop communication.
2. Set SIMR.IICSCLS[1:0] and SIMR.IICSDAS[1:0] bits to 11b, release the I2C bus, and clear various condition generation requests.
3. When the SSR.RDRF flag is 1, the RDR register is read by dummy and the RDRF bit is set to 0.
4. Set the SCR0.TE and SCR0.RE bits to 1 and restart communication.



Note 1. Refer to section 14, Interrupt Controller (ICUG) for details on the corresponding interrupt vector number.

Figure 33.85 Example 2 of Operations for Master Transmission in Simple I²C Mode (with 7-Bit Slave Addresses, ACK Interrupts, and NACK Interrupts in Use (SIMR.IICINTM Bit = 0))



[0] [0] For transmission, set the SCR0.RIE bit to 0 at initialization. (RXI and ERI interrupts requests are disabled)

[1] [1] Generate a start condition.

[2] [2] Writing to TDR register:
Writing the slave address and value for the R/W bit to TDR register.

[3] [3] Confirming ACK response from the slave address:
Check the SISR.IICACKR bit. If its value is 0, it is indicated that the slave device responded with ACK and operations proceed. If its value is 1, it is indicated that there was no response from a slave device so the next transition is to generation of the stop condition.

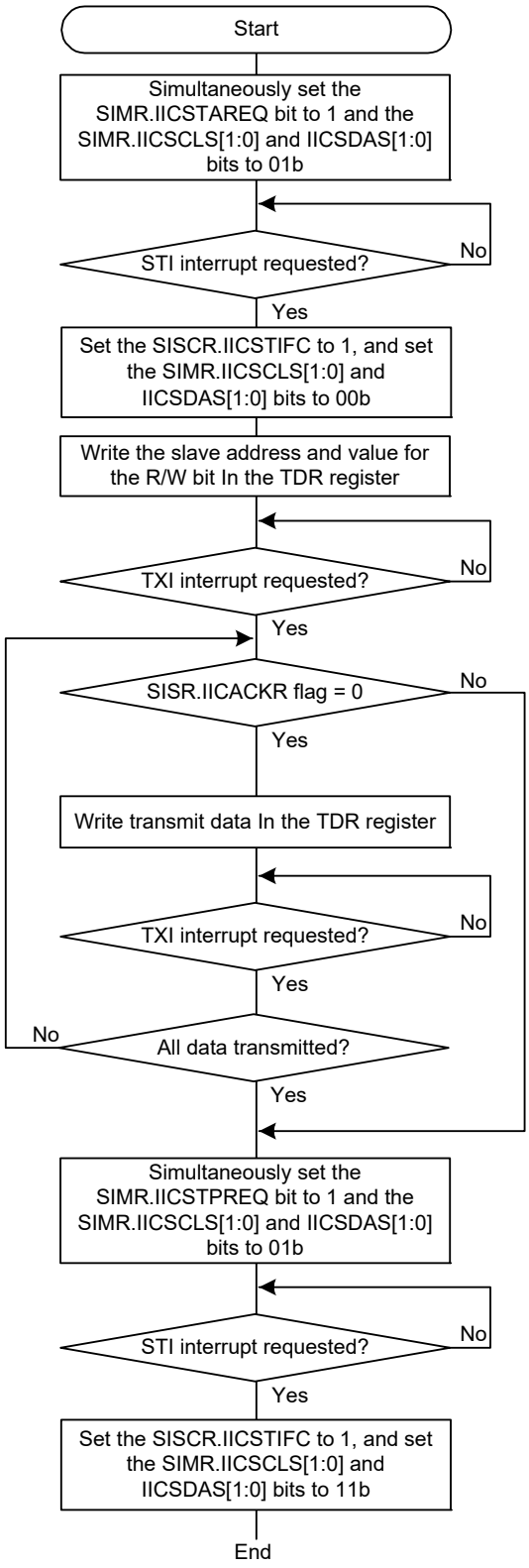
If 10-bit slave addresses are in use, processing of [2] and [3] is repeated twice.

[4] [4] Steps for continuing with serial transmission:
When transmission is to continue, write further transmit data to TDR register. Except for the first data to be transmitted, a TXI request can activate the DMAC or DTC to handle writing of data to TDR register.
But it cannot be confirmed status of ACK or NACK in this case.

[5] [5] Generation of a stop condition.

Note: In simple I^2C mode, the TXI interrupt request is generated when communication is completed, unlike the timing during clock synchronous transmission.

Figure 33.86 Example of the Procedure for Master Transmission Operations in Simple I^2C Mode (when SIMR.IICINTM Bit is 1, and when Confirming ACK/NACK by Address Transmission Only)



[0] [0] For transmission, set the SCR0.RIE bit to 0 at initialization. (RXI and ERI interrupts requests are disabled)

[1] [1] Generate a start condition.

[2] [2] Writing to TDR register:
Writing the slave address and value for the R/W bit to TDR register.

[3] Confirming ACK response from the slave address:
Check the SISR.IICACKR bit. If its value is 0, it is indicated that the slave device responded with ACK and operations proceed. If its value is 1, it is indicated that there was no response from a slave device so the next transition is to generation of the stop condition.

If 10-bit slave addresses are in use, processing of [2] and [3] is repeated twice.

[4] Steps for continuing with serial transmission:
When transmission is to continue, write further transmit data to TDR. To checking whether ACK or NACK in every transmission, you cannot use DMA or DTC.

[5] Generation of a stop condition.

Note: In simple I^2C mode, the TXI interrupt request is generated when communication is completed, unlike the timing during clock synchronous transmission.

Figure 33.87 Example of the Procedure for Master Transmission Operations in Simple I^2C Mode (when SIMR.IICINTM Bit is 1, and when Confirming ACK/NACK in All Transmissions)

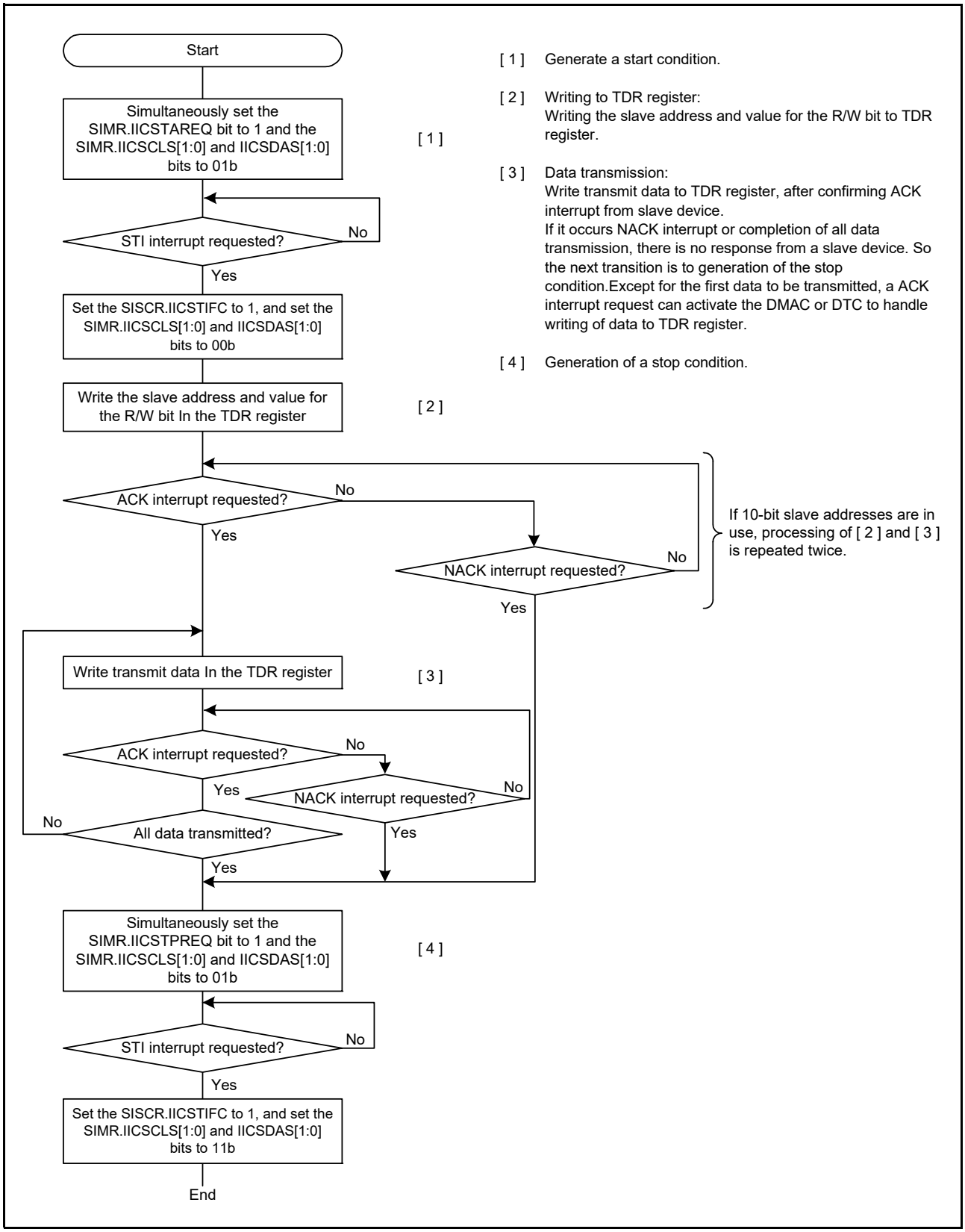
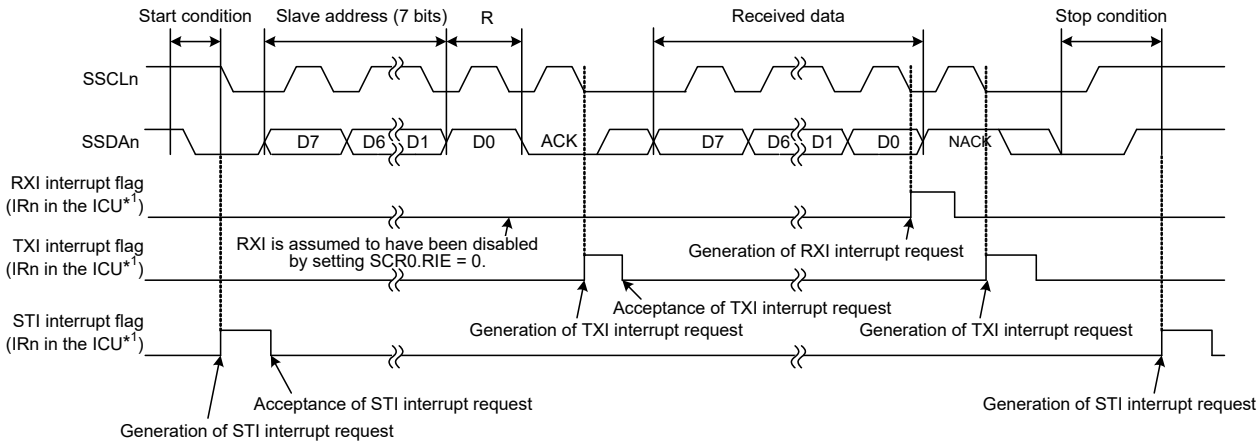


Figure 33.88 Example of the Procedure for Master Transmission Operations in Simple I²C Mode (when SIMR.IICINTM Bit is 0)

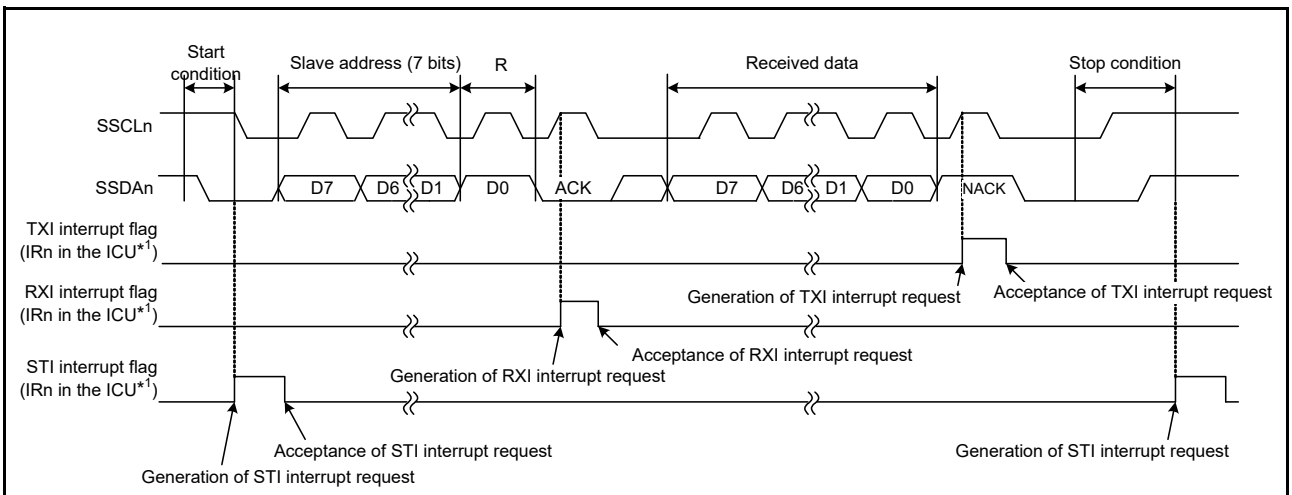
33.9.6 Master Reception (Simple I2C Mode)

Figure 33.89 and Figure 33.90 show example of operations in simple I2C mode master reception. Figure 33.91 and Figure 33.92 show flowchart of the master reception. The value of the SIMR.IICINTM bit is assumed to be 1 (use reception and transmission interrupts) and 0 (use ACK and NACK interrupts).
In simple I2C mode, the transmit end interrupt (TXI) is generated when communication of one frame is completed.



Note 1. Refer to section 14, Interrupt Controller (ICUG) for details on the corresponding interrupt vector number.

Figure 33.89 Example of Operations for Master Reception in Simple I2C Mode (with 7-Bit Slave Addresses, Transmission Interrupts, and Reception Interrupts in Use (SIMR.IICINTM Bit = 1))



Note 1. Refer to section 14, Interrupt Controller (ICUG) for details on the corresponding interrupt vector number.

Figure 33.90 Example of Operations for Master Reception in Simple I2C Mode (with 7-Bit Slave Addresses, ACK and NACK Interrupt in Use (SIMR.IICINTM Bit = 0))

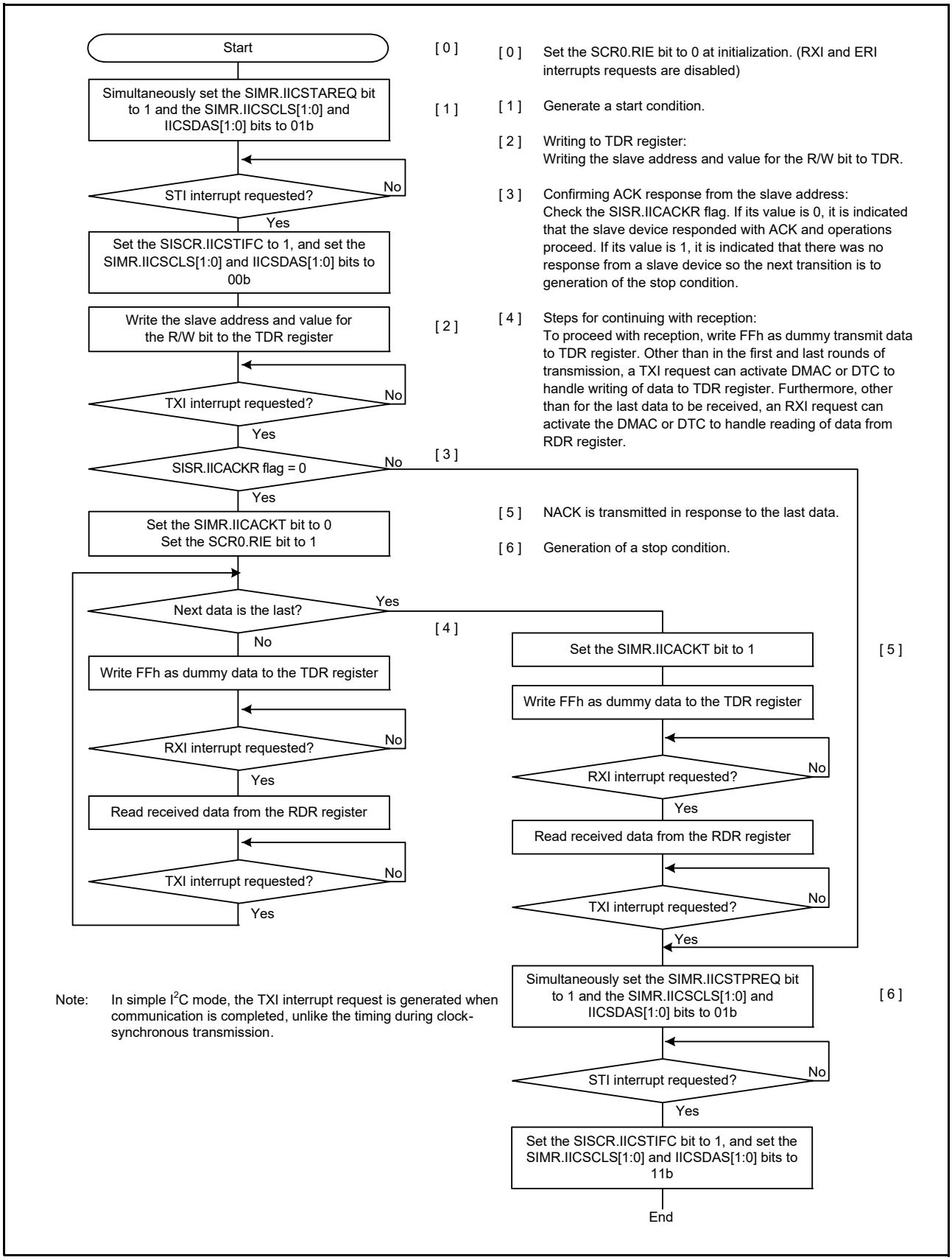


Figure 33.91 Example of the Procedure for Master Reception Operations in Simple I²C Mode (When SIMR.IICINTM Bit is 1, and Transmission Interrupts and Reception Interrupts in Use)

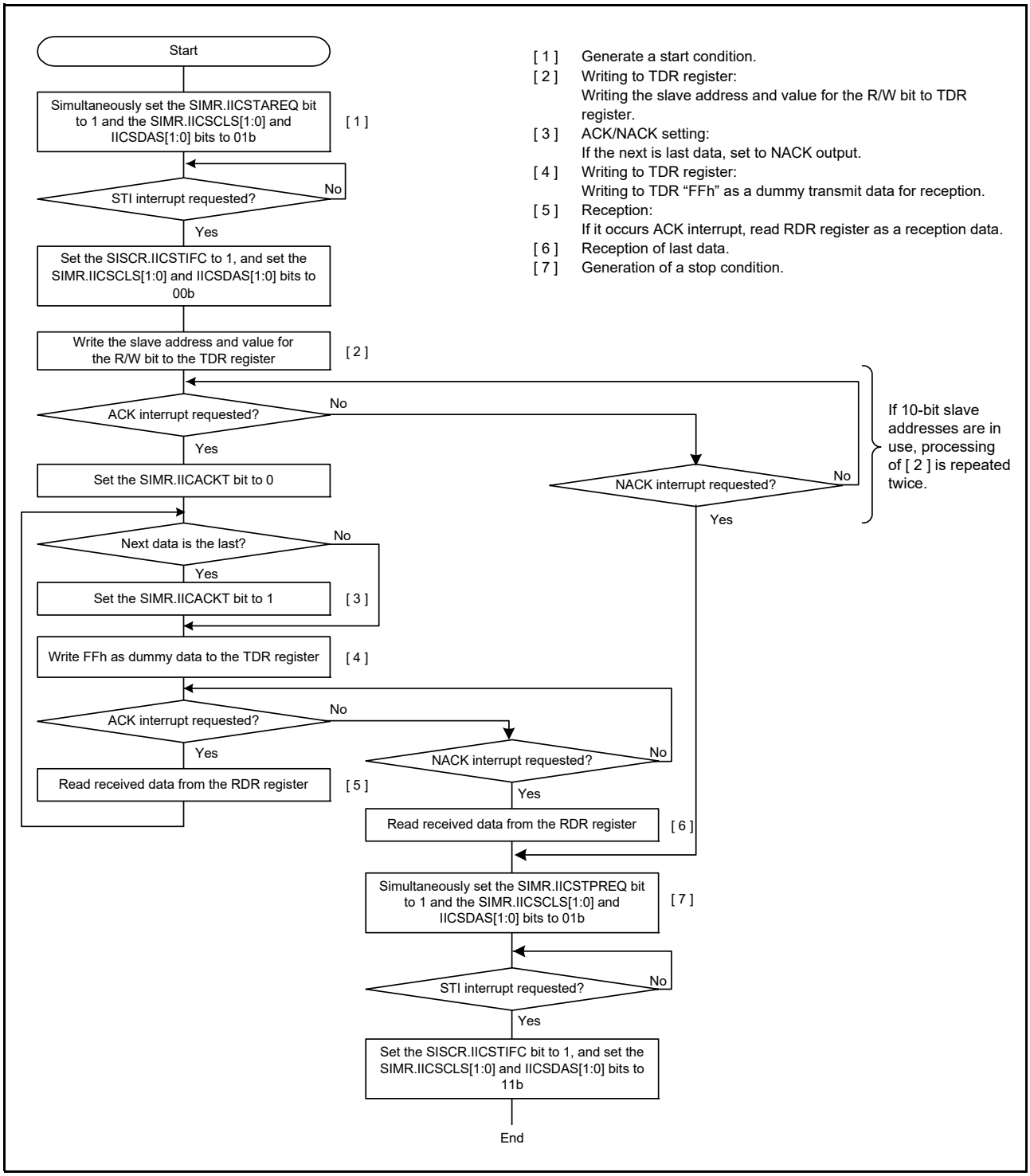


Figure 33.92 Example of the Procedure for Master Reception Operations in Simple I²C Mode (When SIMR.IICINTM Bit is 0, and ACK Interrupts and NACK Interrupts are in Use)

33.10 Operation in Clock Synchronous Mode

Figure 33.93 shows the communication data format of clock synchronous serial communication.
In clock synchronous mode, data is transmitted and received in synchronization with clock pulses. A communication data character consists of 8-bit data. In clock synchronous mode, no parity bit can be added.
In data transmission when CPHA bit = 1 and CPOL bit = 1, the RSCI outputs data from the falling edge of the sync clock until the next falling edge. In data reception, data is read at the rising edges of the sync clock. After 8-bit data is output, the communication line holds the final-bit output state. In slave communication when CPHA bit = 0, however, the communication line holds the first-bit output state.
Because the RSCI has an internal transmitter and a receiver independently, RSCI enable full-duplex communication by sharing a communication clock of the transmitter and the receiver. Furthermore, because both the transmitter and the receiver have a double- buffer structure, continuous transmission and reception are possible by writing the next transmit data during transmission and reading the previous receive data during reception.

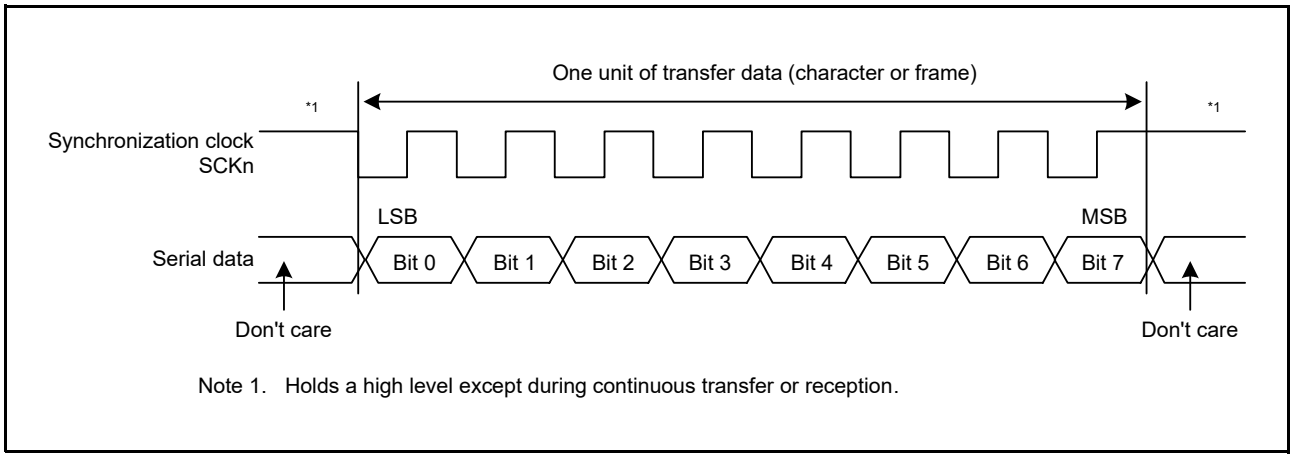


Figure 33.93 Data Format in Clock Synchronous Serial Communications (LSB First, CPHA Bit = 1, CPOL Bit = 1)

33.10.1 Clock

(1) When the Internal Clock is Selected

When the SCR3.CKE[1:0] bits are set to 00b or 01b (master mode), the internal clock generated by the on-chip baud rate generator can be selected and the sync clock is output from the SCKn pin. Eight pulses of the sync clock are output during single-character transmission/reception. The sync clock remains at a high level*1 while no transmission or reception is performed. In transmission-only or transmission/reception, the sync clock is not output unless transmit data is prepared.
When the internal clock is selected, the clock with a delay from the SCKn signal is used for the master reception sampling clock. This ensures the data setup time and hold time for high-speed communication.

Note 1. When SCR3.CPHA bit = 0 and SCR3.CPOL bit = 1 or when SCR3.CPHA bit = 1 and SCR3.CPOL bit = 1, the sync clock stops at a high level. When SCR3.CPHA bit = 0 and SCR3.CPOL bit = 0 or when SCR3.CPHA bit = 1 and SCR3.CPOL bit = 0, the sync clock stops at a low level.

(2) When the External Clock is Selected

When the SCR3.CKE[1:0] bits are set to 10b or 11b (slave mode), data is transmitted and received using the external clock that is input from the SCKn pin.

33.10.2 CTS and RTS Functions

The CTS function performs transmission/reception and controls transmission start using the CTSn# pin input when the internal clock is selected. Setting the SCR1.CTSE bit to 1 enables the CTS function. In clock synchronous communication, the CTS function can be used for the internal clock and the RTS function can be used for the external clock, so the CTS function and RTS function cannot be used at the same time.
When the CTS function is enabled, transmission/reception and transmission start only when the CTSn# pin input level is low.
When using the FIFO, if the CTSn# signal remains high before transmission, transmission will not start, but the number of data stored will be “number written to the TDR register – 1” (unlike using asynchronous FIFO). This is because data is transferred to the TSR register after writing to the TDR register, but if the CTSn# signal is set to low level, transmission starts from the TSR register, so there is no problem.
Even if the CTSn# pin input becomes high level during transmission/reception or transmission operation, frames that are being transmitted/received or being transmitted are not affected and transmission/reception or transmission operation continues.
The RTS function makes a serial communication start request using the RTSn# pin output when the external sync clock is selected. When serial communication is enabled, the RTSn# pin outputs a low level. A low level and a high level are output under the following conditions.

(a) When the SCR3.FM Bit is 0 (Non-FIFO Mode)

[Conditions for low-level output]
When the following conditions are all satisfied

- The SCR0.RE or SCR0.TE bit is 1
- No receive data are present before reading and reception is not in progress. (when SCR0.RE bit = 1)
- Data written in the TDR register is ready for transmission (when SCR0.TE bit = 1)
- The SSR.ORER flag is 0

[Condition for high-level output]
The conditions for low-level output have not been satisfied.
When SCR0.RE bit is set to 0 without reading the RDR register to terminate reception after reception is complete, the RTSn# pin output level remains high. At this time, write 0 to SCR0.RE bit.

(b) When the SCR3.FM Bit is 1 (FIFO Mode)

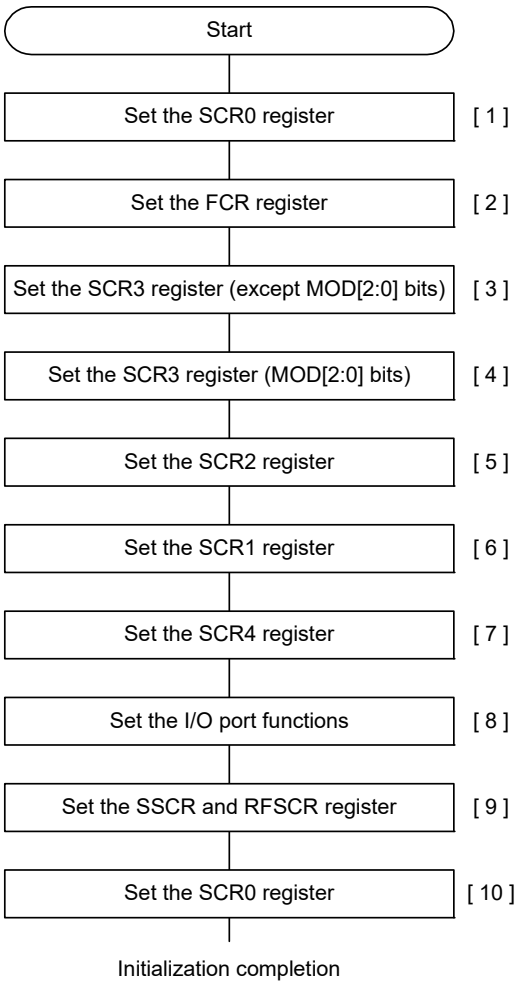
[Conditions for low-level output]
When the following conditions are all satisfied

- The SCR0.RE or SCR0.TE bit is 1
- The number of receive data stored in the receive FIFO (RDR register) is less than the value set in FCR.RSTRG[4:0] (when SCR0.RE bit = 1)
- Data written in the transmit FIFO (TDR register) is ready for transmission (when SCR0.TE bit = 1)
- The SSR.ORER flag is 0

[Condition for high-level output]
When the conditions for low-level output are not satisfied

33.10.3 RSCI Initialization (Clock Synchronous Mode)

Before starting data transmission/reception, write 0 to SCR0.TE bit and SCR0.RE bit (or write initial values to the SCR0 register) and initialize the RSCI according to the flowchart example in Figure 33.94.
Before changing operating mode or communication format, also be sure to write 0 to TE bit and RE bit.
Note that writing 0 to the RE bit does not initialize the ORER, AFER, APER, and RDRF flags in SSR register and the RDR register. Also note that writing 0 to the TE bit does not initialize the SSR.TEND flag in FIFO mode. Attention is also needed for changing operating mode.
When the SCR0.TIE bit = 1, note that setting the TE bit to 1 from 0 generates a TXI interrupt.



[1] Set the following bits of SCR0 register to 0[*1].
TEIE, TIE, RIE, TE, RE
If you have not changed from the initial settings, you can skip this step.

[2] Set the TFRST and RFRST bits to 1 to empty FIFO.
Set the TTRG[4:0], RTRG[4:0], and RSTRG[4:0] bits.

[3] Set SCR3 except of communication mode.
FIFO use/no-use
Transmission/reception format
Clock setting
Leave unused bits (CHR[1:0], STOP, RXDESEL, MP, DEEN, GM, BLK) at their initial values.[*2]

[4] Set communication mode (MOD[2:0] bits = 010b)[*2].

[5] Select clock, set bit rate[*3]. Leave unused bits (SCR2.BCP[2:0], ABCS, ABCSE, BRME, and MDDR[7:0]) at their initial values.

[6] Set up the half-duplex communication, the loop-back function, communication pin status, and the CTS/RTS function.

[7] Set up the adjust sampling timing function. Leave unused bits (SCR4.CMPD[8:0] bits) at their initial values.

[8] Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.

[9] Write 1 to the following bits and clear the corresponding flag.
SSCR.RDRFC, AFERC, APERC, MFFC, ORERC, DFERC, DPERC, DCMFC, ERSC
RFSCR.DRC

[10] Set the TE or RE bit to 1.[*1, *4] To enable interrupts, set the TE bit and TIE bit, and the RE bit and RIE bit to 1 with one instruction at the same time.
Setting the TE and RE bits allows TXDn and RXDn to be used.

Note 1. In simultaneous transmit and receive operations, the TE and RE bits in SCR0 should both be to 0 or set to 1 simultaneously.
Note 2. Set the clock (CPOL and CPHA bits) before setting the communication mode.
Note 3. If you use an external clock, you do not need to set it.
Note 4. When using the internal clock (master), the setting of reception only is prohibited.

Figure 33.94 Example of RSCI Initialization Flowchart (Clock Synchronous Mode)

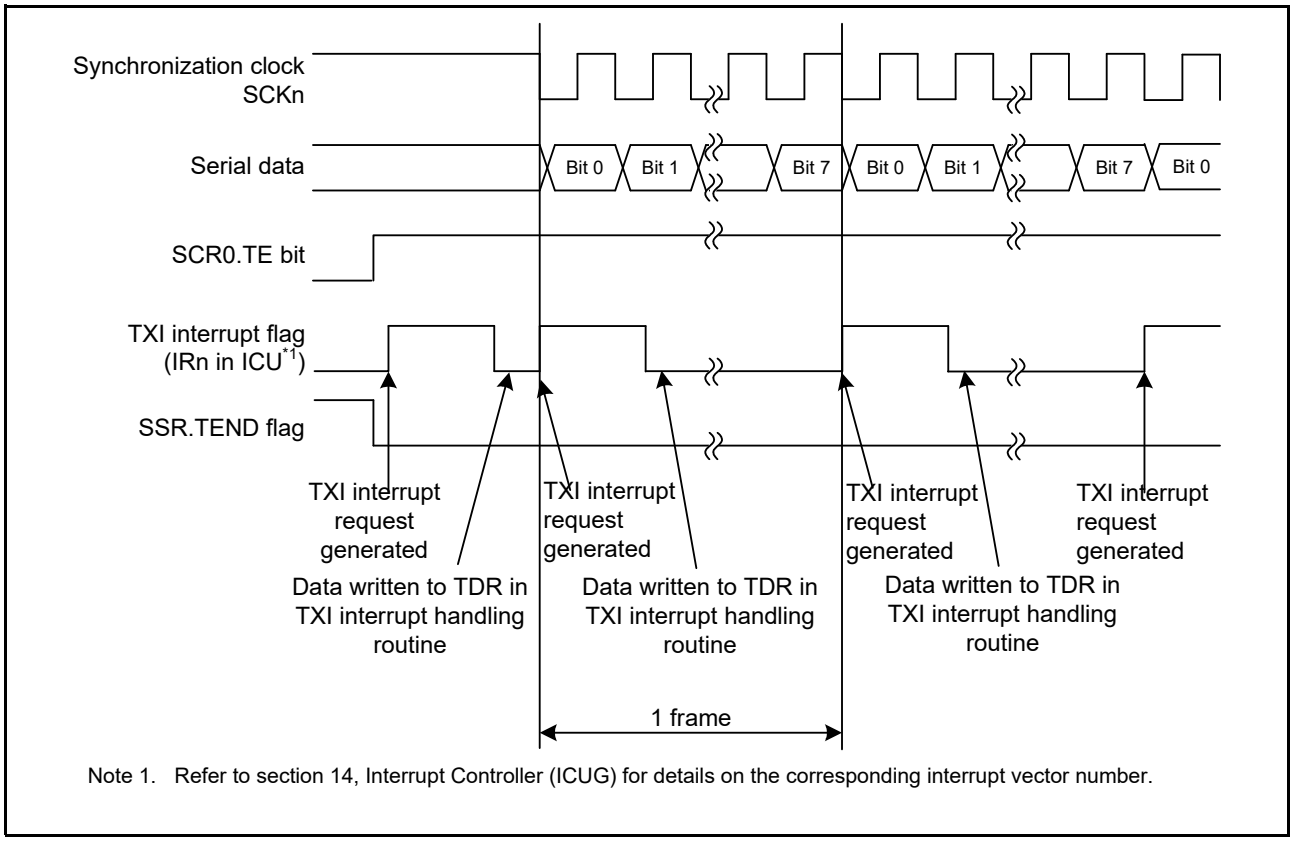
33.10.4 Serial Data Transmission (Clock Synchronous Mode)

(1) Non-FIFO Mode

Figure 33.95 to Figure 33.97 show an example of the operation for serial transmission in clock synchronous mode. In serial data transmission, the RSCI operates as described below.

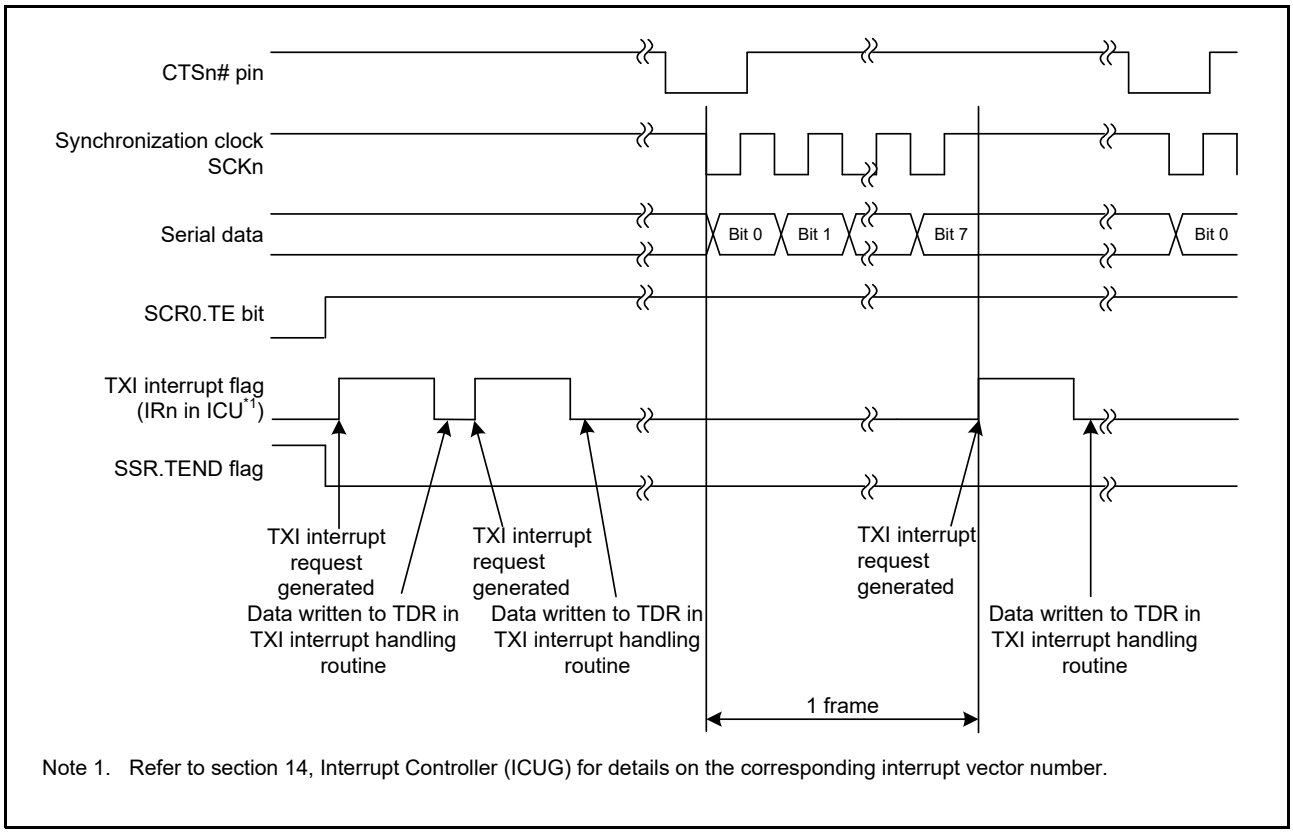
1. When data is written to the TDR register in the TXI interrupt routine, the RSCI transfers the data from the TDR register to the TSR register. When starting data transmission, set the SCR0.TIE bit and the SCR0.TE bit to 1 simultaneously by a single instruction. Then a TXI interrupt request is generated.
2. The written data is transferred from the TDR register to the TSR register, which starts transmission. When the SCR0.TIE bit is set to 1 at this time, a TXI interrupt request is generated. Writing the next transmit data to the TDR register before transmission of data transferred previously in the TXI interrupt routine is complete enables continuous transmission. When a TEI interrupt request is used, after the final transmit data is written to the TDR register in the TXI interrupt request processing routine and the final data's transmission is started, set 0 to the SCR0.TIE bit and set 1 to the TEIE bit.
3. The TXDn pin outputs 8-bit data in synchronization with the output clock (in clock output mode) or with the input clock (when external clock is selected). When the SCR1.CTSE bit = 1 (CTS function enabled), the output clock starts after the CTS signal input becomes low level.
4. Update (data write) of the TDR register is checked at the final-bit transmission timing.
5. When the TDR register has been updated, data is transferred from the TDR register to the TSR register to start sending the next frame.
6. If the TDR register has not been updated, the SSR.TEND flag is set to 1 and the final-bit output state is retained. When the SCR0.TEIE bit is set to 1 at this time, a TEI interrupt request is generated. The SCKn pin is held high.

Figure 33.98 shows a sample flowchart of serial data transmission.



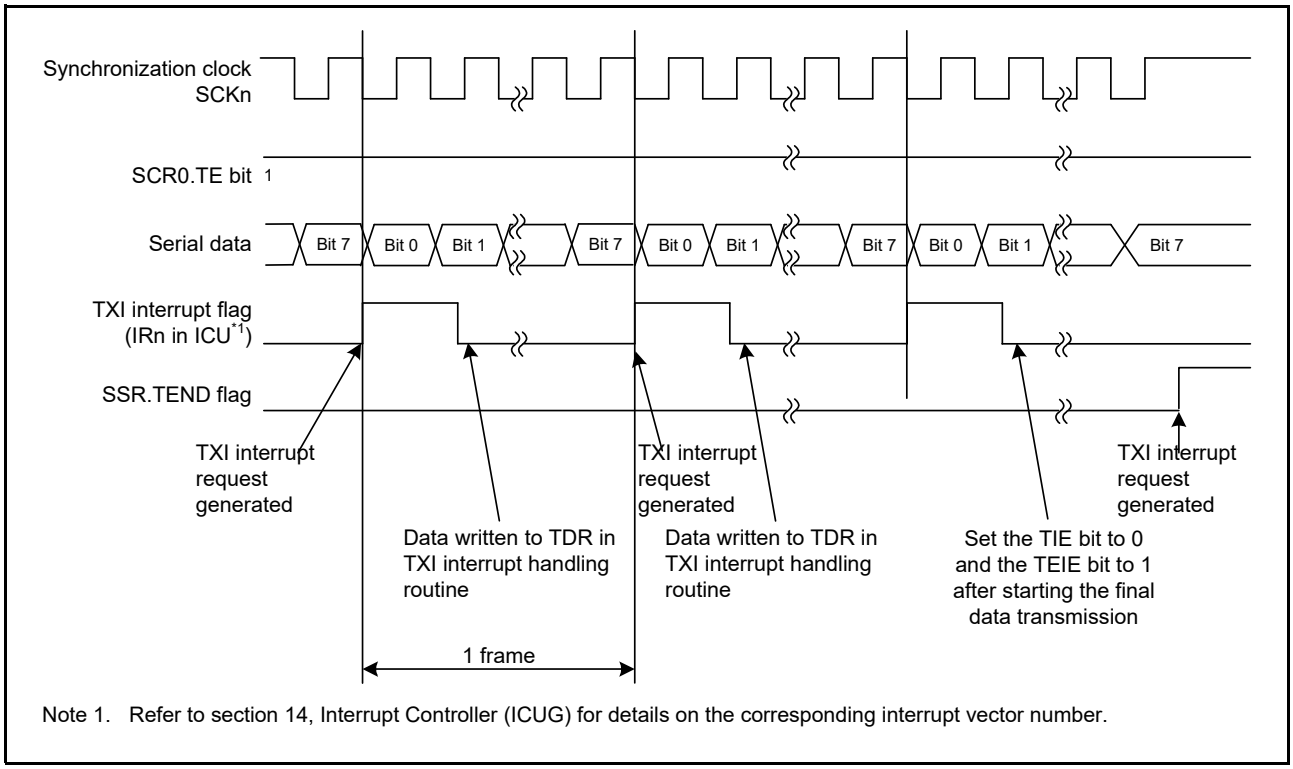
Note 1. Refer to section 14, Interrupt Controller (ICUG) for details on the corresponding interrupt vector number.

Figure 33.95 Serial Transmission Example in Clock Synchronous Mode (1) (CTS Function Not Used/ Transmission Start/CPHA Bit = 1, CPOL Bit = 1)



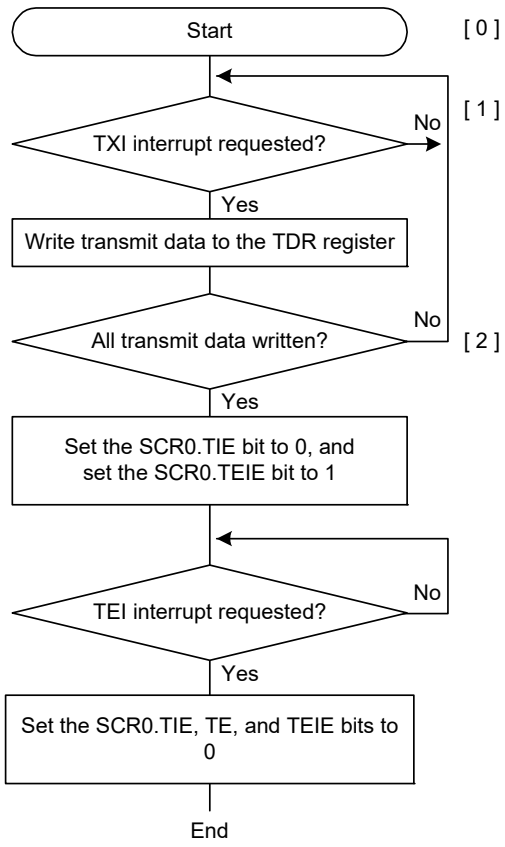
Note 1. Refer to section 14, Interrupt Controller (ICUG) for details on the corresponding interrupt vector number.

Figure 33.96 Serial Transmission Example in Clock Synchronous Mode (2) (CTS Function Used/Transmission Start/CPHA Bit = 1, CPOL Bit = 1)



Note 1. Refer to section 14, Interrupt Controller (ICUG) for details on the corresponding interrupt vector number.

Figure 33.97 Serial Transmission Example in Clock Synchronous Mode (3) (During Transmission to Transmission End/CPHA Bit = 1, CPOL Bit = 1)



[0] Set the SCR0.TE and TIE bits to 1 at initialization.

[1] Writing transmit data to TDR register by a TXI interrupt request:
The transmit data empty interrupt (TXI) is generated when transmission data forwards to TSR from TDR register. Write transmit data to TDR register once in the TXI interrupt handling routine.

[2] Serial transmission continuation procedure:
To continue serial transmission, write transmit data to TDR register in the TXI interrupt handling routine. Transmit data can also be written by activating the DMAC or DTC.
If you use TEI interrupt request, set SCR0.TIE bit to 0 and TEIE bit to 1 after writing the last transmit data to TDR register.

Note: When the external clock is in use (the value of the SCR3.CKE[1:0] bits is 10b or 11b), the rising edge on the SCKn pin for the last bit sets the SSR.TEND flag to 1. Setting SCR0.TE bit to 0 immediately after this, then it may lead to insufficient received data hold time on the receiver side.

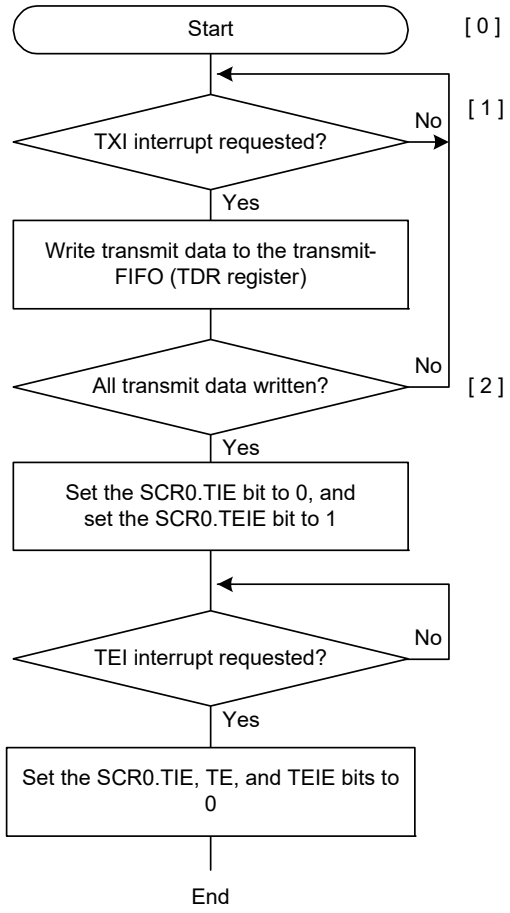
Figure 33.98 Example Flowchart of Serial Transmission in Clock Synchronous Mode (Non-FIFO Mode)

(2) FIFO Mode

Figure 33.99 shows an example of flowchart of serial transmission (a FIFO buffer selected) in clock synchronous mode with FIFO enabled.
The RSCI operates as follows when serial data transmission.

1. When data is written to the transmit FIFO (TDR register) in the TXI interrupt routine, the RSCI transfers the data from the transmit FIFO (TDR register) to the TSR register. The number of writable transmit data is [32 – number of unsent transmit data stored in the transmit FIFO (TDR register)]. If the SCR0.TIE and SCR0.TE bits are simultaneously set to 1 at the start of data transmission, a TXI interrupt request is generated.
2. Data is transferred from the transmit FIFO (TDR register) to the TSR register and transmission starts. When the number of data stored in the transmit FIFO (TDR register) is equal to or less than the threshold value of the transmit FIFO, the SSR.TDRE flag is set to 1. When the SCR0.TIE bit is set to 1 at this time, a TXI interrupt request is generated. Writing the next transmit data to the transmit FIFO (TDR register) in the TXI interrupt routine before transmission of data written to the transmit FIFO (TDR register) is complete enables continuous transmission. When a TEI interrupt request is used, after the final transmit data is written to the transmit FIFO (TDR register) in the TXI interrupt request processing routine, set 0 to the SCR0.TIE bit and set 1 to the TEIE bit.
3. The TXDn pin outputs 8-bit data in synchronization with the output clock (in clock output mode) or with the input clock (when external clock is selected). When the SCR1.CTSE bit = 1 (CTS function enabled), the output clock starts after the CTSn# pin input becomes low level.
4. The RSCI checks whether unsent transmit data is remaining in the transmit FIFO (TDR register)*1 at the final-bit transmission timing.
5. When data is remaining in the transmit FIFO (TDR register), the data is transferred from the transmit FIFO (TDR register) to the TSR register to start sending the next frame.
6. If no data is remaining in the transmit FIFO (TDR register), the SSR.TEND flag is set to 1 and the final-bit output state is retained. When the SCR0.TEIE bit is set to 1 at this time, a TEI interrupt request is generated. The SCKn pin is held high.

Note 1. The number of unsent transmit data stored in the TDR register (transmit FIFO) can be monitored by reading the TFSR.T[5:0] bits.



[0] Set the SCR0.TE and TIE bits to 1 at initialization.

[1] Writing transmit data to transmit FIFO (TDR) by a TXI interrupt request:
The transmit data empty interrupt (TXI) is generated when transmission data forwards to TSR from the transmit FIFO (TDR register) and the number of transmit-data written in the transmit FIFO is equal to or less than the transmit FIFO threshold.

[2] Serial transmission continuation procedure:
To continue serial transmission, write transmit data to the transmit FIFO (TDR register) by using a TXI interrupt and clear the SSR.TDRE flag to 0.
The number of transmission data it's possible to write in is 32 minus (the number of stored data in transmit FIFO).
Transmit data can also be written to the transmit FIFO (TDR register) by activating the DMAC or DTC by TXI interrupt request. In this case, the TDRE flag is cleared automatically, so please don't operate the TDRE flag.
If you use TEI, set TIE bit to 0 and TEIE bit to 1 after writing the last transmit data to TDR register.

Note: When the external clock is in use (the value of the CKE[1:0] bits is 10b or 11b), the rising edge on the SCKn pin for the last bit sets the SSR.TEND flag to 1. Setting SCR0.TE bit to 0 immediately after this, then it may lead to insufficient received data hold time on the receiver side.

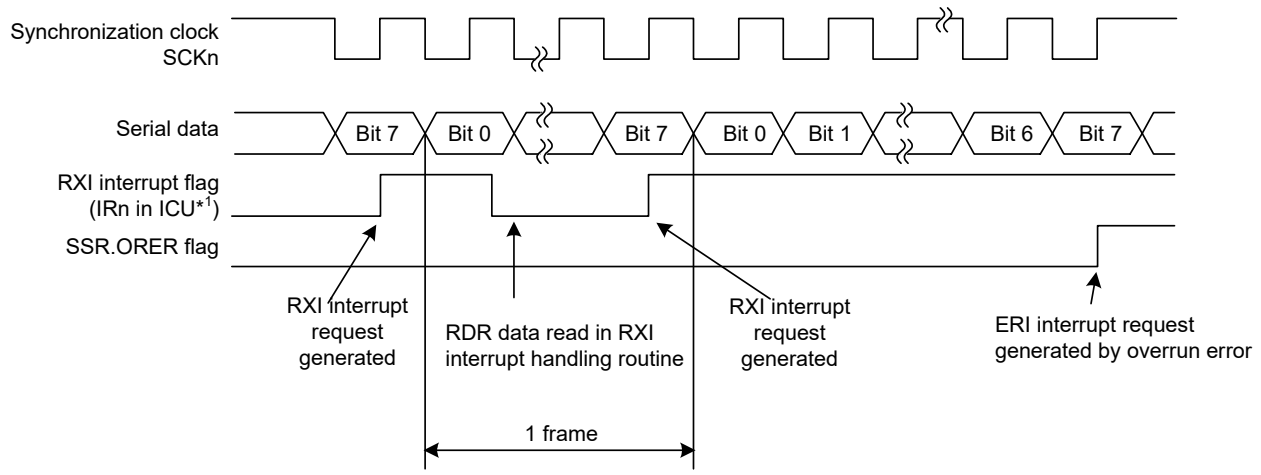
Figure 33.99 Example Flowchart of Serial Transmission in Clock Synchronous Mode (FIFO Mode)

33.10.5 Serial Data Reception (Clock Synchronous Mode)

(1) Non-FIFO Mode

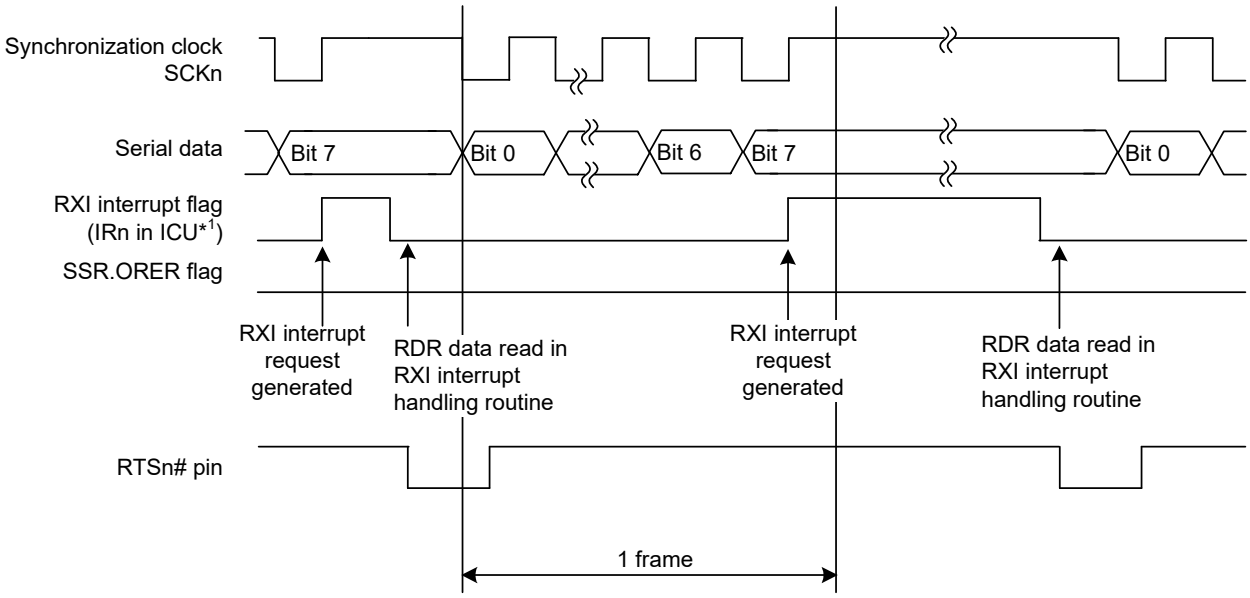
Figure 33.100 and Figure 33.101 show operation examples of serial data reception in clock synchronous mode.
The RSCI operates as follows during serial data reception. Reception-only operation is possible only in slave mode. (In master mode, reception-only operation is prohibited.)

1. When the SCR0.RE bit is set to 1, the RTSn# pin output becomes low (when the RTS function is used).
2. The RSCI starts data reception in synchronization with input or output of the sync clock, and transfers receive data to the RSR register.
3. When an overrun error occurs, the SSR.ORER flag is set to 1. When the SCR0.RIE bit = 1 at this time, an ERI interrupt request is generated and the received data is not transferred to the RDR register.
4. When data is normally received, the received data is transferred to the RDR register. When the RIE bit = 1 at this time, an RXI interrupt request is generated. Reading the received data transferred to the RDR register in the RXI interrupt handling routine before the next data is completely received enables continuous reception. When the received data transferred to the RDR register is read, the RTSn# pin output becomes low (when the RTS function is used).
   If you want to prevent the RTSn# pin output from turning low after the final data is received, clear the SCR0.RE bit to 0 and then read the RDR register.



Note 1. Refer to section 14, Interrupt Controller (ICUG) for details on the corresponding interrupt vector number.

Figure 33.100 Example of Operation for Serial Reception in Clock Synchronous Mode (1) (When RTS Function is Not Used/CPHA Bit = 1, CPOL Bit = 1)

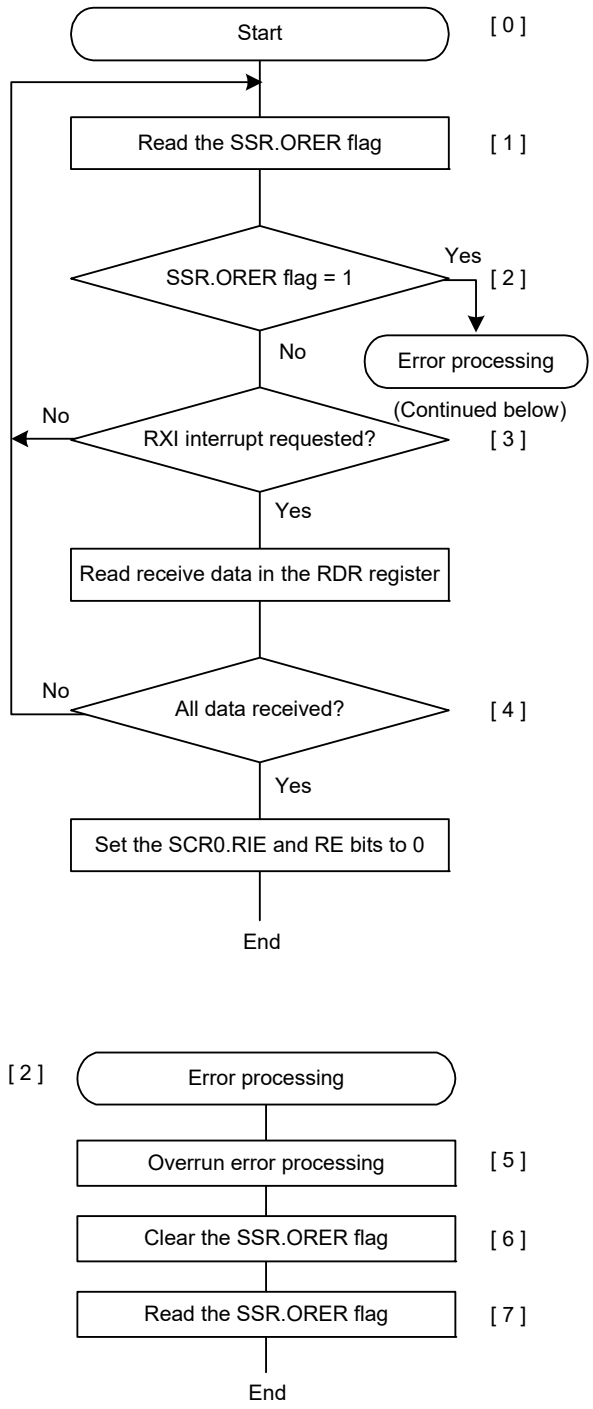


Note 1. Refer to section 14, Interrupt Controller (ICUG) for details on the corresponding interrupt vector number.

Figure 33.101 Example of Operation for Serial Reception in Clock Synchronous Mode (2) (When RTS Function is Used/CPHA Bit = 1, CPOL Bit = 1)

While the reception error flag is set to 1, subsequent reception are disabled. Therefore, before continuing reception, be sure to clear the ORER, AFER, and APER flag in SSR to 0. Also be sure to read the RDR register in the overrun error processing. If the SCR0.RE bit is set to 0 during reception to forcibly terminate the reception operation, unread receive data may be remaining in the RDR register. In this case, read the RDR register.

Figure 33.102 shows a sample flowchart for serial data reception.



[0] Set the SCR0.RE and RIE bits to 1 at initialization.

[1][2]Receive error processing:
If a receive error occurs, after performing the appropriate error processing, be sure to ORER flag is 0. Reception cannot be resumed if ORER flag is 1.

[3] Read the receive data in the RDR register once in the receive data full interrupt (RXI) request handling routine.

[4] Serial reception continuation procedure:
To continue serial reception, before the MSB (bit 7) of the current frame is received, read data from RDR register in the RXI interrupt handling routine. The RDR data can also be read by activating the DMAC or DTC by the receive data full interrupt (RXI) request.

[5] Overrun error processing:
Read the RDR register.

[6] Clearing the error flag:
Set the flag clear bit corresponding to the reception error to 1. Done this step and step [5], the next frame can be received normally.

[7] Confirming that the error flag is cleared:
Confirm all error flags are cleared.

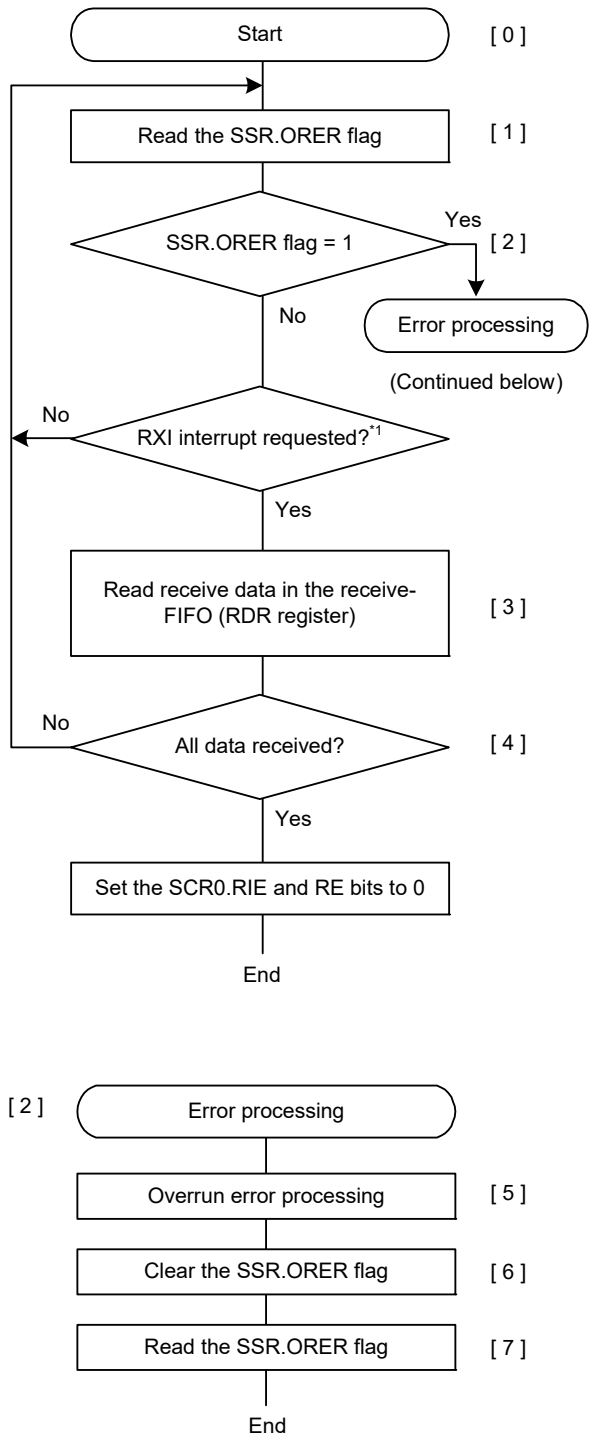
Figure 33.102 Example Flowchart of Serial Reception in Clock Synchronous Mode (Non-FIFO Mode)

(2) FIFO Mode

Figure 33.103 shows an example of serial data reception flowchart in clock synchronous mode with FIFO enabled. The RSCI operates as follows during serial data reception. Reception-only operation is possible only in slave mode. (In master mode, reception-only operation is prohibited.)

1. When the SCR0.RE bit is set to 1, the RTSn# pin output turns low (when the RTS function is used).
2. The RSCI starts receiving data in synchronization with input or output of the sync clock, and transfers the received data to the receive FIFO (RDR register).
3. When an overrun error occurs, the SSR.ORER flag is set to 1. When the SCR0.RIE bit = 1 at this time, an ERI interrupt request is generated and the received data is not transferred to the receive FIFO (RDR register)*1.
4. When data is normally received, the received data is transferred to the receive FIFO (RDR register)*1. When the number of receive data stored in the receive FIFO (RDR register) is equal to or more than the threshold value of the receive FIFO, the SSR.RDRF flag is set to 1. When the RIE bit = 1 at this time, an RXI interrupt request is generated. Reading the received data transferred to the receive FIFO (RDR register) in the RXI interrupt handling routine before an overrun error occurs enables continuous reception. When the received data transferred to the receive FIFO (RDR register) is read and the number of data becomes lower than the RTS# output threshold value, the RTSn# pin output becomes low (when the RTS function is used).

Note 1. The RDR.RDAT[8] bit is not used.



[0] Set the SCR0.RE and RIE bits to 1 at initialization.

[1][2]Receive error processing:
If a receive error occurs, after performing the appropriate error processing, be sure to ORER flag is 0. Reception cannot be resumed if ORER flag is 1.

[3] Read the receive data in the receive FIFO (RDR register) in the RXI interrupt handling routine. The reception data stored in the receive FIFO (RDR register) is read until the number of stored data will be below the FCR.RTRG value at least. When the RDR register is read, the SSR.RDRF flag is cleared. You can confirm the number of the reception data in the FIFO by RFSR.R[5:0] bits.

[4] Serial reception continuation procedure:
To continue serial reception, before the overrun error occurs, read data from RDR register. The RDR data can also be read by activating the DMAC or DTC by the receive data full interrupt (RXI) request.

Note 1. Set all received data to an integer multiple of the receive FIFO threshold. If it is not an integral multiple, an RXI interrupt is not generated and it will become an infinite loop, so set a timeout time separately using a timer etc.

[5] Overrun error processing:
Read the RDR register.

[6] Clearing the error flag:
Set the flag clear bit corresponding to the reception error to 1. Done this step and step [5], the next frame can be received normally.

[7] Confirming that the error flag is cleared:
Confirm all error flags are cleared.

Figure 33.103 Example Flowchart of Serial Reception in Clock Synchronous Mode (FIFO Mode)

33.10.6 Simultaneous Serial Data Transmission and Reception (Clock Synchronous Mode)

In clock synchronous mode, transmission and reception are simultaneously performed, so the number of transmitted data and the number of received data are the same.

(1) Non-FIFO Mode

Figure 33.104 shows an example of serial data concurrent transmission/reception flowchart in clock synchronous mode.
After the RSCI is initialized, perform the following procedure for serial data concurrent transmission/reception.
When switching mode from transmission to concurrent transmission/reception, check that the SSR.TEND flag is set to 1 to ensure that the RSCI is in the transmission complete state. Then set SCR0.TE bit = 0 and RE bit = 0 and then set the TE, RE, TIE, and RIE bits in SCR0 register to 1 simultaneously by a single instruction.
When switching mode from reception to concurrent transmission/reception, check that the RSCI is in the reception complete state, and then set SCR0.TE bit = 0 and RE bit = 0. After that, check that the error flags (ORER, AFER, and APER flags) in SSR are cleared to 0, and then set the TE, RE, TIE, and RIE bits in SCR0 register to 1 simultaneously by a single instruction.
When the RTS function is used in the concurrent transmission/reception operation, if you want to prevent the RTSn# pin output from turning to low after the final data is received as in the reception operation, clear the RE and TE bits in SCR0 register to 0 simultaneously, and then read the RDR register.

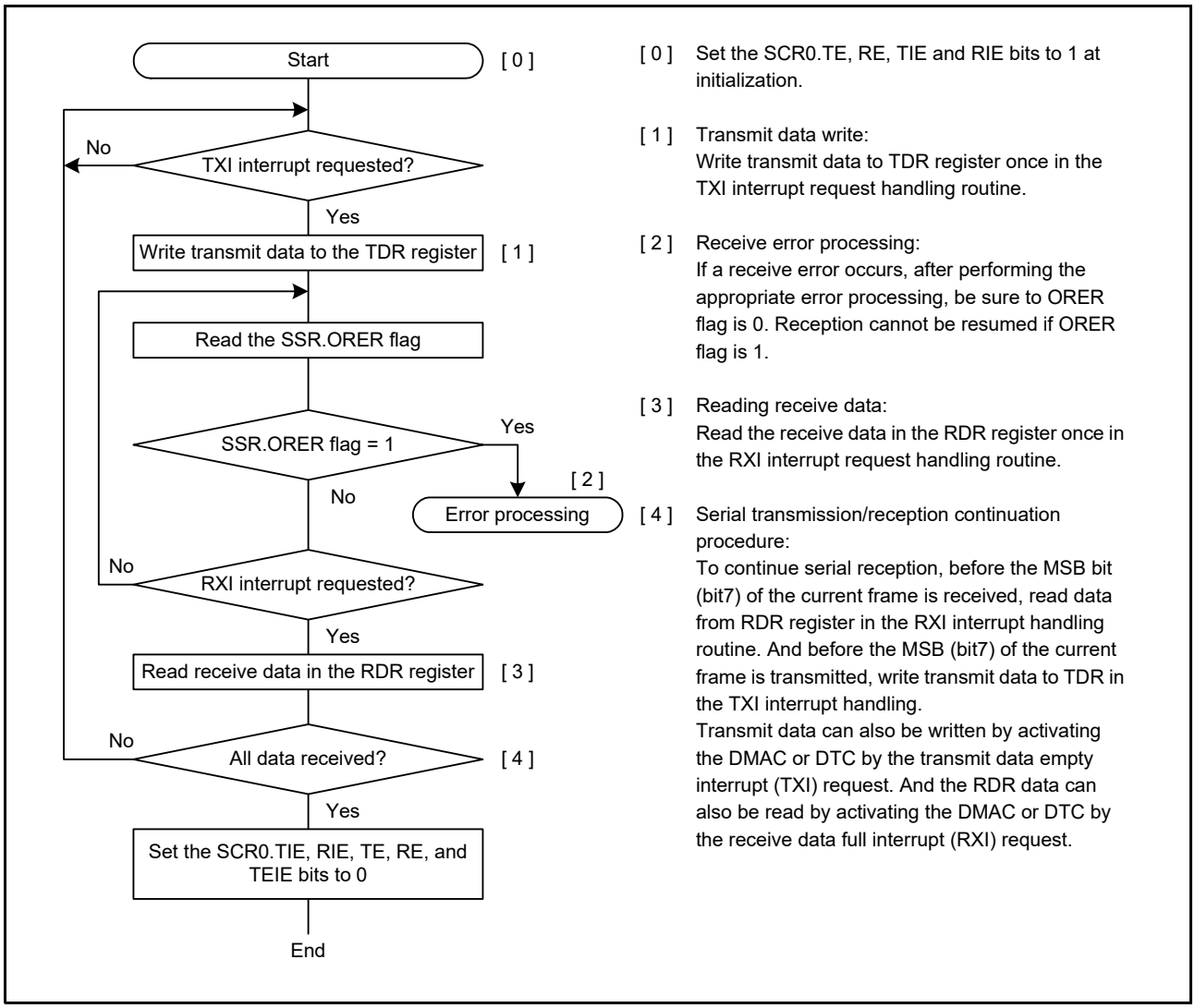


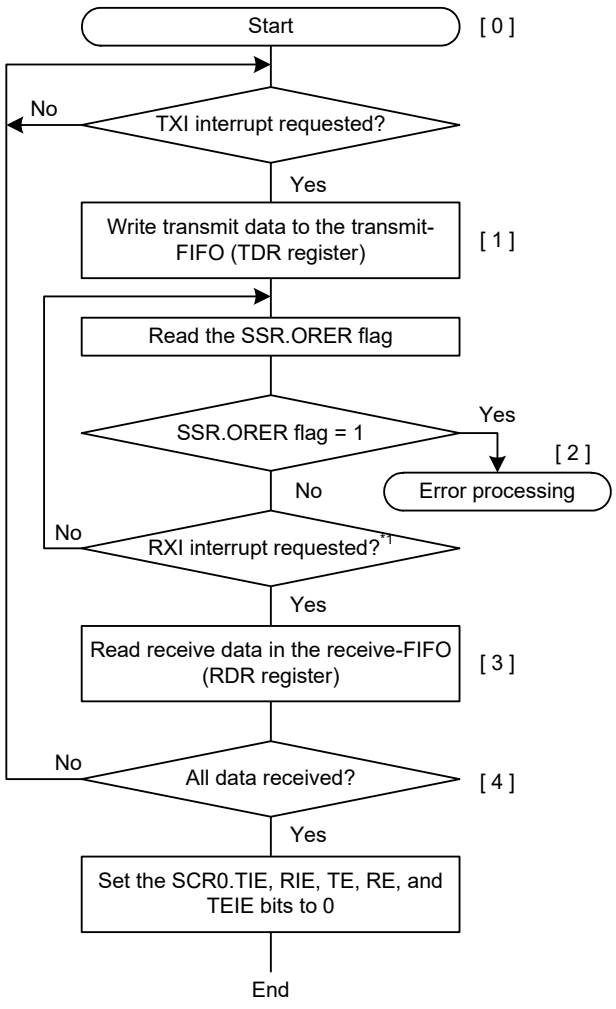
Figure 33.104 Example Flowchart of Simultaneous Serial Transmission and Reception in Clock Synchronous Mode (Non-FIFO Mode)

(2) FIFO Mode

Figure 33.105 shows an example of serial data concurrent transmission/reception flowchart in clock synchronous mode with FIFO enabled.

After the RSCI is initialized, perform the following procedure for serial data concurrent transmission/reception.When switching mode from transmission to concurrent transmission/reception, check that the SSR.TEND flag is set to 1 to ensure that the RSCI is in the transmission complete state. Then set SCR0.TE bit = 0 and RE bit = 0 and then set the TE, RE, TIE, and RIE bits in SCR0 register to 1 simultaneously by a single instruction.

When switching mode from reception to concurrent transmission/reception, check that the RSCI is in the reception complete state, and then set SCR0.TE bit = 0 and RE bit = 0. After that, check that the error flags (ORER, AFER, and APER flags) in SSR are cleared to 0, and then set the TE, RE, TIE, and RIE bits in SCR0 to 1 simultaneously by a single instruction.



Note 1. Set the number of all received data to an integer multiple of the receive FIFO threshold. If it is not an integral multiple, an RXI interrupt is not generated and it will become an infinite loop, so set a timeout time separately using a timer etc.

[0] Set the SCR0.TE, RE, TIE and RIE bits to 1 at initialization.

[1] Transmit data write:
Write transmit data to transmit FIFO (TDR register) in the TXI Interrupt request handling routine. The number of data to be written should match the number specified in the receive FIFO threshold.

[2] Receive error processing:
If a receive error occurs, read the SSR.ORER flag, perform the appropriate error processing such as stopping communication and reading all received data to clear the SSR.RDRF flag, and then set the ORER flag to 0. Reception cannot be resumed if ORER flag is 1.

[3] Reading receive data:
Read the same number of received data stored in the receive FIFO (RDR register) as the number specified in the receive FIFO threshold.

[4] Serial transmission/reception continuation procedure:
To continue serial reception, before the overrun error occurs, read data from the receive FIFO (RDR register) and clear SSR.RDRF flag. To continue serial transmission, write transmit data to the transmit FIFO (TDR) by using a TXI interrupt and clear TDRE flag to 0. Transmit data can also be written to the transmit FIFO (TDR register) by activating the DMAC or DTC by the transmit data empty interrupt (TXI) request. And The RDR data can also be read by activating the DMAC or DTC by the receive data full interrupt (RXI) request. In this case, the TDRE and RDRF flags are cleared automatically, so please don't operate the TDRE and RDRF flags.

Figure 33.105 Example Flowchart of Simultaneous Serial Transmission and Reception in Clock Synchronous Mode (FIFO Mode)

33.10.7 Reception Sampling Timing Adjustment Function in Clock Synchronous Mode with Internal Clock Used

When the clock synchronous mode with internal clock is used (master mode), MRCLK is used as a reception sampling clock.

This function adjusts the reception sampling timing by delaying MRCLK by 1 to 4 PCLK cycles and adding a digital delay. MRCLK's analog delay cannot be adjusted by this function.

Setting the SCR4.RTADJ bit to 1 enables this function. The delay value is set in SCR4.RTMG[3:0] bits.

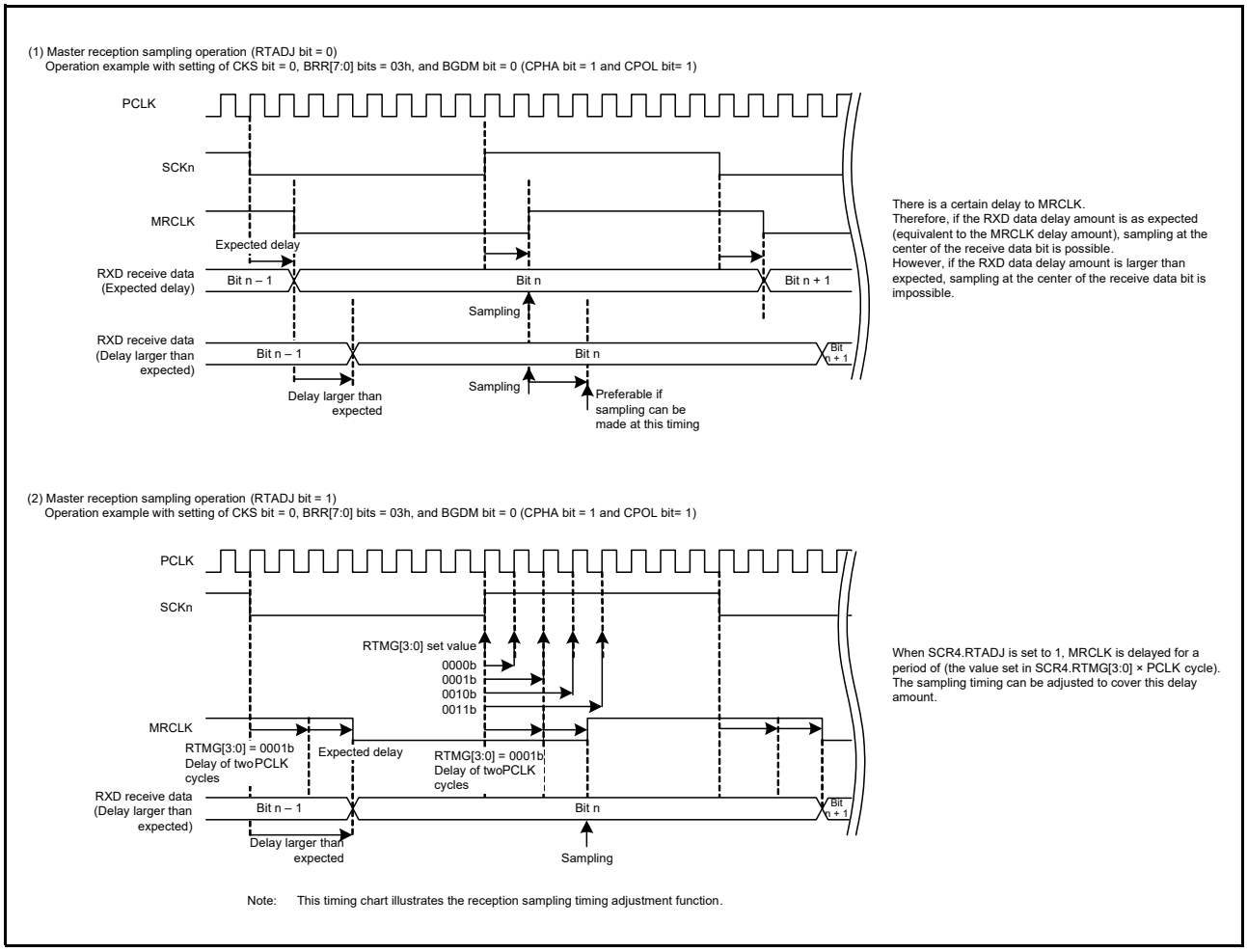


Figure 33.106 Reception Sampling Timing Adjustment Operation in Clock Synchronous Mode (Master) and Simple SPI Mode (Master)

33.11 Operation in Simple SPI Mode

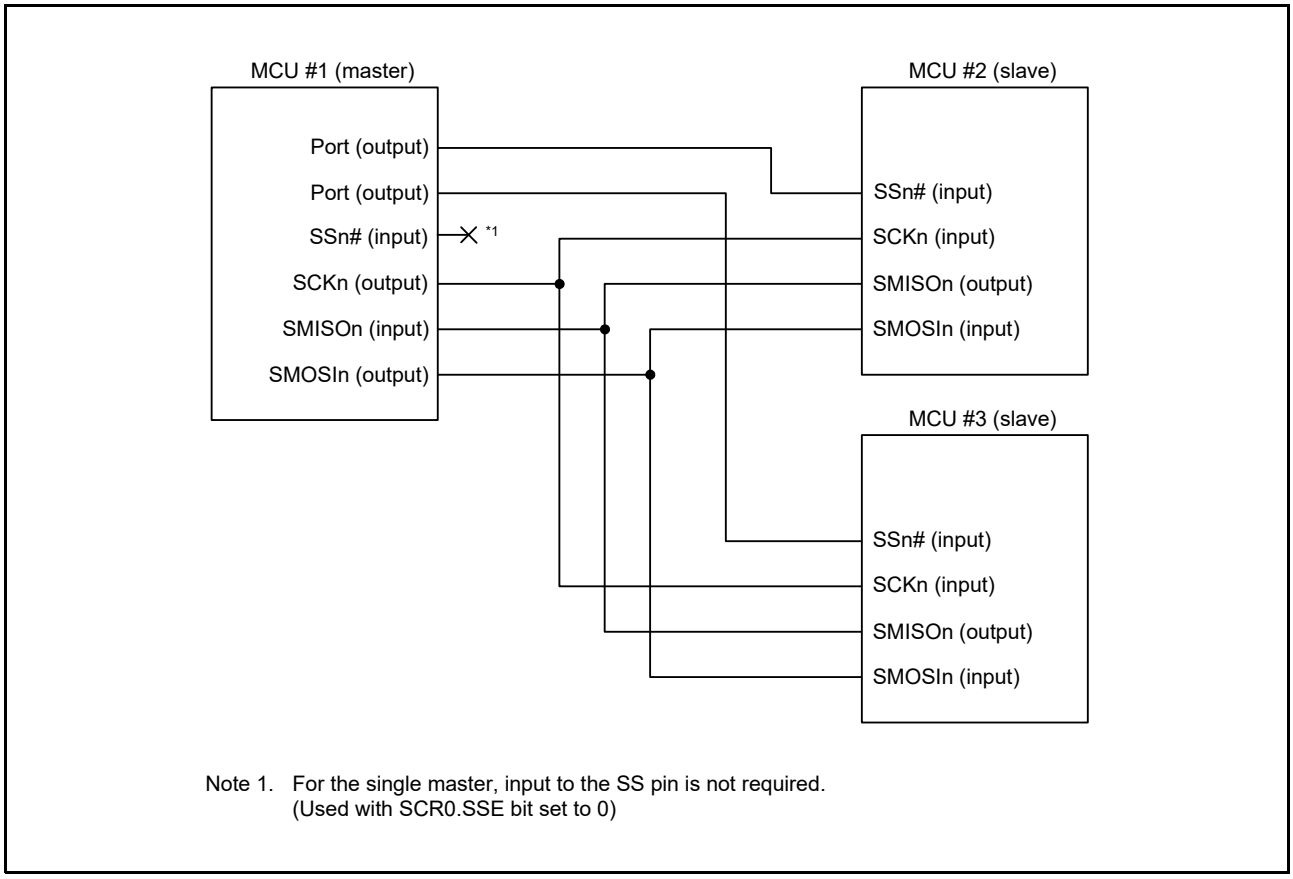
As an extended function, the RSCI supports a simple SPI mode that handles transfer among one or multiple master devices and multiple slave devices.

Making the settings for simple SPI mode (SCR3.MOD[2:0] bits = 011b) plus setting the SSE bit in the SCR0 register to 1 places the RSCI in simple SPI mode. However, the SS pin function on the master side is unnecessary for connection of the device used as the master in simple SPI mode when the configuration only has a single master, so set the SSE bit in the SCR0 to 0 in such cases.

Figure 33.107 shows an example of connections for simple SPI mode.

In simple SPI mode, data are transferred in synchronization with clock pulses in the same way as in clock synchronous mode. One character of data for transfer consists of 8 bits of data, and parity bits cannot be appended to this.

Since the receiver and transmitter are independent of each other within the RSCI module, full-duplex communications are possible, with a common clock signal. Furthermore, since both the transmitter and receiver have a double-buffered structure, writing of further transmit data while transmission is in progress and reading of previously received data while reception is in progress are both possible. Continuous transfer is thus possible.



Note 1. For the single master, input to the SS pin is not required.
(Used with SCR0.SSE bit set to 0)

Figure 33.107 Example of Connections via a Simple SPI Mode

33.11.1 States of Pins in Master and Slave Modes

In simple SPI mode, input and output directions of each pin vary depending on master mode (SCR3.CKE[1:0] bits = 00b or 01b) and slave mode (SCR3.CKE[1:0] bits = 10b or 11b).
Table 33.41 lists the states of pins according to the mode and the level on the SSn# pin.

Table 33.41 States of Pins by Mode and Input Level on the SSn# Pin

Mode	Input on SSn# Pin	State of SMOSIn Pin	State of SMISOn Pin	State of SCKn Pin
Master mode*1	High level (transfer can proceed)	Output for data transmission*2	Input for received data	Clock output*3
	Low level (transfer cannot proceed)	High-impedance	Input for received data (but disabled)	High-impedance
Slave mode	High level (transfer cannot proceed)	Input for received data (but disabled)	High-impedance	Clock input (but disabled)
	Low level (transfer can proceed)	Input for received data	Output for data transmission*2	Clock input

Note 1. When there is only a single master (SCR0.SSE bit = 0), transfer is possible regardless of the input level on the SSn# pin (this is equivalent to input of a high level on the SSn# pin). Since the SSn# pin function is not required, the pin is available for other purposes.
Note 2. The SMISOn pin is in the high-impedance state when serial transmission is disabled (SCR0.TE bit = 0).
Note 3. The SCKn pin is in the high-impedance state when serial transmission is disabled (SCR0.TE bit = 0 and RE bit = 0) in a multi-master configuration (SCR0.SSE bit = 1).

33.11.2 SS Function in Master Mode

Setting the SCR3.CKE[1:0] bits to 00b or 01b enables master mode.
In single master mode (SSE bit = 0), the SSn# pin is not used and data transmission and reception are enabled regardless of the SSn# pin input level. The SSn# pin is available for other purposes.
When in multi-master mode (SSE bit = 1) and the SSn # pin input is high, the master outputs a clock from the SCKn pin and performs transmission and reception operations. And outputting clock indicates “There are no other masters” or “Another master is not performing reception or transmission”. When the SSn# pin input level is low in multi-master mode (SSE bit = 1), this means that another master exists and it is performing data transmission/reception. At this time, the RSCI makes the TXDn pin output and the SCKn pin output high impedance and does not start data transmission/reception. In addition, the SSR.MFF flag is set to 1 as a mode fault. In multi-master mode, read this flag bit to perform the error processing. If a mode fault occurs during the transmission/reception operation, the SCKn pin and the TXDn pin output are made high impedance while the SSn# pin input level is low. In this case, any of TXI, RXI, and TEI interrupts occurs.
Control the SS signal output in master mode with a general-purpose port.

33.11.3 SS Function in Slave Mode

Setting the SCR3.CKE[1:0] bits to 10b or 11b enables slave mode.
When the SSn# pin input level is high, the RXDn pin output becomes high impedance and the clock input from the SCKn pin is ignored. When the SSn# pin input level is low, the clock input from the SCKn pin becomes effective, enabling data transmission and reception.
When the SSn# pin input changes from low to high level during the transmission/reception operation, the RXDn pin output is made high impedance and the transmission/reception operation is immediately suspended. If the transmission is in progress, the SSR.TEND flag will not be set, a transmit end interrupt will not be output, and an abnormal stop status will occur. So, do not negate the SSn # pin during slave transmission/reception. If an abnormal stop occurs, set SCR0.RE bit and SCR0.TE bit to 0 to stop transmission/reception. To resume transmission/reception, set SCR0.RE bit and SCR0.TE bit to 1 after at least 3 PCLK cycles.

33.11.4 Relationship between Clock and Transmit/Receive Data

The clock to be used for data transmission/reception is selectable from four types using the SCR3.CPOL and CPHA bits. Figure 33.108 shows the relationship between clock and transmit data/receive data. The same relationship applies to master mode and slave mode.

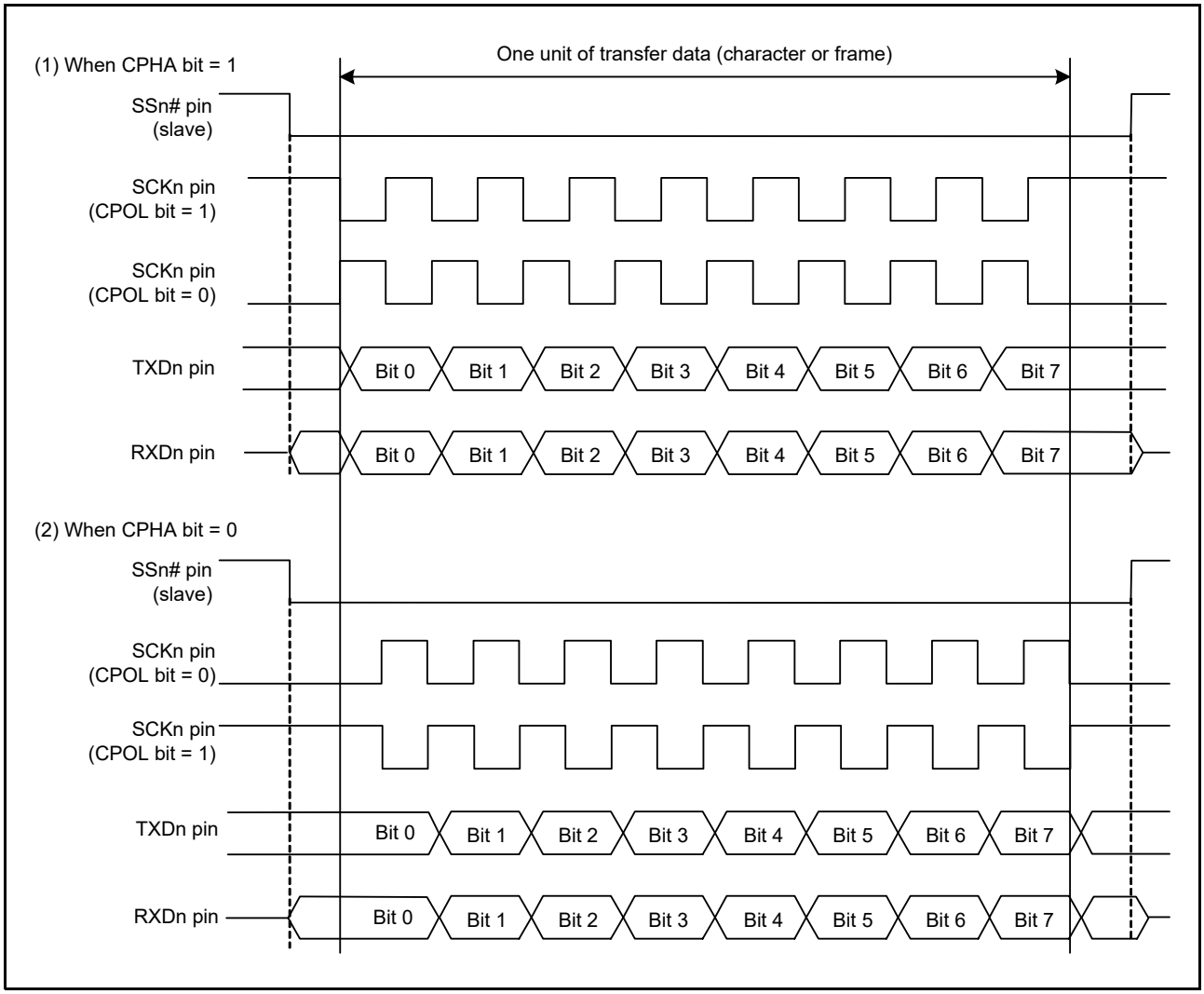


Figure 33.108 Relation between Clock Signal and Transmit/Receive Data in Simple SPI Mode

33.11.5 RSCI Initialization (Simple SPI Mode)

The RSCI can be initialized using the same initialization procedure as for clock synchronous mode (Figure 33.94, Example of RSCI Initialization Flowchart (Clock Synchronous Mode)). The master devices and slave devices use the same clock type selected by the SCR3.CPOL and CPHA bits.
Before performing initialization or changing operating mode or communication format, be sure to stop communication (SCR0.RE bit = 0 and SCR0.TE bit = 0).
Note that setting the RE bit to 0 does not initialize the ORER, AFER, and APER flags in SSR register and the RDR register.
Note that, when SCR0.TIE bit = 1, setting the TE bit to 1 from 0 generates a TXI interrupt.

33.11.6 Transmission and Reception of Serial Data (Simple SPI Mode)

In master mode, set the SSn# pin of the destination slave device to low level before starting data transmission/reception and set to high level after the end of data transmission/reception. In multiple master operation with SCR0.SSE bit = 1 even in master mode, a mode fault will occur if the SSn# pin goes low. Therefore, make sure that no mode fault has occurred before starting communication, and start communication, and make sure that no mode fault has occurred even after communication ends. If a mode fault has occurred, communication may be incomplete, so measures such as retransmission are required. The other procedures are the same as in clock synchronous mode.
In slave mode, it operates according to the SSn# pin input level. Other steps are the same as those of clock synchronous mode.

33.11.7 Reception Sampling Timing Adjustment Function in Simple-SPI Mode with Internal Clock Used

The reception sampling timing adjustment function in simple SPI mode is the same as the reception sampling timing adjustment function in clock synchronous mode. For the description of operation, see section 33.10.7, Reception Sampling Timing Adjustment Function in Clock Synchronous Mode with Internal Clock Used.

33.12 Bit Rate Modulation Function

Using the bit rate modulation function, the bit rate can be corrected by evenly enabling the clocks of the number specified in the SCR2.MDDR[7:0] bits among 256 clock cycles of internal clocks which is selected by the CKS[1:0] bits in SCR2 register.
Figure 33.109 shows an example where the PCLK is selected by the CKS[1:0] bits in SCR2 register and the BRR[7:0] bits and MDDR[7:0] bits are set to 0 and 160 respectively in asynchronous mode. In this example, the cycle of the base clock is evenly corrected (256/160) and the bit rate is also corrected (160/256). Note that there is an imbalance in enabling the internal clock, and expansion and contraction occur in the pulse width of the base clock.

Note: Do not use this function in clock synchronous mode, simple-SPI mode, smart card Interface mode, manchester mode and extended serial mode.

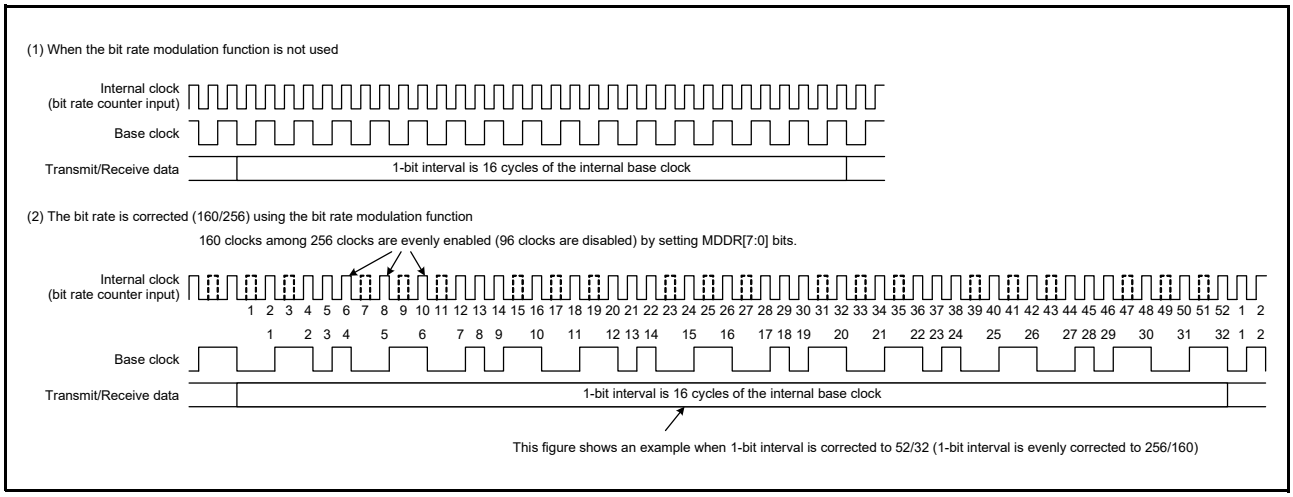


Figure 33.109 Example of Base Clock when Using the Bit Rate Modulation Function

33.13 Noise Cancellation Function

Figure 33.110 shows the configuration of the noise filter used for noise cancellation. The noise filter consists of a two-stage flip-flop circuits and a match detection circuit. When the input signals of the noise filter and the output signals of the two-stage flip-flop circuits completely match, the matched level is conveyed as an internal signal. Unless otherwise matched, the previous value is retained. (When the same level is retained for three cycles or longer on the sampling clock of the noise filter, it is considered as a valid receive signal. A change in pulse for three cycles or shorter is considered as a noise, not as a receive signal).

In asynchronous mode, manchester mode and extended serial mode, the noise cancellation function can be applied to the receive signal input to the RXDn pin. The sampling period of the noise filter can be selected from the base clock period and the divided clock of the baud rate generator clock source by SCR1.NFCS[2:0] bits.

(When SCR1.NFCS[2:0] bits = 000b, SCR2.ABCS bit = 0 and SCR2.ABCSE bit = 0, the cycle is 1/16 of a period 1 transfer bit.

When SCR1.NFCS[2:0] bits = 000b, SCR2.ABCS bit = 1 and SCR2.ABCSE bit = 0, the cycle is 1/8 of a period 1 transfer bit.

When SCR1.NFCS[2:0] bits = 000b, SCR2.ABCSE bit = 1, the cycle is 1/6 of a period 1 transfer bit.)

In simple I^2C mode, the noise elimination function can be used for the input pins of TXDn/SSDAn and RXDn/SSCLn. The sampling period of the noise filter can be selected from the divided clock of the baud rate generator clock source by SCR1.NFCS[2:0] bits.

If the base clock is stopped once with the noise filter enabled and then the base clock input is restarted again, the noise filter operation resumes from the state where the clock was stopped. When SCR0.TE and SCR0.RE bits are set to 0 during base clock input, all of the noise filter flip-flop values are initialized to 1. Accordingly, if the input data is 1 when reception operation resumes, it is determined that a level match is detected and is conveyed as an internal signal. When the level being input corresponds to 0, the initial output of the noise filter is retained until the level matches in three consecutive sampling cycles.

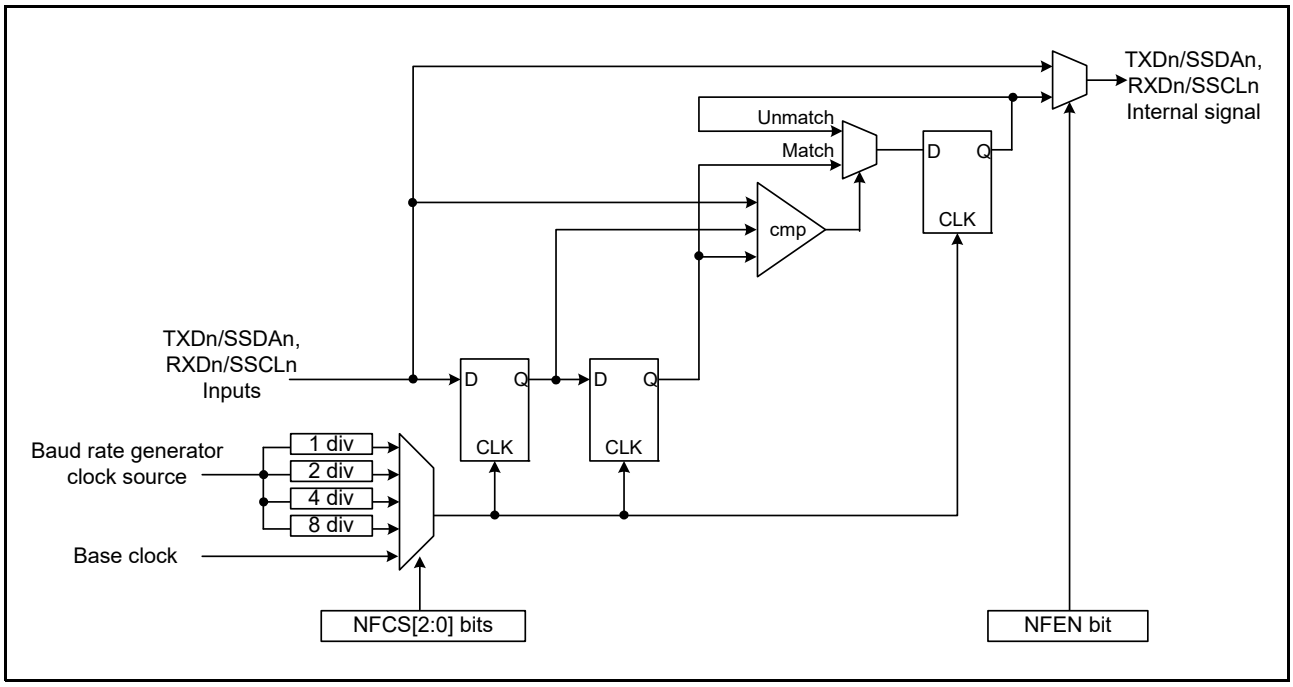


Figure 33.110 Block Diagram of Digital Noise Filter

33.14 RS-485 Driver Control Function

Setting the DEEN bit in the RSCI control register 3 (SCR3) to 1 enables the RS-485 driver control function and generates a DE (Driver Enable) signal that enables the external transceiver transmission mode. The DE signal outputs a valid level for the period with setup time and hold time added before and after data transmission. The DE signal valid level is set by the DELVL bit in the DE signal control register (DECR).

The setup time is the time from when the DE signal is valid until the start bit starts. Set by DESU[4:0] bits of DE signal control register (DECR).

The hold time is the time from the end of the last stop bit of the transmitted message to the invalidation of the DE signal. Set with DEHLD[4:0] bits of the DE signal control register (DECR).

DESU[4:0] and DEHLD[4:0] bits are expressed in base clock units (1/6, 1/8, or 1/16 bit time). For details, refer to section 33.2.13, DE Signal Control Register (DECR).

When this function is used (DEEN bit = 1), the TEND set timing and TEI interrupt output timing are at the end of the DE signal hold time.

When transmission is completed and the next transmission data is not written before the DE signal is negated, the DE signal is negated once. If the timing for writing the next transmit data is not in time, the DE signal is negated and asserted again, the setup time is inserted, and the next data is transmitted. If you want to perform the next transmission with the DE signal asserted, write the next transmission data to the TDR quickly enough.

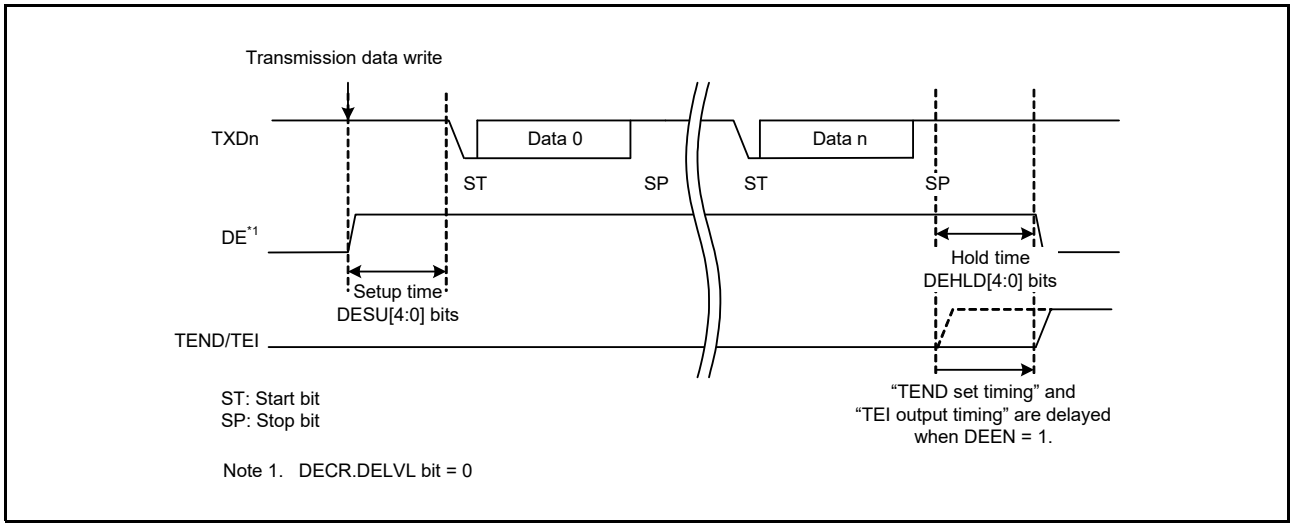


Figure 33.111 Image Waveform for RS-485 Driver Control DE Signal Output

33.15 Loopback Function

The loopback function can be used in Asynchronous mode with the internal clock, and manchester mode with the internal clock, and Clock synchronous mode with the internal clock.
When 1 is written to the LOOP bit in the SCR1 register, RSCI blocks the external input (RXD) path and connects the output path of the transmit data register and the input path of the receive data register.
When this function is used with TINV bit = 1, inversion of transmission data becomes reception data. However, this function can be used with TINV bit = 1 only when operating in clock synchronous mode internal clock.
Table 33.42 shows the relationship between the TINV and LOOP bit settings and the received data.

Table 33.42 TINV and LOOP Bit Settings and Received Data

			Communication Mode		
TINV	**LOOP**	**Receive Data**	**Async Internal Clock**	**Manchester Internal Clock**	**Clock Sync Internal Clock**
0 or 1	0	Receive data from RXDn pin	Possible	Possible	Possible
0	1	Transmit data	Possible	Possible	Possible
1	1	Inverted transmit data	Impossible	Impossible	Possible

Figure 33.112 shows the configuration of the shift register input/output path in loopback mode.

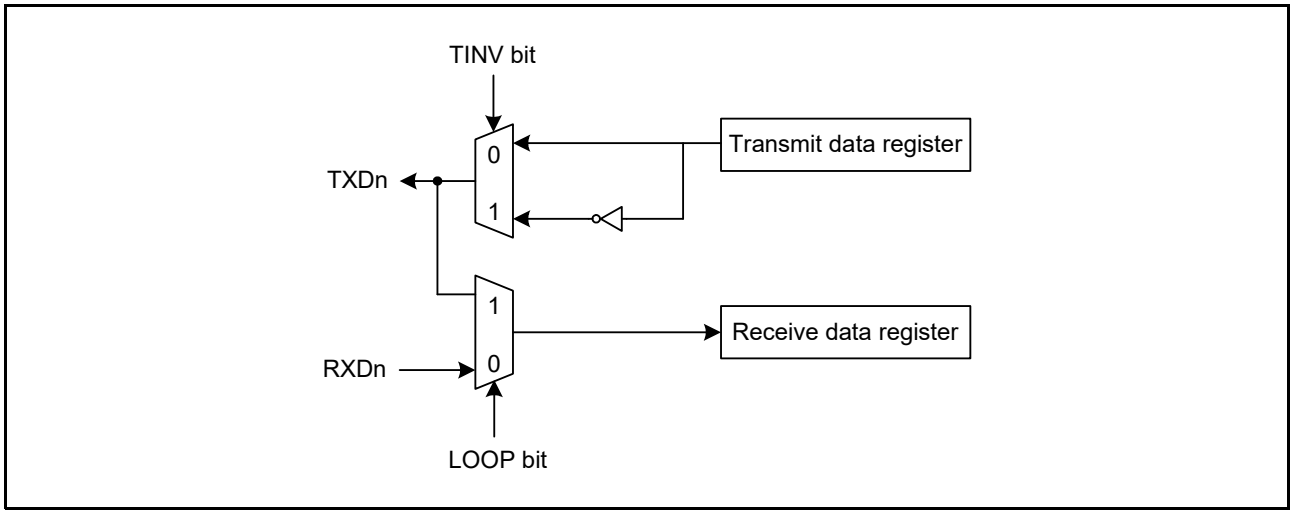


Figure 33.112 Shift Register Input/Output Configuration Image in Loopback Mode

33.16 Half-Duplex Communication Function

Do not use the half-duplex communication function in simple I^2C mode, simple-SPI mode and smart card interface mode.

In other communication modes, if the SCR1.HDSEL bit is set to 1, half-duplex communication using the TXDn pin is possible. When half-duplex communication is used, transmission and reception must be performed exclusively. Transmission and reception settings (SCR0.TE bit = 1 and SCR0.RE bit = 1) is prohibited.

However, if half-duplex communication is performed as the master reception in clock synchronous mode, perform transmission/reception settings (SCR0.TE bit = 1 and SCR0.RE bit = 1) and perform dummy transmission. By dummy transmission (arbitrary transmission data is written to TDR), SCKn is output and reception is enabled. The dummy transmission data is discarded inside the RSCI and is not actually transmitted.

During half-duplex communication, the only communication port terminal used is the TXDn pin. Output when SCR0.TE bit = 1, input when SCR0.TE bit = 0.

33.17 Interrupt Signal

Table 33.43 lists RSCI interrupt signals.

The interrupt explanation corresponding to each operation mode is described in sections 33.17.2 to 33.17.5. Also, TXI and RXI have an interrupt buffer function. Refer to section 33.17.1, Buffer Operations for TXI and RXI Interrupts. When performing transmission and reception using DTC or DMAC, be sure to set DTC or DMAC first, and then enable RSCI before setting. Refer to section 18, Data Transfer Controller (DTCb) and section 17, DMA Controller (DMACAa) for how to set DTC or DMAC.

Table 33.43 RSCI Interrupt List

Interrupt Symbol	Interrupt Function	Pulse/ Level	Pulse Width	Active Level	SYNC Clock	Note
ERI	Error interrupt Bus collision detection interrupt	Level	—	Low	PCLK	
RXI	Simple I²C: Reception end interrupt Other mode: Receive data full interrupt	Pulse	1 cycle	Low	PCLK	
TXI	Simple I²C and smart card interface: Transmit end interrupt Other mode: Transmit data empty interrupt, Break Field transmission completion	Pulse	1 cycle	Low	PCLK	
TEI	Simple I²C: Completion of generation of a start, restart, or stop condition (STI) Other mode: Transmit end interrupt	Level	—	Low	PCLK	
AED	Active edge detection interrupt	Pulse	1 cycle	Low	PCLK	
BFD	Break Field detection interrupt	Level	—	Low	PCLK	Only when extended serial mode

33.17.1 Buffer Operations for TXI and RXI Interrupts

The TXI and RXI interrupts have an interrupt buffer function. When the first interrupt request is generated during interrupt handling and the next interrupt request is generated (when the status flag of the interrupt controller (ICU) is 1), the RSCI does not output the interrupt request, and holds it internally. The interrupt that can be held is up to one.

33.17.2 Interrupt in Asynchronous Mode, Manchester Mode, Clock Synchronous Mode, and Simple SPI Mode

A different interrupt vector is assigned to each interrupt source, and individual interrupt sources can be enabled or disabled with the enable bits in SCR0 register.

(1) Non-FIFO Mode

Table 33.44 lists interrupt sources in asynchronous mode, manchester mode, clock synchronous mode, and simple SPI mode with FIFO disabled.

If the SCR0.TIE bit is 1, a TXI interrupt request is generated when transmit data is transferred from the TDR register to the TSR register. At the start of transmission, a TXI interrupt request can also be generated by using a single instruction to set the SCR0.TE and SCR0.TIE bit to 1 at the same time. A TXI interrupt request can activate the DTC or DMAC to handle data transfer.

A TXI interrupt request is not generated by setting the SCR0.TIE bit to 1 while the setting of the SCR0.TE bit is 1.*1

When the SCR0.TEIE bit is 1, the SSR.TEND flag becomes 1 and the TEI interrupt request is generated if the next data is not written to the TDR register by the timing to transmit the last bit of transmission data. In addition, the TEND flag holds 1 during the period from setting the SCR0.TE bit to 1 until writing transmit data to the TDR register, and if the TEIE bit is set to 1, a TEI interrupt request is generated.

Writing data to the TDR register clears the TEND flag and cancels the TEI interrupt request, but it takes time to cancel it.

If the SCR0.RIE bit is 1, an RXI interrupt request is generated when received data is stored in the RDR register. An RXI interrupt request can activate the DTC or DMAC to handle data transfer.

Setting of any from among the ORER, AFER, and APER flags in the SSR register or the MCER, SYER (if SYERIE = 1)*2, PFER (if PFERIE = 1)*2, and SBER (if SBERIE = 1)*2 flags in MMSR register to 1 while the SCR0.RIE bit is 1 leads to the generation of an ERI interrupt request.

An RXI interrupt request is not generated at this time. Clearing all flags (ORER, AFER, APER, MCER, SYER (if SYERIE = 1)*2, PFER (if PFERIE = 1)*2, and SBER (if SBERIE = 1)*2) leads to discarding of the ERI interrupt request.

Note 1. To temporarily prohibit TXI interrupts at the time of transmission of the last of the data and so on when you wish a new round of transmission to start after handling of the transmit end interrupt, control prohibiting and permitting of the interrupt by using the interrupt request enable bit in the interrupt controller rather than using the TIE bit. This can prevent the suppression of TXI interrupt requests in the transfer of new data.

Note 2. In manchester mode only, MMSR.SYER (if SYERIE bit = 1), PFER (if PFERIE bit = 1), SBER (if SBERIE bit = 1) flags are added to the ERI interrupt sources.

(2) FIFO Mode

Table 33.45 lists interrupt sources in asynchronous mode, manchester mode, clock synchronous mode, and simple SPI mode with FIFO enabled.

If the SCR0.TIE bit is 1, a TXI interrupt request is generated when the stored number of data in transmit FIFO becomes equal to or less than the transmit FIFO threshold. A TXI interrupt request can also be generated by using a single instruction to set the SCR0.TE and SCR0.TIE bit to 1 at the same time. A TXI interrupt request is not generated by setting the SCR0.TE bit to 1 while the setting of the SCR0.TIE bit is 0 or by setting the SCR0.TIE bit to 1 while the setting of the SCR0.TE bit is 1.

If SCR0.TEIE bit is 1, when the next data isn't being written in transmit FIFO by the timing to which the last bit of the transmission data is sent, SSR.TEND flag will be 1 and TEI interrupt request is generated.

If the SCR0.RIE bit is 1, RXI interrupt request is generated when the stored number of data in receive FIFO exceeds the threshold. When the threshold value is set to 0, RXI interrupt request is occurred if the quantity of data in receive FIFO is greater than or equal to 1.

If the SCR0.RIE bit is 1, when SSR.ORER flag is set to 1 or the data a framing error or a parity error generated is stored in receive FIFO, ERI interrupt request is generated.

When the number of data stored in a receive FIFO at this time is a threshold value or above, RXI interrupt request is also

generated. The ERI interrupt request can be canceled by clearing all flags (SSR.ORER, AFER, APER).

Table 33.44 RSCI Interrupt Sources with FIFO Disabled

Name	Interrupt Source	Interrupt Flag	Interrupt Enable	DTC/DMAC Activation
ERI	Receive error	ORER, AFER, APER, DFER, DPER, MCER, SYER (SYERIE = 1), PFER (PFERIE = 1), SBER (SBERIE = 1)	RIE	Not possible
RXI	Receive data full	RDRF	RIE	Possible
	Receive data match	DCMF		
TXI	Transmit data empty	TDRE	TIE	Possible
	TE = 0 → 1 detection			
TEI	Transmit end	TEND	TEIE	Not possible

Table 33.45 RSCI Interrupt Sources with FIFO Enabled

Name	Interrupt Source	Interrupt Flag	Interrupt Enable	DTC/DMAC Activation
ERI	Receive error	ORER, AFER, APER, DFER, DPER, DR (FCR.DRES bit = 1)	RIE	Not possible
RXI	Receive FIFO data full	RDRF	RIE	Possible
	Receive data ready	DR (FCR.DRES bit = 0)		
	Receive data match	DCMF		
TXI	Transmit FIFO data empty	TDRE	TIE	Possible
	TE = 0 → 1 detection			
TEI	Transmit end	TEND	TEIE	Not possible

33.17.3 Interrupt in Smart Card Interface Mode

Table 33.46 lists interrupt sources in smart card interface mode. A transmit end interrupt (TEI) request cannot be used in this mode.

Table 33.46 RSCI Interrupt Sources in Smart Card Interface Mode

Name	Interrupt Source	Interrupt Flag	Interrupt Enable	DTC/DMAC Activation
ERI	Receive error or error signal detection	ORER, APER, ERS	RIE	Not possible
RXI	Receive data full	RDRF	RIE	Possible
TXI	Transmit end	TNED	TIE	Possible
	TE = 0 → 1 detection			

Data transmission/reception using the DTC or DMAC is also possible in smart card interface mode. In transmission operation, when the SSR.TEND flag is set to 1, a TXI interrupt request is generated. This TXI interrupt request activates the DTC or DMAC allowing transfer of transmit data if the TXI request is specified beforehand as a source of DTC or DMAC activation. The TEND flag is automatically set to 0 when the DTC or DMAC transfers the data.
If an error occurs, the RSCI automatically retransmits the same data. During the retransmission, the TEND flag is kept to 0 and the DTC or DMAC is not activated. Therefore, the RSCI and DTC or DMAC automatically transmit the specified number of bytes, including retransmission in the case of error occurrence. However, the SSR.ERS flag is not automatically cleared to 0 at error occurrence. Therefore, the ERS flag must be cleared by previously setting the SCR0.RIE bit to 1 to enable an ERI interrupt request to be generated at error occurrence.
In reception operation, an RXI interrupt request is generated when receive data is set to RDR register. This RXI interrupt request activates the DTC or DMAC allowing transfer of receive data if the RXI request is specified beforehand as a source of DTC or DMAC activation. If an error occurs, the error flag is set. Therefore, the DTC or DMAC is not activated and an ERI interrupt request is issued to the CPU instead; the error flag must be cleared.
When transmitting/receiving data using the DTC or DMAC, be sure to make settings to enable the DTC or DMAC before making RSCI settings. For DTC or DMAC settings, refer to section 18, Data Transfer Controller (DTCb) and section 17, DMA Controller (DMACAa).

33.17.4 Interrupts in Simple I2C Mode

Table 33.47 lists RSCI interrupts in Simple I2C mode.

The STI interrupt is allocated to the transmit end interrupt (TEI) request. The receive error interrupt (ERI) request cannot be used.

The DTC or DMAC can also be used to handle transfer in simple I2C mode.

When the value of the SIMR.IICINTM bit is 1, a RXI request will be generated on the falling edge of the SSCLn signal for the eighth bit. If the RXI has been set up as an activating request for the DTC or DMAC beforehand, the RXI request will activate the DTC or DMAC to handle transfer of the received data. Furthermore, a TXI request is generated on the falling edge of the SSCLn signal for the ninth bit (acknowledge bit). If the TXI has been set up as an activating request for the DTC or DMAC beforehand, the TXI request will activate the DTC or DMAC to handle transfer of the transmit data. (In this case, ACK/NACK judging are impossible.)

When the value of the SIMR.IICINTM bit is 0, RSCI moves as follows. RXI request (ACK detection) is generated if the input on the SSDAn pin is at the low level on the rising edge of the SSCLn signal for the ninth bit (acknowledge bit). TXI request (NACK detection) is generated if the input on the SSDAn pin is at the high level on the rising edge of the SSCLn signal for the ninth bit (acknowledge bit). If the RXI has been set up as an activating request for the DTC or DMAC beforehand, the RXI request will activate the DTC or DMAC to handle transfer of the received data. Also, if the DTC or DMAC is used for data transfer in reception or transmission, be sure to set up and enable the DTC or DMAC before setting up the RSCI.

When the SIMR.IICSTAREQ, IICRSTAREQ, and IICSTPREQ bits are used to generate a start condition, restart condition, or stop condition, the STI request is issued when generation is complete.

Table 33.47 RSCI Interrupt Sources in Simple I2C Mode

Name	Interrupt Source		Interrupt Flag	Interrupt Enable	DTC/DMAC Activation
	IICINTM bit = 1	IICINTM bit = 0			
RXI	Reception end	—	—	RIE	Possible*1
	—	ACK detection	—		Possible
TXI	Transmit end	—	—	TIE	Possible*1
	—	NACK detection	—		Possible
STI	Completion of generation of a start, restart, or stop condition		IICSTIF	TEIE	Not possible

Note 1. If the DMAC or DTC are being used, you can not confirm whether ACK or NACK.

33.17.5 Interrupts in Extended Serial Mode

Table 33.48 lists interrupt sources in extended serial mode.

Table 33.48 RSCI Interrupt Sources in Extended Serial Mode

Name	Interrupt Source	Interrupt Flag	Flag the needs to be confirmed	Interrupt Enable	DTC/DMAC Activation
ERI	Error	ORER, AFER, APER	—	RIE	Not possible
		BCDF		BCDIE	
		COF		RIE, COFIE	
RXI	Reception data full	RDRF	CF0MF, CF1MF, PIBDF	RIE	XSR0.SFSF flag = 0: Possible XSR0.SFSF flag = 1: Not possible
AED	Active edge detection	AEDF	—	AEDIE	Possible
TXI	Transmit data empty	TDRE	—	TIE	Possible
	TE = 0 → 1 detection				
	Break Field transmission completion	BFOF	—	TIE, BFOIE	
TEI	Transmit end	TEND	—	TEIE	Not possible
BFD	Break Field detection	BFDF	—	BFDIE	Not Possible (Unnecessary)

In extended serial mode, in addition to reception errors (overrun, framing, and parity errors), an ERI interrupt request is output when a bus conflict is detected during transmission, or when a counter overflow of the extended serial module occurs. At this time, a RXI interrupt request is not output. The ERI interrupt request can be canceled by clearing all the flags.

When transmitting Start Frame, if SCR0.TIE bit = 1 and XCR0.BFOIE bit = 1, a TXI interrupt request is output when Break Field transmission is completed. When Control Field 0 data is written to the TDR register, data transmission starts. Therefore, transmission using DTC or DMAC is possible.

Set SCR0.TEIE bit = 1 after writing the last transmit data to the TDR register and transmission starts.

During Start Frame reception (XSR0.SFSF flag = 1), reception using DTC or DMAC by RXI interrupt is not possible. Check the SSR register and XSR0 register, check the reception status (See Figure 33.72), and then clear the flag.When data is received, read the RDR register so that an overrun error does not occur (if you do not need to check the received data value, clear the RDRF flag without reading the RDR register). When reception of Control Field 1 is completed (XSR0.CF1MF flag = 1), Start Frame detection is disabled (XSR0.SFSF flag = 0) and reception using DTC or DMAC is possible. Be sure to read the RDR register.

When Start Frame/Break Field detection is enabled (XCR1.SDST bit = 1), if a Break Field longer than the period set in XCR2.BFLW[15: 0] bits is received, the BFDF flag is set and a BFD interrupt request is output. Then RSCI becomes the Start Frame reception state. Clear the BFDF flag.

When Start Frame/Break Field detection is enabled (XCR1.SDST bit = 1) and the bit rate measurement function is enabled (XCR1.BRME bit = 1), an AED interrupt factor is output when an active edge is detected. Read the timer count capture value (XSR1.CCV[15:0] bits).

33.18 Event Linking

By employing interrupt request signals as event signals, RSCI can provide linked operation through the event link controller (ELC) for modules selected in advance.
Event signals can be output regardless of the values of the corresponding interrupt request enable bits. And even when the next interrupt source occurs while the corresponding interrupt status flag is 1, the event signal can be output.
All event output is a pulsed output and becomes negated behind 1 PCLK of asserting.
Table 33.49 lists RSCI event link signals.

Table 33.49 RSCI Event Link Signal List

Function	Pulse/Level	Pulse Width	Active Level	Synchronize Clock
Error event	Pulse	1 cycle	High	PCLK
Receive data full event	Pulse	1 cycle	High	PCLK
Transmit data empty event	Pulse	1 cycle	High	PCLK
Transmit end event	Pulse	1 cycle	High	PCLK

(1) Error Event Output (Receive Error, Error Signal Detection)

- Indicates abnormal termination due to a parity error during reception.
- Indicates abnormal termination due to a framing error during reception.
- Indicates abnormal termination due to an overrun error during reception.
- Indicates abnormal termination due to a manchester code error during reception (only in manchester mode).
- Indicates that a preface error occurred upon reception and abnormal termination occurred (only in manchester mode and MMCR.PFERIE bit = 1).
- Indicates that a start bit error occurred during reception and abnormal termination occurred (only in manchester mode and MMCR.SBERIE bit = 1).
- Indicates that a reception sync error occurred during reception and abnormal termination occurred (only in manchester mode and only when MMCR.SYERIE bit = 1).
- Indicates detection of the error signal during transmission in smart card interface mode.
- SSR.AFER and APER flags are 0, and reception data less than receive FIFO data trigger number is set in a receive FIFO buffer, and it indicates that time of 15etu has passed when FIFO enabled and FCR.DRES bit is 1.
- In extended serial mode, indicates that the 16-bit counter in the extended serial module has overflowed.
- In extended serial mode, a bus collision is detected during transmission (SCR0.TE bit = 1).

(2) Receive Data Full Event Output

- In non-FIFO mode, indicates that received data have been set in the receive data register (RDR).
- In FIFO mode, indicates that receive data has been transferred to the receive FIFO (RDR register), and the quantity of data in the register has exceeded the specified receive triggering number. If the receive FIFO threshold is set to 0, no event output occurs unless at least one data is received. However, in FIFO mode, it is prohibited because it causes inconvenience when processing events.
- When FIFO is enabled and FCR.DRES bit is 0, indicates followings.
  - SSR.AFER flag = 0 and SSR.APER flag = 0
  - Time of 15 etu has passed after the reception data less than receive FIFO data trigger number is set in receive FIFO
- Indicates that ACK has been detected if the SIMR.IICINTM bit is 0 in simple I^2C mode.
- Indicates that the 8th-bit SSCLn falling edge has been detected if the SIMR.IICINTM bit is 1 in simple I^2C mode.
- When the SIMR.IICINTM bit is 1 during master transmission in simple I^2C mode, set the event link controller (ELC) so that receive data full events are not used.

(3) Transmit Data Empty Event Output

- Indicates that the SCR0.TE bit has been changed from 0 to 1.
- In non-FIFO mode, indicates that transmit data have been transferred from the transmit data register (TDR) to the transmit shift register (TSR).
- In FIFO mode, indicates that the quantity of data in transmit FIFO (TDR register) has fallen below the specified transmit triggering number. However, in FIFO mode, it is prohibited because it causes inconvenience when processing events.
- Indicates that transmission has been completed in smart card interface mode.
- Indicates that NACK has been detected if the SIMR.IICINTM bit is 0 in simple I^2C mode.
- Indicates that the ninth-bit SSCLn falling edge has been detected if the SIMR.IICINTM bit is 1 in simple I^2C mode.
- In extended serial mode, indicates that Break Field transmission is complete.

(4) Transmit End Event Output

- Indicates the completion of transmission. In FIFO mode, it is prohibited because it causes inconvenience when processing events.
- Indicates that the starting condition, resumption condition, or termination condition has been generated in simple I^2C mode.
- In smart card interface mode, the transmit end event is not output.

33.19 Usage Notes

33.19.1 Setting the Module Stop Function

Module stop control register D (MSTPCRD) is used to stop and start RSCI operations. With the value after a reset, RSCI operations are stopped. The registers of the modules only become accessible after release from the module stop state. For details, refer to section 11, Low Power Consumption.

33.19.2 RSCI Operations during Low Power Consumption State

(1) Transmission

Before using the power consumption reduction function to reduce RSCI's power consumption, please do the following to confirming transmission end (SSR.TEND flag = 1):

- Set the output terminal state after transmission operation is stopped by SCR1.SPB2DT and SPB2IO bits.
- Stop the transmission (SCR0.TIE bit = 0, TE bit = 0, TEIE bit = 0)

When transitions to these states are made during transmission, the data being transmitted become indeterminate.
To transmit data in the same operating mode after cancellation of the low power consumption state, set the TE bit to 1, read SSR register, and write data to TDR sequentially to start data transmission. To transmit data with a different operating mode, initialize the RSCI first.
To start transmission using the DMAC/DTC after cancellation from software standby mode, set the TE and TIE bit to 1 simultaneously. The TXI interrupt is generated and transmission starts using the DMAC/DTC.
Figure 33.113 shows a sample flowchart for transition to software standby mode during transmission. Figure 33.114 and Figure 33.115 show the port pin states during transition to software standby mode.

(2) Reception

Before specifying the module stop state or making a transition to software standby mode, stop the receive operations (SCR0.RE bit = 0). If transition is made during data reception, the data being received will be invalid.
Figure 33.116 shows a sample flowchart for reception to software standby mode during reception.

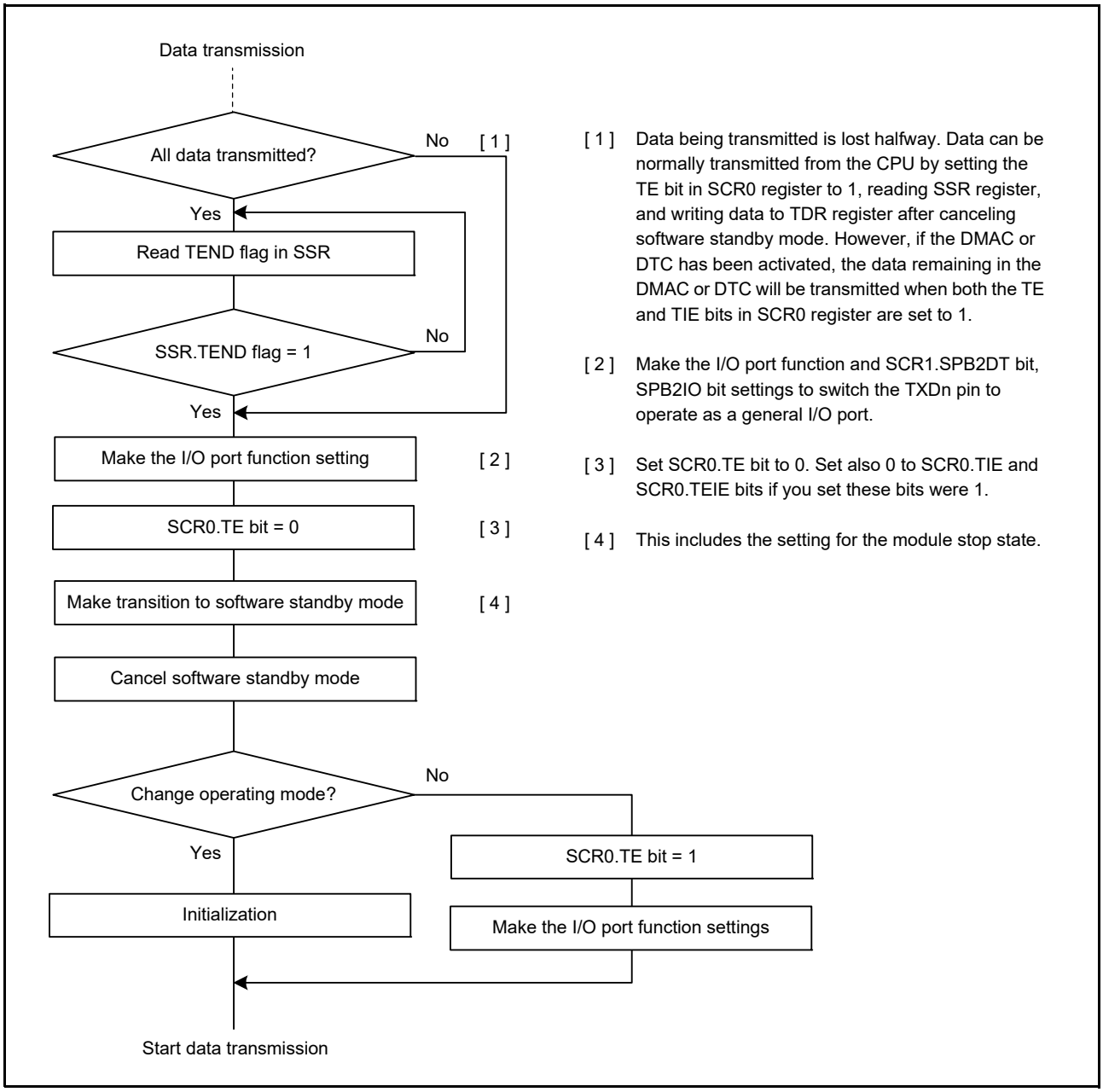


Figure 33.113 Example of Flowchart for Transition to Software Standby Mode during Transmission

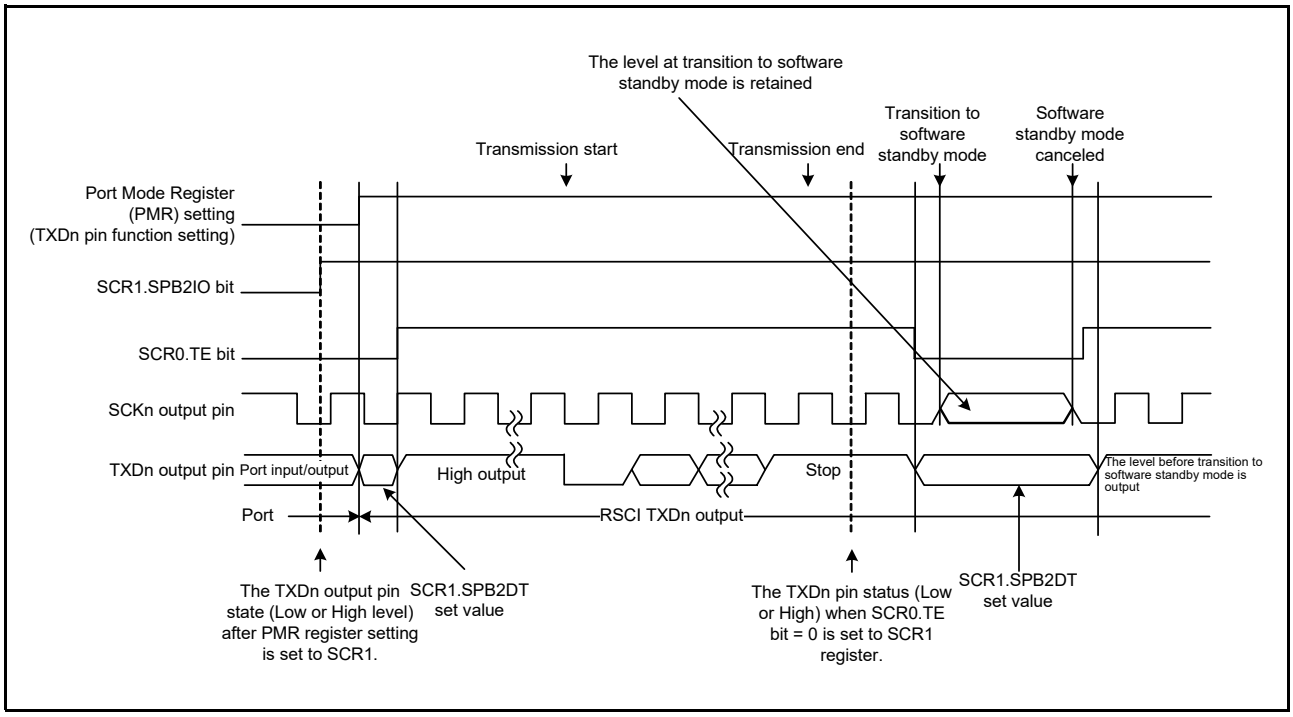


Figure 33.114 Port Pin States during Transition to Software Standby Mode (Internal Clock, Asynchronous Transmission)

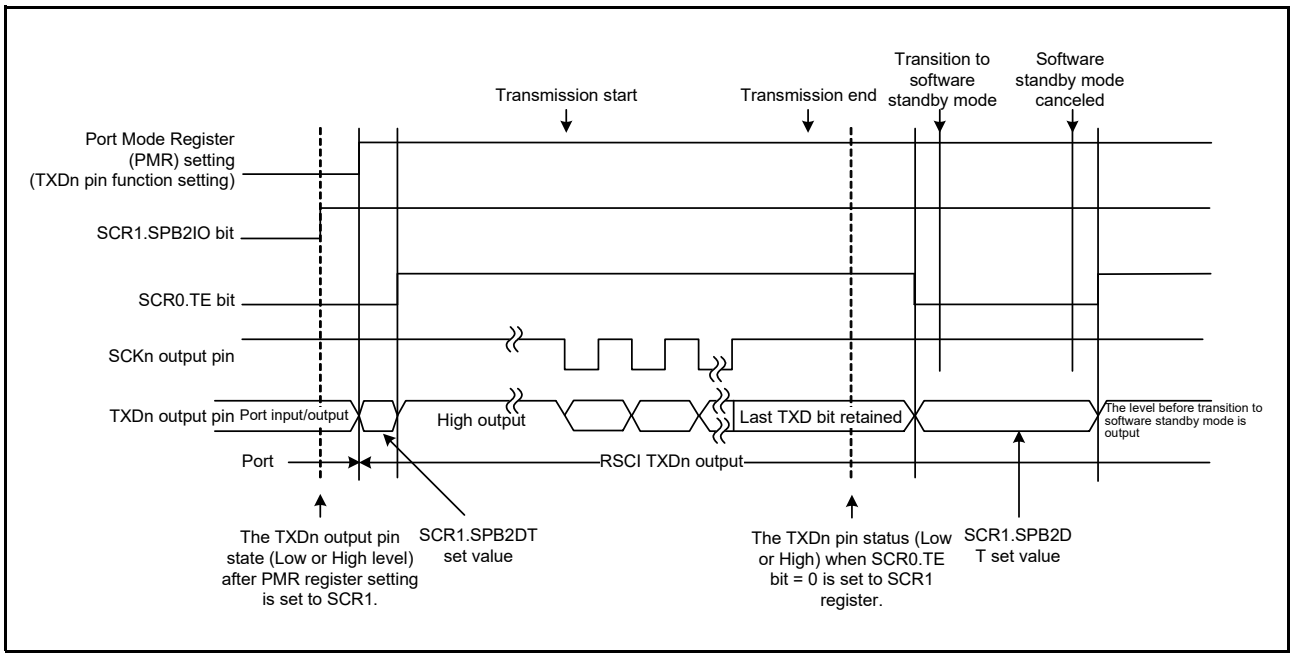


Figure 33.115 Port Pin States during Transition to Software Standby Mode (Internal Clock, Clock Synchronous Transmission)

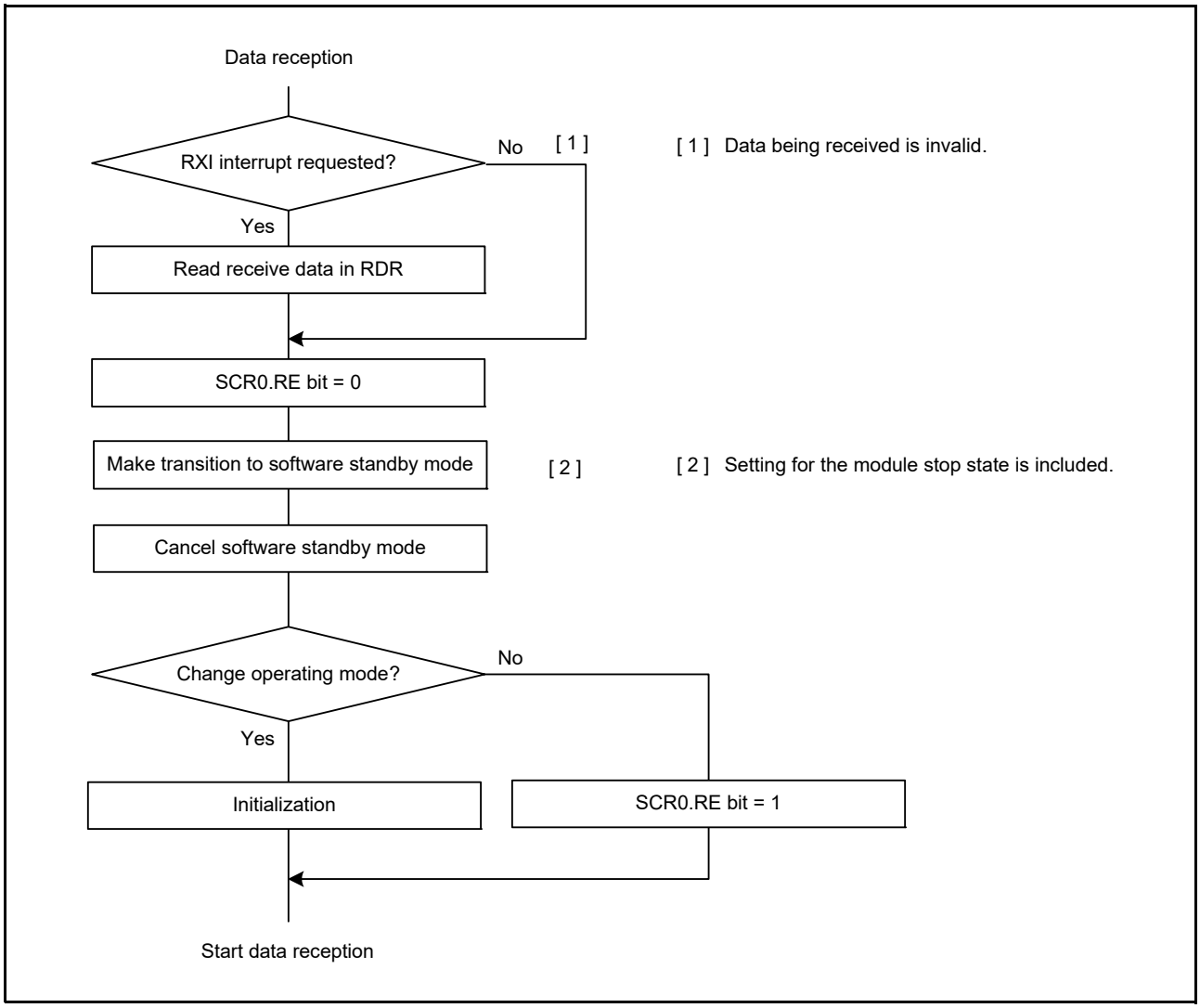


Figure 33.116 Example of Flowchart for Transition to Software Standby Mode during Reception

33.19.3 Break Detection and Processing

(1) Non-FIFO Mode

When a framing error is detected, a break can be detected by reading SSR.RXDMON flag value. In a break, the input from the RXDn pin becomes all low, and so the SSR.AFER flag is set to 1 (framing error has occurred), and the SSR.APER flag may also be set to 1 (parity error has occurred). When the SCR3.RXDESEL bit is 0, the RSCI continues the receive operation even after a break is received. Therefore, note that even if the AFER flag is set to 0 (no framing error occurred), it will be set to 1 again. When the SCR3.RXDESEL bit is 1, the RSCI sets the SSR.AFER flag to 1 and stops receiving operation until a start bit of the next data frame is detected. If the SSR.AFER flag is set to 0 at this time, the SSR.AFER flag retains 0 during the break. When the RXDn pin becomes high and the break ends, detecting the start bit at the first falling edge of the RXDn pin allows the RSCI to start the receiving operation.

(2) FIFO Mode

After a framing error is detected, when RSCI detects that continuous receive data is low for 1 frame, the data stored into the receive FIFO (RDR register) and reception stops. When a framing error is detected, a break can be detected by reading SSR.RXDMON flag value. After the RXD signal is in the mark state and it has finished the break, a stock of reception data to the receive FIFO (RDR register) resumes.

33.19.4 Mark State and Sending Breaks

When the SCR0.TE bit is 0 (serial transmission is disabled), the state of the TXDn pin can be set by SCR1.SPB2IO bit and SCR1.SPB2DT bit. Using this, it's possible to do a TXDn pin in the mark state and send a break out. When you want to make a communication-line is in the mark state (the state of 1) until the SCR0.TE bit is set to 1 (serial transmission is enabled). First, set it as High-level output by setting SPB2IO bit and SPB2DT. Next, it's changed to a TXDn pin by I/O port function. On the other hand, if you want to send a break when sending data, set the SCR0.TE bit to 0 after setting the SPB2IO and SPB2DT bits in the SCR1 register to low level output. Setting the TE bit to 0 initializes the transmitter regardless of the current transmission status.

33.19.5 Receive Error Flags and Transmit Operations (Clock Synchronous Mode and Simple SPI Mode)

Transmission can be start by writing transmit-data to TDR register even if SSR.ORER flag is 1. However, reception can not be started. Note also that the receive error flags cannot be set to 0 even if the SCR0.RE bit is set to 0 (serial reception is disabled).

33.19.6 Writing Data to the TDR Register

(1) Non-FIFO Mode

Data can be written to TDR register anytime when TE bit is 1. However, if new data is written to TDR register when transmit data is remaining in TDR register, the previous data in TDR register is lost because it has not been transferred to TSR register yet. If you use DTC or DMAC, be sure to write transmit data to TDR register in the TXI interrupt request handling routine.

(2) FIFO Mode

Data can be written to transmit FIFO (TDR register) when TE bit is 1. Check the number of writable data with the TFSR.T[5:0] bits.

33.19.7 Restrictions on Clock Synchronous Transmission (Clock Synchronous Mode and Simple SPI Mode)

When the external clock source is used as a synchronization clock, the following restrictions apply.

(1) Start of Transmission

Update TDR register by the CPU, DMAC, or DTC and wait at least the following time until the start of the external clock input: (See Figure 33.117)
Time considering the output AC characteristics of the SMISO pin of this product and the input AC characteristics of the master reception + 1 PCLK cycle.

(2) Continuous Transmission

The next transmit data must be transferred to the TSR register before the falling edge*1 of the transmit clock for bit 7. Please write the transmit data to TDR register with this in mind. If the transmit data can not be written in time, the previous frame data is resent. (See Figure 33.117)

Note 1. When SCR3.CPOL bit = 1 and SCR3.CPHA bit = 0, or SCR3.CPOL bit = 0 and SCR3.CPHA bit = 1. In the case of SCR3.CPOL bit = 0 and SCR3.CPHA bit = 0, or SCR3.CPOL bit = 1 and SCR3.CPHA bit = 1, it's the rising edge.

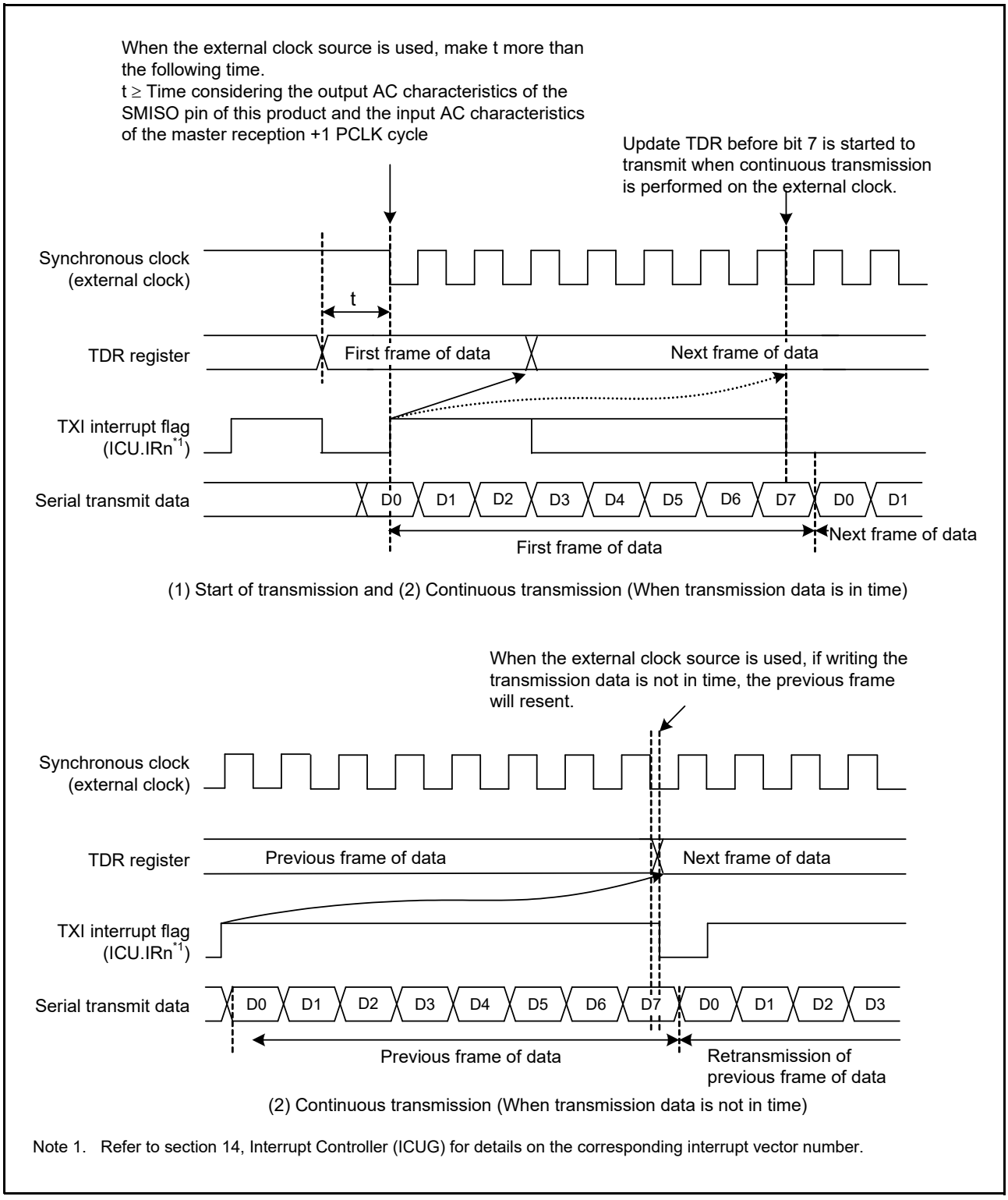


Figure 33.117 Restrictions on Use of External Clock in Clock Synchronous Transmission

33.19.8 Restrictions on Using DMAC or DTC

When using the DMAC or DTC to read RDR register, be sure to set the receive data full interrupt (RXI) as the activation source of the relevant RSCI.
During the operation in transmission/reception using the DMAC or DTC, it should not set transfer information of DMAC/DTC.

33.19.9 Notes on Starting Transfer

At the point where transfer starts when the interrupt status flag (IRn.IR flag) in the interrupt controller is 1, follow the procedure below to clear interrupt requests before permitting operations (by setting the SCR0.TE or SCR0.RE bit to 1). For details on the interrupt status flag, refer to section 14, Interrupt Controller (ICUG).

- Confirm that transfer has stopped (the setting of the SCR0.TE or SCR0.RE bits is 0).
- Set the corresponding interrupt enable bit (SCR0.TIE or SCR0.RIE) to 0.
- Read the corresponding interrupt enable bit (SCR0.TIE or SCR0.RIE bit) to check that it has become 0.
- Set the interrupt status flag (IRn.IR flag) in the interrupt controller to 0.

33.19.10 Limitations on Simple SPI Mode

(1) Master Mode

- Use a resistor to pull up or pull down the clock line matching the initial settings for the transfer clock set by the SCR3.CPHA and CPOL bits when the SCR0.SSE bit is 1. This prevents the clock line from being placed in the high-impedance state when the SCR0.TE bit is set to 0 or unexpected edges from being generated on the clock line when the SCR0.TE bit is changed from 0 to 1.
  In a single-master configuration, pull up or pull down the clock line is not necessary because the clock line does not become high-impedance state when SCR0.SSE bit = 0 and SCR0.TE bit = 0.
- In the case of the setting for clock delay (SCR3.CPHA bit is 0), the receive data full interrupt (RXI) is generated before the final clock edge on the SCKn pin as indicated in Figure 33.118. If the TE and RE bits become 0 at this time before the final edge of the clock signal on the SCKn pin, the SCKn pin is placed in the high-impedance state, so the width of the last clock pulse of the transfer clock is shortened. Furthermore, an RXI interrupt may lead to the input signal on the SSn# pin of a connected slave going to the high level before the final edge of the clock signal on the SCKn pin, leading to incorrect operation of the slave.
- In a multi-master configuration, take care because the SCKn pin output becomes high-impedance while the input on the SSn# pin is at the low level if a mode fault occurs as the current character is being transferred. And stopping supply of the clock signal to the connected slave. Remake the settings for the connected slave to avoid misaligned bits when transfer is restarted.

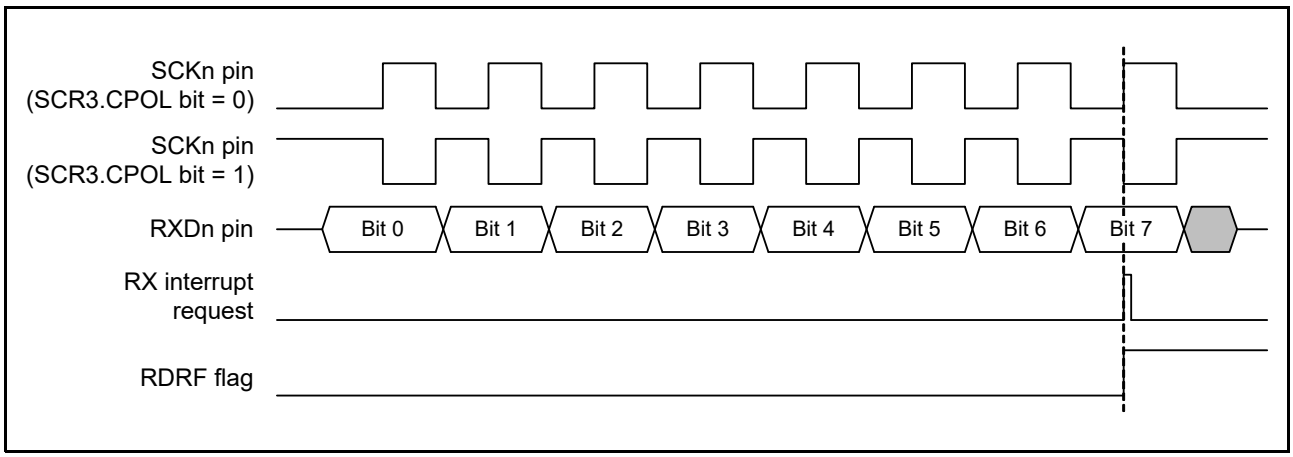


Figure 33.118 Timing of the RXI Interrupt in Simple SPI Mode (with Clock Delay)

(2) Slave Mode

- It takes “1 PCLK + data output delay time (AC characteristics)” from writing the transmit data to the TDR register until the data is output to the RXDn pin. Take these into account when starting external clock input.
- Provide an external clock signal to the master the same as the data length for transfer.
- Secure the SS input setup time (AC characteristics) from the SSn# low-level input to the start of external clock input.
- Control the input on the SSn# pin before the start and after the end of data transfer.
- When the level being input on the SSn# pin is to be changed from low to high while the current character is being transferred, the transmission and reception is stopped immediately. Set the TE and RE bits in the SCR0 register to 0 and, after remaking the settings, restart transfer of the first byte.

33.19.11 Note on Transmit Enable Bit (TE Bit)

In the initial register value, when setting the SCR.TE bit to 0 (serial transmission is disabled) while the pin function is "TXDn", output of the TXDn pin becomes high-impedance.
Prevent the TXDn line from becoming high-impedance by any of the following ways:

(a) Connect a pull-up or pull-down resistor to the TXDn line.
(b) Set the SCR1 register to determine the level of the TXDn pin when the TE bit is 0.

33.19.12 Notes on Extended Serial Mode

In extended serial mode (SCR3.MOD[2:0] bits = 110b), the following functions cannot be used.

- CTS and RTS functions
- Multi-processor communication function
- Bit Rate Modulation function
- Loopback function
- FIFO buffer

33.19.13 Notes on RS-485 Driver Control Function

- RS-485 Driver control function is valid only in Asynchronous mode.
- When RS-485 Driver control function is active (SCR3.DEEN bit = 1), the TEND set timing/TEI output timing changes as follows. Wait for the TEI interrupt and set the TE bit to 0.
  When RS-485 Driver control function is inactive: When STOP bit output is completed.
  When RS-485 Driver control function is active: At the end of DE signal hold time.

33.19.14 Notes on Loopback Function

The Loopback function is valid in Asynchronous mode with internal clock, in manchester mode with internal clock and Clock synchronous mode with internal clock.
It can also operate in the asynchronous HBS support mode, and when the HBSCR.AOE bit = 1, it loops back the signal with the logical AND of the TXDAn and TXDBn pin outputs (Use with TINV bit = RINV bit = 0).

33.19.15 Notes on Aborting Operation

If 0 is written to SCR0.RE bit during data reception and the receive operation is aborted, the status may become invalid depending on the timing. Therefore, do not use the received data (value stored in the RDR register) or the flag value of each status register. To abort the receive operation, disable the interrupt and event link related to reception, and then write 0 to the SCR0.RE bit.

34. I2C-bus Interface (RIICa)

This MCU has a single-channel I2C-bus interface (RIIC0).
The RIIC module conforms with the NXP I2C-bus (Inter-IC bus) interface and provides a subset of its functions.
In this section, “PCLK” is used to refer to PCLKB.

34.1 Overview

Table 34.1 lists the specifications of the RIIC, Figure 34.1 shows a block diagram of the RIIC. Table 34.2 lists the I/O pins of the RIIC.

Table 34.1 RIIC Specifications (1/2)

Item	Description
Communications format	• I2C-bus format or SMBus format • Master mode or slave mode selectable • Automatic securing of the various setup times, hold times, and bus-free times for the transfer rate
Transfer rate	Fast-mode is supported (up to 400 kbps)
Serial clock (SCL)	For master operation, the duty cycle of the SCL is selectable in the range from 4 to 96%.
Generating and detecting conditions	Start, restart, and stop conditions are automatically generated. Start conditions (including restart conditions) and stop conditions are detectable.
Slave address	• Up to three different slave addresses can be set. • 7-bit and 10-bit address formats are supported (along with the use of both at once). • General call addresses, device ID addresses, and SMBus host addresses are detectable.
Acknowledgment	• For transmission, the acknowledgment bit is automatically loaded. Transfer of the next data for transmission can be automatically suspended on detection of a not-acknowledge signal. • For reception, the acknowledgment bit is automatically transmitted. If a wait between the eighth and ninth clock pulses has been selected, software control of the acknowledgment in response to the received data is possible.
Wait function	• During reception, cycles of waiting by holding the SCL line low can be inserted at the following two types of timing: Waiting between the eighth and ninth clock pulses Waiting between the ninth clock pulse and the first clock pulse of the next byte
SDA output delay function	Changes in the timing of the output of data bits for transmission, and of the acknowledgment bit, can be delayed relative to the falling edge of SCL.
Arbitration	• For multi-master operation Clock synchronization for the SCL line in cases of conflict with the SCL signal from another master is possible. When generating the start condition would create conflict on the bus, loss in arbitration is detected by testing for non-matching between the internal data level and the actual level on the SDA line. During master operation, loss in arbitration is detected by testing for non-matching between the actual level on the SDA line and the internal data level. • Loss in arbitration due to detection of the start condition while the bus is busy is detectable (to prevent the generating of double start conditions). • Loss in arbitration in transfer of a not-acknowledge signal due to the internal signal (NACK) and the actual level on the SDA line not matching is detectable. • Loss in arbitration due to non-matching of internal data level and the actual level on the SDA line is detectable in slave transmission.
Timeout function	The internal timeout function is capable of detecting long-interval stop of the SCL.
Noise cancellation	The interface incorporates digital noise filters for both the SCL and SDA signals, and the width for noise cancellation by the filters is adjustable by software.
Interrupt sources	Four sources: • Communication error/communication event Detection of arbitration-lost, NACK, timeout, a start condition including a restart condition, or a stop condition • Receive data full (including matching with a slave address) • Transmit data empty (including matching with a slave address) • Transmission end

Table 34.1 RIIC Specifications (2/2)

Item	Description
Low power consumption function	Module stop state can be set.
RIIC operating modes	• Four Master transmit mode, master receive mode, slave transmit mode, and slave receive mode
Event link function (output)	Four sources (RIIC0): • Communication error/communication event Detection of arbitration-lost, NACK, timeout, a start condition including a restart condition, or a stop condition • Receive data full (including matching with a slave address) • Transmit data empty (including matching with a slave address) • Transmission end

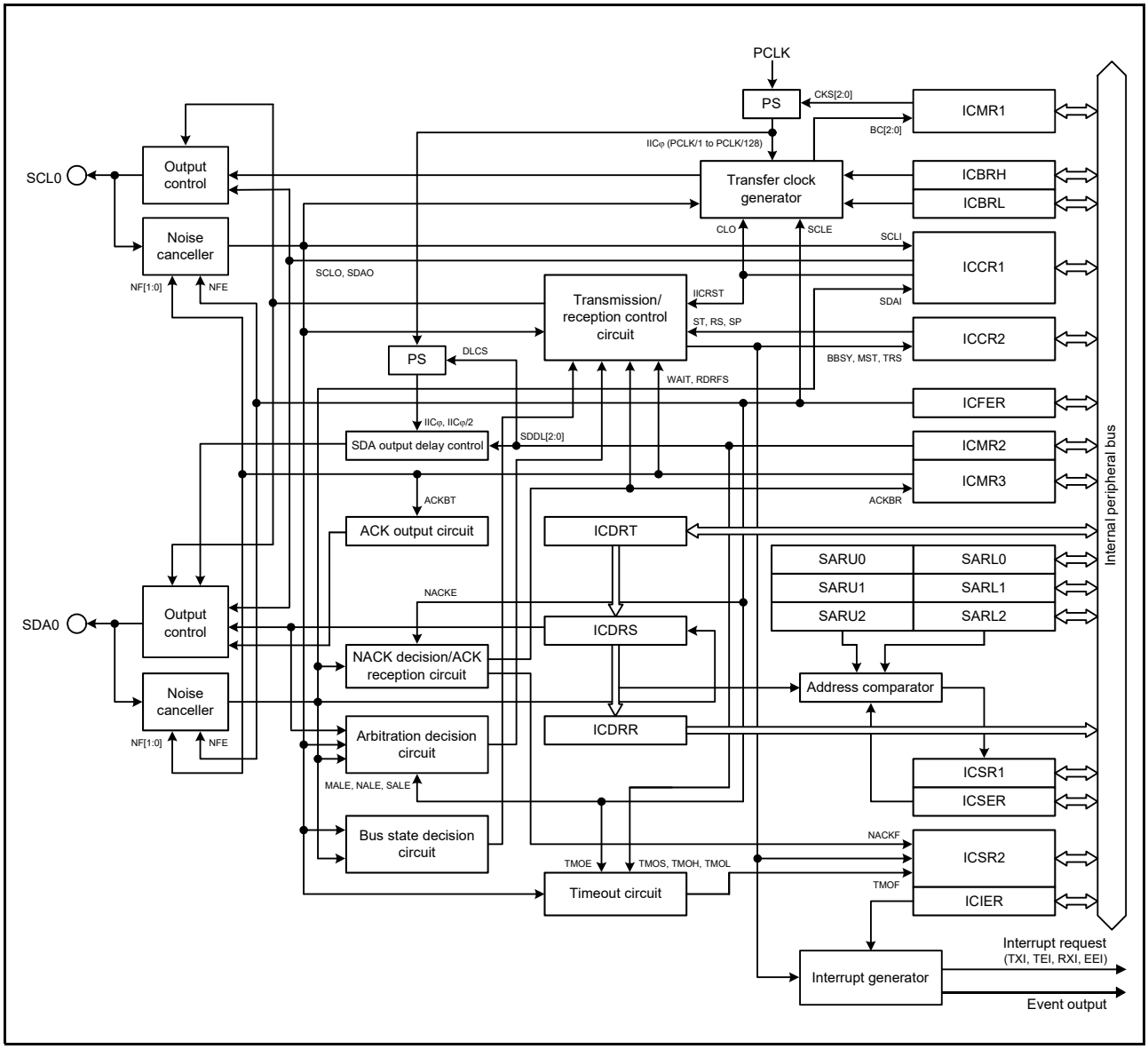


Figure 34.1 RIIC Block Diagram

The logic levels of the input signals for RIIC are CMOS when the I2C-bus is selected (ICMR3.SMBS bit is 0), or TTL when the SMBus is selected (ICMR3.SMBS bit is 1).

Table 34.2 RIIC Pin Configuration

Channel	Pin Name	I/O	Function
RIIC0	SCL0	I/O	RIIC0 serial clock I/O pin
	SDA0	I/O	RIIC0 serial data I/O pin

34.2 Register Descriptions

34.2.1 I2C-bus Control Register 1 (ICCR1)

Address(es): RIIC0.ICCR1 0008 8300h

	b7	b6	b5	b4	b3	b2	b1	b0
	ICE	IICRST	CLO	SOWP	SCLO	SDAO	SCLI	SDAI
Value after reset:	0	0	0	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	SDAI	SDA Line Monitor	0: SDA0 line is low. 1: SDA0 line is high.	R
b1	SCLI	SCL Line Monitor	0: SCL0 line is low. 1: SCL0 line is high.	R
b2	SDAO	SDA Output Control/Monitor	• Read: 0: The RIIC has driven the SDA0 pin low. 1: The RIIC has released the SDA0 pin. • Write: 0: The RIIC drives the SDA0 pin low. 1: The RIIC releases the SDA0 pin. (High level output is achieved through an external pull-up resistor.)	R/W
b3	SCLO	SCL Output Control/Monitor	• Read: 0: The RIIC has driven the SCL0 pin low. 1: The RIIC has released the SCL0 pin. • Write: 0: The RIIC drives the SCL0 pin low. 1: The RIIC releases the SCL0 pin. (High level output is achieved through an external pull-up resistor.)	R/W
b4	SOWP	SCLO/SDAO Write Protect	0: SCLO and SDAO bits can be written. 1: SCLO and SDAO bits are protected. (This bit is read as 1.)	R/W
b5	CLO	Additional SCL Output	0: Does not output an additional SCL (default). 1: Outputs an additional SCL. (The CLO bit is cleared automatically after one clock pulse is output.)	R/W
b6	IICRST	I2C-bus Interface Internal Reset	0: Releases the RIIC reset or internal reset. 1: Initiates the RIIC reset or internal reset. (Clears the bit counter and the SCL0/SDA0 output latch)	R/W
b7	ICE	I2C-bus Interface Enable	0: Disable (SCL0 and SDA0 pins in inactive state) 1: Enable (SCL0 and SDA0 pins in active state) (Combined with the IICRST bit to select either RIIC or internal reset.)	R/W

SDAO Bit (SDA Output Control/Monitor) and SCLO Bit (SCL Output Control/Monitor)

These bits are used to directly control the SDA0 and SCL0 signals output from the RIIC.

When writing to these bits, also write 0 to the SOWP bit.

The result of setting these bits is input to the RIIC via the input buffer. When slave mode is selected, a start condition may be detected and the bus may be released depending on the bit settings.

Do not rewrite these bits during a start condition, stop condition, restart condition, or during transmission or reception. Operation after rewriting under the above conditions is not guaranteed.

When reading these bits, the state of signals output from the RIIC can be read.

CLO Bit (Additional SCL Output)

This bit is used to output an additional SCL for debugging or error processing.
Normally, set the bit to 0. Setting the bit to 1 in a normal communication state causes a communication error.
For details on this function, refer to section 34.11.2, Additional SCL Output Function.

IICRST Bit (I2C-bus Interface Internal Reset)

This bit is used to reset the internal states of the RIIC.
Setting this bit to 1 initiates an RIIC reset or internal reset.
Whether an RIIC reset or internal reset is initiated is determined according to the combination with the ICE bit. Table 34.3 lists the resets of the RIIC.
The RIIC reset initializes all registers and internal states of the RIIC, and the internal reset initializes the bit counter (ICMR1.BC[2:0] bits), the I2C-bus shift register (ICDRS), and the I2C-bus status registers (ICSR1 and ICSR2) as well as the internal states of the RIIC. For the reset conditions for each register, refer to section 34.14, Initialization of Registers and Functions When a Reset is Applied or a Condition is Detected.
An internal reset initiated with the IICRST bit set to 1 during operation (with the ICE bit set to 1) resets the internal states of the RIIC without initializing the port settings and the control and setting registers of the RIIC when the bus or RIIC hangs up due to a communication error.
If the RIIC hangs up in a low level output state, resetting the internal states cancels the low level output state and releases the bus with the SCL0 pin and SDA0 pin at a high impedance.

Note: If an internal reset is initiated using the IICRST bit for a bus hang-up occurred during communication with the master device in slave mode, the states may become different between the slave device and the master device (due to the difference in the bit counter information). For this reason, do not initiate an internal reset in slave mode, but initiate restoration processing from the master device. If it is necessary to perform an internal reset in slave mode, perform it during bus free state. If an internal reset is necessary because the RIIC has hung with the SCL0 line in a low level output state in slave mode, initiate an internal reset and then generate a restart condition from the master device or resume communications from the start condition after having generated a stop condition. If communication is restarted by initiating a reset solely in the slave device without generating a start condition or restart condition from the master device, synchronization will be lost because the master and slave devices operate asynchronously.

Table 34.3 RIIC Resets

IICRST	ICE	State	Specifications
1	0	RIIC reset	Resets all registers and internal states of the RIIC.
	1	Internal reset	Resets the ICMR1.BC[2:0] bits, registers ICSR1, ICSR2, and ICDRS, and the internal states of the RIIC.

ICE Bit (I2C-bus Interface Enable)

This bit selects the active or inactive state of the SCL0 and SDA0 pins. It can also be combined with the IICRST bit to initiate two types of resets. See Table 34.3, RIIC Resets, for the types of resets.
Set the ICE bit to 1 when using the RIIC. The SCL0 and SDA0 pins are placed in the active state when the ICE bit is set to 1.
Set the ICE bit to 0 when the RIIC is not to be used. The SCL0 and SDA0 pins are placed in the inactive state when the ICE bit is set to 0. Do not assign the SCL0 or SDA0 pin to the RIIC when setting up the multi-function pin controller (MPC). Note that the slave address comparison operation is carried out if the pins are assigned to the RIIC.

34.2.2 I2C-bus Control Register 2 (ICCR2)

Address(es): RIIC0.ICCR2 0008 8301h

	b7	b6	b5	b4	b3	b2	b1	b0
	BBSY	MST	TRS	—	SP	RS	ST	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b1	ST	Start Condition Generation Request	0: Does not request to generate a start condition. 1: Requests to generate a start condition.	R/W
b2	RS	Restart Condition Generation Request	0: Does not request to generate a restart condition. 1: Requests to generate a restart condition.	R/W
b3	SP	Stop Condition Generation Request	0: Does not request to generate a stop condition. 1: Requests to generate a stop condition.	R/W
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	TRS	Transmit/Receive Mode	0: Receive mode 1: Transmit mode	R/W*1
b6	MST	Master/Slave Mode	0: Slave mode 1: Master mode	R/W*1
b7	BBSY	Bus Busy Detection Flag	0: The I2C-bus is released (bus free state). 1: The I2C-bus is occupied (bus busy state).	R

Note 1. When the ICMR1.MTWP bit is set to 1, bits MST and TRS can be written to.

ST Bit (Start Condition Generation Request)

This bit is used to request transition to master mode and generation of a start condition.

When this bit is set to 1 to request to generate a start condition, a start condition is generated when the BBSY flag is set to 0 (bus free state).

For details on the start condition generation, refer to section 34.10, Start Condition/Restart Condition/Stop Condition Generating Function.

[Setting condition]

- When 1 is written to the ST bit

[Clearing conditions]

- When 0 is written to the ST bit
- When a start condition has been generated (a start condition is detected)
- When the ICSR2.AL (arbitration-lost) flag is set to 1
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

Note: Set the ST bit to 1 (requests to generate a start condition) when the BBSY flag is set to 0 (bus free state). Note that arbitration may be lost due to a start condition generation error if the ST bit is set to 1 (requests to generate a start condition) when the BBSY flag is set to 1 (bus busy state).

RS Bit (Restart Condition Generation Request)

This bit is used to request that a restart condition be generated in master mode.
When this bit is set to 1 to request to generate a restart condition, a restart condition is generated when the BBSY flag is set to 1 (bus busy state) and the MST bit is set to 1 (master mode).
For details on the restart condition generation, refer to section 34.10, Start Condition/Restart Condition/Stop Condition Generating Function.
[Setting condition]

- When 1 is written to the RS bit with the ICCR2.BBSY flag set to 1

[Clearing conditions]

- When 0 is written to the RS bit
- When a restart condition has been generated (a start condition is detected)
- When the ICSR2.AL (arbitration-lost) flag is set to 1
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

Note: Do not set the RS bit to 1 while generating a stop condition.
Note: If 1 (requests to generate a restart condition) is written to the RS bit in slave mode, the restart condition is not generated but the RS bit remains set to 1. If the operating mode changes to master mode with the bit not being cleared, note that the restart condition may be generated.

SP Bit (Stop Condition Generation Request)

This bit is used to request that a stop condition be generated in master mode.
When this bit is set to 1 to request to generate a stop condition, a stop condition is generated when the BBSY flag is set to 1 (bus busy state) and the MST bit is set to 1 (master mode).
For details on the stop condition generation, refer to section 34.10, Start Condition/Restart Condition/Stop Condition Generating Function.
[Setting condition]

- When 1 is written to the SP bit with both the BBSY flag and the ICCR2.MST bit set to 1

[Clearing conditions]

- When 0 is written to the SP bit
- When a stop condition has been generated (a stop condition is detected)
- When the ICSR2.AL (arbitration-lost) flag is set to 1
- When a start condition and a restart condition are detected
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

Note: Writing to the SP bit is not possible while the setting of the BBSY flag is 0 (bus free state).
Note: Do not set the SP bit to 1 while a restart condition is being generated.

TRS Bit (Transmit/Receive Mode)

This bit indicates transmit or receive mode.
The RIIC is in receive mode when the TRS bit is set to 0 and is in transmit mode when the bit is set to 1. Combination of this bit and the MST bit indicates the operating mode of the RIIC.
The value of the TRS bit is automatically changed to 1 for transmission or 0 for reception in response to the generation or detection of a start condition and setting of the R/W# bit. Although writing to the TRS bit is possible when the ICMR1.MTWP bit is set to 1, writing to this bit is not necessary during normal usage.
[Setting conditions]

- When a start condition is generated normally according to the start condition generation request (when a start condition is detected with the ST bit set to 1)
- When a restart condition is generated normally according to the restart condition generation request (when a restart condition is detected with the RS bit set to 1)
- When the R/W# bit added to the slave address is set to 0 in master mode
- When the address received in slave mode matches the address enabled in the ICSER register, with the R/W# bit set to 1
- When 1 is written to the TRS bit with the ICMR1.MTWP bit set to 1

[Clearing conditions]

- When a stop condition is detected
- The ICSR2.AL (arbitration-lost) flag being set to 1
- In master mode, reception of a slave address to which an R/W# bit with the value 1 is appended
- In slave mode, a match between the received address and the address enabled in the ICSER register when the value of the received R/W# bit is 0 (including cases where the received address is the general call address)
- In slave mode, a restart condition is detected (that is, a start condition is detected while the ICCR2.BBSY flag is 1 and the ICCR2.MST bit is 0)
- When 0 is written to the TRS bit with the ICMR1.MTWP bit set to 1
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

MST Bit (Master/Slave Mode)

This bit indicates master or slave mode.
The RIIC is in slave mode when the MST bit is set to 0 and is in master mode when the bit is set to 1. Combination of this bit and the TRS bit indicates the operating mode of the RIIC.
The value of the MST bit is automatically changed to 1 for master mode or 0 for slave mode by generating of a start condition and generating or detection of a stop condition, etc. Although writing to the MST bit is possible when the ICMR1.MTWP bit is set to 1, writing to this bit is not necessary during normal usage.
[Setting conditions]

- When a start condition is generated normally according to the start condition generation request (when a start condition is detected with the ST bit set to 1)
- When 1 is written to the MST bit with the ICMR1.MTWP bit set to 1

[Clearing conditions]

- When a stop condition is detected
- When the ICSR2.AL (arbitration-lost) flag is set to 1
- When 0 is written to the MST bit with the ICMR1.MTWP bit set to 1
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

BBSY Flag (Bus Busy Detection Flag)

The BBSY flag indicates whether the I2C-bus is occupied (bus busy state) or released (bus free state).
This bit is set to 1 when the SDA0 line changes from high to low under the condition of SCL0 line = high, assuming that a start condition has been generated.
The RIIC recognizes the SDA0 line changing from low to high while the SCL0 line is high as generation of the stop condition. After that, this flag becomes 0 if the RIIC does not detect a start condition during the bus free time (the period set in the ICBRL register).
[Setting condition]

- When a start condition is detected

[Clearing conditions]

- When the bus free time (specified in the ICBRL register) start condition is not detected after detecting a stop condition
- When 1 is written to the ICCR1.IICRST bit with the ICCR1.ICE bit set to 0 (RIIC reset)

34.2.3 I2C-bus Mode Register 1 (ICMR1)

Address(es): RIIC0.ICMR1 0008 8302h

	b7	b6	b5	b4	b3	b2	b1	b0
	MTWP	CKS[2:0]			BCWP	BC[2:0]		
Value after reset:	0	0	0	0	1	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	BC[2:0]	Bit Counter	b2 b0 0 0 0: 9 bits 0 0 1: 2 bits 0 1 0: 3 bits 0 1 1: 4 bits 1 0 0: 5 bits 1 0 1: 6 bits 1 1 0: 7 bits 1 1 1: 8 bits	R/W*1
b3	BCWP	BC Write Protect	0: Enables a value to be written in the BC[2:0] bits. (This bit is read as 1.)	R/W*1
b6 to b4	CKS[2:0]	Internal Reference Clock Select	Select the internal reference clock (IICφ) source for the RIIC. b6 b4 0 0 0: PCLK/1 0 0 1: PCLK/2 0 1 0: PCLK/4 0 1 1: PCLK/8 1 0 0: PCLK/16 1 0 1: PCLK/32 1 1 0: PCLK/64 1 1 1: PCLK/128	R/W
b7	MTWP	MST/TRS Write Protect	0: Disables writing to the ICCR2.MST and TRS bits. 1: Enables writing to the ICCR2.MST and TRS bits.	R/W

Note 1. Rewrite the BC[2:0] bits and set the BCWP bit to 0 at the same time.

BC[2:0] Bits (Bit Counter)

These bits function as a counter that indicates the number of bits remaining to be transferred at the detection of a rising edge on the SCL0 line. Although these bits are writable and readable, it is not necessary to access these bits under normal conditions.

To write to these bits, specify the number of bits to be transferred plus one (data is transferred with an additional acknowledgment bit) between transferred bytes when the SCL0 line is low.

The values of the BC[2:0] bits return to 000b at the end of a data transfer including the acknowledgment bit or when a start condition including a restart condition is detected.

34.2.4 I2C-bus Mode Register 2 (ICMR2)

Address(es): RIIC0.ICMR2 0008 8303h

	b7	b6	b5	b4	b3	b2	b1	b0
	DLCS	SDDL[2:0]			—	TMOH	TMOL	TMOS
Value after reset:	0	0	0	0	0	1	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	TMOS	Timeout Detection Time Select	0: Long mode is selected. 1: Short mode is selected.	R/W
b1	TMOL	Timeout L Count Control	0: Count-up is disabled while the SCL0 line is low. 1: Count-up is enabled while the SCL0 line is low.	R/W
b2	TMOH	Timeout H Count Control	0: Count-up is disabled while the SCL0 line is high. 1: Count-up is enabled while the SCL0 line is high.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6 to b4	SDDL[2:0]	SDA Output Delay Counter	• When ICMR2.DLCS bit is 0 (IICφ) b6 b4 0 0: No output delay 0 1: 1 IICφ cycle 1 0: 2 IICφ cycles 1 1: 3 IICφ cycles 0 0: 4 IICφ cycles 0 1: 5 IICφ cycles 1 0: 6 IICφ cycles 1 1: 7 IICφ cycles • When ICMR2.DLCS bit is 1 (IICφ/2) b6 b4 0 0: No output delay 0 1: 1 or 2 IICφ cycles 1 0: 3 or 4 IICφ cycles 1 1: 5 or 6 IICφ cycles 0 0: 7 or 8 IICφ cycles 0 1: 9 or 10 IICφ cycles 1 0: 11 or 12 IICφ cycles 1 1: 13 or 14 IICφ cycles	R/W
b7	DLCS	SDA Output Delay Clock Source Select	0: The internal reference clock (IICφ) is selected as the clock source of the SDA output delay counter. 1: The internal reference clock divided by 2 (IICφ/2) is selected as the clock source of the SDA output delay counter.*1	R/W

Note 1. The DLCS bit setting of 1 (IICφ/2) only becomes valid when SCL pin is low. When SCL pin is high, the DLCS bit setting of 1 becomes invalid and the clock source becomes the internal reference clock (IICφ).

TMOS Bit (Timeout Detection Time Select)

This bit is used to select long mode or short mode for the timeout detection time when the timeout function is enabled (ICFER.TMOE bit is 1). When this bit is set to 0, long mode is selected. When this bit is set to 1, short mode is selected. In long mode, the timeout detection internal counter functions as a 16-bit counter. In short mode, the counter functions as a 14-bit counter. While the SCL0 line is in the state that enables this counter as specified by bits TMOH and TMOL, the counter counts up in synchronization with the internal reference clock (IICφ) as a count source.

For details on the timeout function, refer to section 34.11.1, Timeout Function.

TMOL Bit (Timeout L Count Control)

This bit is used to enable or disable the internal counter of the timeout function to count up while the SCL0 line is held low when the timeout function is enabled (ICFER.TMOE bit is 1).

TMOH Bit (Timeout H Count Control)

This bit is used to enable or disable the internal counter of the timeout function to count up while the SCL0 line is held high when the timeout function is enabled (ICFER.TMOE bit is 1).

SDDL[2:0] Bits (SDA Output Delay Counter)

The SDA output can be delayed by the SDDL[2:0] setting. This counter works with the clock source selected by the DLCS bit. The setting of this function can be used for all types of SDA output, including the transmission of the acknowledgment bit.

Set the SDA output delay time to meet the I2C-bus specification (within the data valid time/data valid acknowledge time*1) or the SMBus specification (more than the data hold time (300 ns) and less than "clock low period – data setup time (250 ns)"). Note that, if a value outside the specification is set, communication with communication devices may malfunction or it may seemingly become a start condition or stop condition depending on the bus state.

For details on this function, refer to section 34.5, SDA Output Delay Function.

Note 1. Data valid time/data valid acknowledge time
3,450 ns (up to 100 kbps: Standard-mode (Sm))
900 ns (up to 400 kbps: Fast-mode (Fm))

34.2.5 I2C-bus Mode Register 3 (ICMR3)

Address(es): RIIC0.ICMR3 0008 8304h

	b7	b6	b5	b4	b3	b2	b1	b0
	SMBS	WAIT	RDRFS	ACKWP	ACKBT	ACKBR	NF[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	NF[1:0]	Noise Filter Stage Select	b1 b0 0 0: Noise of up to one IICφ cycle is filtered out (single-stage filter). 0 1: Noise of up to two IICφ cycles is filtered out (2-stage filter). 1 0: Noise of up to three IICφ cycles is filtered out (3-stage filter). 1 1: Noise of up to four IICφ cycles is filtered out (4-stage filter).	R/W
b2	ACKBR	Received Acknowledge	0: 0 is received as the acknowledgment bit (ACK reception). 1: 1 is received as the acknowledgment bit (NACK reception).	R
b3	ACKBT	Transmit Acknowledge	0: 0 is to be sent as the acknowledgment bit (ACK transmission). 1: 1 is to be sent as the acknowledgment bit (NACK transmission).	R/W*1
b4	ACKWP	ACKBT Write Protect	0: Modification of the ACKBT bit is disabled. 1: Modification of the ACKBT bit is enabled.	R/W*1
b5	RDRFS	RDRF Flag Set Timing Select	0: The RDRF flag is set at the rising edge of the ninth SCL. (The SCL0 line is not held low at the falling edge of the eighth clock pulse.) 1: The RDRF flag is set at the rising edge of the eighth SCL. (The SCL0 line is held low at the falling edge of the eighth clock pulse.) Low-hold is released by writing a value to the ACKBT bit.	R/W*2
b6	WAIT	WAIT	0: No WAIT (The period between ninth clock pulse and first clock pulse is not held low.) 1: WAIT (The period between ninth clock pulse and first clock pulse is held low.) Low-hold is released by reading the ICDRR register.	R/W*2
b7	SMBS	SMBus/I2C-bus Select	0: The I2C-bus is selected. 1: The SMBus is selected.	R/W

Note 1. Write to the ACKBT bit only while the ACKWP bit is already 1. If it is attempted to write 1 to both the ACKWP and ACKBT bits at the same time, the ACKBT bit will not be set to 1.

Note 2. The WAIT and RDRFS bits are valid only in receive mode (invalid in transmit mode).

NF[1:0] Bits (Noise Filter Stage Select)

These bits are used to select the number of stages in the digital noise filter.

For details on the digital noise filter function, refer to section 34.6, Digital Noise Filters.

Note: Set the noise range to be filtered out by the noise filter within a range less than the SCL0 line high period or low period. If the noise filter width is set to a value of [the shorter one of either SCL high width or SCL low width] – 1.5 × t_{IICcyc} (cycle time of internal reference clock (IICφ)) or a greater value, the serial clock is regarded as noise by the noise filter function of the RIIC, which may prevent the RIIC from operating normally.

ACKBR Bit (Received Acknowledge)

This bit is used to store the value of the acknowledgment bit received from the receiver in transmit mode.
[Setting condition]

- When 1 is received as the acknowledgment bit with the ICCR2.TRS bit set to 1

[Clearing conditions]

- When 0 is received as the acknowledgment bit with the ICCR2.TRS bit set to 1
- When 1 is written to the ICCR1.IICRST bit while the ICCR1.ICE bit is 0 (RIIC reset)

ACKBT Bit (Transmit Acknowledge)

This bit is used to set the bit to be sent at the acknowledgment timing in receive mode.
[Setting condition]

- When 1 is written to this bit with the ACKWP bit set to 1

[Clearing conditions]

- When 0 is written to this bit with the ACKWP bit set to 1
- When stop condition generation is detected (when a stop condition is detected with the ICCR2.SP bit set to 1)
- When 1 is written to the ICCR1.IICRST bit while the ICCR1.ICE bit is 0 (RIIC reset)

ACKWP Bit (ACKBT Write Protect)

This bit is used to control the modification of the ACKBT bit.

RDRFS Bit (RDRF Flag Set Timing Select)

This bit is used to select the RDRF flag set timing in receive mode and also to select whether to hold the SCL0 line low at the falling edge of the eighth SCL.
When the RDRFS bit is 0, the SCL0 line is not held low at the falling edge of the eighth SCL, and the RDRF flag is set to 1 at the rising edge of the ninth SCL.
When the RDRFS bit is 1, the RDRF flag is set to 1 at the rising edge of the eighth SCL and the SCL0 line is held low at the falling edge of the eighth SCL. The low-hold of the SCL0 line is released by writing a value to the ACKBT bit.
After data is received with this setting, the SCL0 line is automatically held low before the acknowledgment bit is sent. This enables processing to send ACK (ACKBT bit is 0) or NACK (ACKBT bit is 1) according to receive data.

WAIT Bit (WAIT)

This bit is used to control whether to hold the period between the ninth SCL and the first SCL low until the I2C-bus receive data register (ICDRR) is completely read each time single-byte data is received in receive mode.
When the WAIT bit is 0, the receive operation is continued without holding the period between the ninth and the first SCL low. When both the RDRFS and WAIT bits are 0, continuous receive operation is enabled with the double buffer.
When the WAIT bit is 1, the SCL0 line is held low from the falling edge of the ninth SCL until the ICDRR register value is read each time single-byte data is received. This enables receive operation in byte units.

Note: When the value of the WAIT bit is to be read, be sure to read the ICDRR register beforehand.

SMBS Bit (SMBus/I2C-bus Select)

Setting this bit to 1 selects the SMBus and enables the ICSER.HOAE bit.

34.2.6 I2C-bus Function Enable Register (ICFER)

Address(es): RIIC0.ICFER 0008 8305h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	SCLE	NFE	NACKE	SALE	NALE	MALE	TMOE
Value after reset:	0	1	1	1	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	TMOE	Timeout Function Enable	0: The timeout function is disabled. 1: The timeout function is enabled.	R/W
b1	MALE	Master Arbitration-Lost Detection Enable	0: Master arbitration-lost detection is disabled. (Disables the arbitration-lost detection function and does not clear the ICCR2.MST and TRS bits automatically when arbitration is lost.) 1: Master arbitration-lost detection is enabled. (Enables the arbitration-lost detection function and clears the ICCR2.MST and TRS bits automatically when arbitration is lost.)	R/W
b2	NALE	NACK Transmission Arbitration-Lost Detection Enable	0: NACK transmission arbitration-lost detection is disabled. 1: NACK transmission arbitration-lost detection is enabled.	R/W
b3	SALE	Slave Arbitration-Lost Detection Enable	0: Slave arbitration-lost detection is disabled. 1: Slave arbitration-lost detection is enabled.	R/W
b4	NACKE	NACK Reception Transfer Suspension Enable	0: Transfer operation is not suspended during NACK reception (transfer suspension disabled). 1: Transfer operation is suspended during NACK reception (transfer suspension enabled).	R/W
b5	NFE	Digital Noise Filter Enable	0: Digital noise filters are not used. 1: Digital noise filters are used.	R/W
b6	SCLE	SCL Synchronization Enable	0: SCL synchronization is disabled. 1: SCL synchronization is enabled.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

TMOE Bit (Timeout Function Enable)

This bit is used to enable or disable the timeout function.

For details on the timeout function, refer to section 34.11.1, Timeout Function.

MALE Bit (Master Arbitration-Lost Detection Enable)

This bit is used to specify whether to use the arbitration-lost detection function in master mode. Normally, set this bit to 1.

NALE Bit (NACK Transmission Arbitration-Lost Detection Enable)

This bit is used to specify whether to cause arbitration to be lost when ACK is detected during transmission of NACK in receive mode (such as when slaves with the same address exist on the bus or when two or more masters select the same slave device simultaneously with different number of receive bytes).

SALE Bit (Slave Arbitration-Lost Detection Enable)

This bit is used to specify whether to cause arbitration to be lost when a value different from the value being transmitted is detected on the bus in slave transmit mode (such as when slaves with the same address exist on the bus or when a mismatch with the transmit data occurs due to noise).

NACKE Bit (NACK Reception Transfer Suspension Enable)

This bit is used to specify whether to continue or discontinue the data transfer when NACK is received in transmit mode. Normally, set this bit to 1.

When NACK is received with the NACKE bit set to 1, the next transfer operation is suspended.

When the NACKE bit is 0, the next transfer operation is continued regardless of the value of the received acknowledgment bit.

For details on the NACK reception transfer suspension function, refer to section 34.8.2, NACK Reception Transfer Suspension Function.

SCLE Bit (SCL Synchronization Enable)

This bit is used to specify whether to the SCL output is to be synchronized with the SCL input. Normally, set this bit to 1.

When the SCLE bit is set to 0 (SCL synchronization is disabled), the RIIC does not synchronize the SCL output with the SCL input. In this setting, the RIIC outputs the clock with the transfer rate set in registers ICBRH and ICBRL regardless of the SCL0 line state. For this reason, if the load of the I2C-bus line is much larger than the specification value or if the SCL output overlaps in multiple masters, the short-cycle SCL that does not meet the specification may be output. When the SCL synchronization is not used, it also affects the generation of a start condition, restart condition, and stop condition, and the continuous output of additional SCL.

This bit must not be set to 0 except for checking the output of the set transfer rate.

34.2.7 I2C-bus Status Enable Register (ICSER)

Address(es): RIIC0.ICSER 0008 8306h

	b7	b6	b5	b4	b3	b2	b1	b0
	HOAE	—	DIDE	—	GCAE	SAR2E	SAR1E	SAR0E
Value after reset:	0	0	0	0	1	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	SAR0E	Slave Address Register 0 Enable	0: Slave address in registers SARL0 and SARU0 is disabled. 1: Slave address in registers SARL0 and SARU0 is enabled.	R/W
b1	SAR1E	Slave Address Register 1 Enable	0: Slave address in registers SARL1 and SARU1 is disabled. 1: Slave address in registers SARL1 and SARU1 is enabled.	R/W
b2	SAR2E	Slave Address Register 2 Enable	0: Slave address in registers SARL2 and SARU2 is disabled. 1: Slave address in registers SARL2 and SARU2 is enabled.	R/W
b3	GCAE	General Call Address Enable	0: General call address detection is disabled. 1: General call address detection is enabled.	R/W
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	DIDE	Device-ID Address Detection Enable	0: Device-ID address detection is disabled. 1: Device-ID address detection is enabled.	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	HOAE	Host Address Enable	0: Host address detection is disabled. 1: Host address detection is enabled.	R/W

SARyE Bit (Slave Address Register y Enable) (y = 0 to 2)

This bit is used to enable or disable the slave address set in registers SARLy and SARUy.

When this bit is set to 1, the slave address set in registers SARLy and SARUy is enabled and is compared with the received slave address.

When this bit is set to 0, the slave address set in registers SARLy and SARUy is disabled and is ignored even if it matches the received slave address.

GCAE Bit (General Call Address Enable)

This bit is used to specify whether to ignore the general call address (0000 000b + 0 (write): All 0) when it is received.

When this bit is set to 1, if the received slave address matches the general call address, the RIIC recognizes the received slave address as the general call address independently of the slave addresses set in registers SARLy and SARUy (y = 0 to 2) and performs data receive operation.

When this bit is set to 0, the received slave address is ignored even if it matches the general call address.

DIDE Bit (Device-ID Address Detection Enable)

This bit is used to specify whether to recognize and execute the device-ID address when a device ID (1111 100b) is received in the first byte after a start condition or restart condition is detected.

When this bit is set to 1, if the received first byte matches the device ID, the RIIC recognizes that the device-ID address has been received. When the following R/W# bit is 0 (write), the RIIC recognizes the second and the following bytes as slave addresses and continues the receive operation.

When this bit is set to 0, the RIIC ignores the received first byte even if it matches the device ID address and recognizes the first byte as a normal slave address.

For details on the device-ID address detection, refer to section 34.7.3, Device-ID Address Detection.

HOAE Bit (Host Address Enable)

This bit is used to specify whether to ignore received host address (0001 000b) when the ICMR3.SMBS bit is 1.
When this bit is set to 1 while the ICMR3.SMBS bit is 1, if the received slave address matches the host address, the RIIC recognizes the received slave address as the host address independently of the slave addresses set in registers SARLy and SARUy (y = 0 to 2) and performs the receive operation.
When the ICMR3.SMBS bit or the HOAE bit is set to 0, the received slave address is ignored even if it matches the host address.

34.2.8 I2C-bus Interrupt Enable Register (ICIER)

Address(es): RIIC0.ICIER 0008 8307h

	b7	b6	b5	b4	b3	b2	b1	b0
	TIE	TEIE	RIE	NAKIE	SPIE	STIE	ALIE	TMOIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TMOIE	Timeout Interrupt Request Enable	0: Timeout interrupt (TMOI) request is disabled. 1: Timeout interrupt (TMOI) request is enabled.	R/W
b1	ALIE	Arbitration-Lost Interrupt Request Enable	0: Arbitration-lost interrupt (ALI) request is disabled. 1: Arbitration-lost interrupt (ALI) request is enabled.	R/W
b2	STIE	Start Condition Detection Interrupt Request Enable	0: Start condition detection interrupt (STI) request is disabled. 1: Start condition detection interrupt (STI) request is enabled.	R/W
b3	SPIE	Stop Condition Detection Interrupt Request Enable	0: Stop condition detection interrupt (SPI) request is disabled. 1: Stop condition detection interrupt (SPI) request is enabled.	R/W
b4	NAKIE	NACK Reception Interrupt Request Enable	0: NACK reception interrupt (NAKI) request is disabled. 1: NACK reception interrupt (NAKI) request is enabled.	R/W
b5	RIE	Receive Data Full Interrupt Request Enable	0: Receive data full interrupt (RXI) request is disabled. 1: Receive data full interrupt (RXI) request is enabled.	R/W
b6	TEIE	Transmission End Interrupt Request Enable	0: Transmission end interrupt (TEI) request is disabled. 1: Transmission end interrupt (TEI) request is enabled.	R/W
b7	TIE	Transmit Data Empty Interrupt Request Enable	0: Transmit data empty interrupt (TXI) request is disabled. 1: Transmit data empty interrupt (TXI) request is enabled.	R/W

TMOIE Bit (Timeout Interrupt Request Enable)

This bit is used to enable or disable timeout interrupt (TMOI) requests when the ICSR2.TMOF flag is set to 1. A TMOI interrupt request is canceled by setting the TMOF flag or the TMOIE bit to 0.

ALIE Bit (Arbitration-Lost Interrupt Request Enable)

This bit is used to enable or disable arbitration-lost interrupt (ALI) requests when the ICSR2.AL flag is set to 1. An ALI interrupt request is canceled by setting the AL flag or the ALIE bit to 0.

STIE Bit (Start Condition Detection Interrupt Request Enable)

This bit is used to enable or disable start condition detection interrupt (STI) requests when the ICSR2.START flag is set to 1. An STI interrupt request is canceled by setting the START flag or the STIE bit to 0.

SPIE Bit (Stop Condition Detection Interrupt Request Enable)

This bit is used to enable or disable stop condition detection interrupt (SPI) requests when the ICSR2.STOP flag is set to 1. An SPI interrupt request is canceled by setting the STOP flag or the SPIE bit to 0.

NAKIE Bit (NACK Reception Interrupt Request Enable)

This bit is used to enable or disable NACK reception interrupt (NAKI) requests when the ICSR2.NACKF flag is set to 1. An NAKI interrupt request is canceled by setting the NACKF flag or the NAKIE bit to 0.

RIE Bit (Receive Data Full Interrupt Request Enable)

This bit is used to enable or disable receive data full interrupt (RXI) requests when the ICSR2.RDRF flag is set to 1.

TEIE Bit (Transmission End Interrupt Request Enable)

This bit is used to enable or disable transmission end interrupt (TEI) requests when the ICSR2.TEND flag is set to 1. An TEI interrupt request is canceled by setting the TEND flag or the TEIE bit to 0.

TIE Bit (Transmit Data Empty Interrupt Request Enable)

This bit is used to enable or disable transmit data empty interrupt (TXI) requests when the ICSR2.TDRE flag is set to 1.

34.2.9 I2C-bus Status Register 1 (ICSR1)

Address(es): RIIC0.ICSR1 0008 8308h

	b7	b6	b5	b4	b3	b2	b1	b0
	HOA	—	DID	—	GCA	AAS2	AAS1	AAS0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	AAS0	Slave Address 0 Detection Flag	0: Slave address 0 is not detected. 1: Slave address 0 is detected.	R/(W) *1
b1	AAS1	Slave Address 1 Detection Flag	0: Slave address 1 is not detected. 1: Slave address 1 is detected.	R/(W) *1
b2	AAS2	Slave Address 2 Detection Flag	0: Slave address 2 is not detected. 1: Slave address 2 is detected.	R/(W) *1
b3	GCA	General Call Address Detection Flag	0: General call address is not detected. 1: General call address is detected.	R/(W) *1
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	DID	Device-ID Address Detection Flag	0: Device-ID command is not detected. 1: Device-ID command is detected. • This bit is set to 1 when the first byte received immediately after a start condition is detected matches a value of (device ID (1111 100b) + 0 (write)).	R/(W) *1
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	HOA	Host Address Detection Flag	0: Host address is not detected. 1: Host address is detected. • This bit is set to 1 when the received slave address matches the host address (0001 000b).	R/(W) *1

Note 1. Only 0 can be written to clear the flag.

AASy Flag (Slave Address y Detection Flag) (y = 0 to 2)

[Setting conditions]

For 7-bit address format: SARUy.FS bit = 0

- When the received slave address matches the SARLy.SVA[6:0] bits value with the ICSER.SARyE bit set to 1 (slave address y detection enabled)
  This flag is set to 1 at the rising edge of the ninth SCL in the first byte.

For 10-bit address format: SARUy.FS bit = 1

- When the received slave address matches a value of (11110b + SARUy.SVA[1:0] bits) and the following address matches the SARLy value with the ICSER.SARyE bit set to 1 (slave address y detection enabled)
  This flag is set to 1 at the rising edge of the ninth SCL in the second byte.

[Clearing conditions]

- When 0 is written to the AASy flag after reading the AASy flag to be 1
- When a stop condition is detected
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

For 7-bit address format: SARUy.FS bit = 0

- When the received slave address does not match the SARLy.SVA[6:0] bits value with the ICSER.SARyE bit set to 1 (slave address y detection enabled)
  This flag is set to 0 at the rising edge of the ninth SCL in the first byte.

For 10-bit address format: SARUy.FS bit = 1

- When the received slave address does not match a value of (11110b + SARUy.SVA[1:0] bits) with the ICSER.SARyE bit set to 1 (slave address y detection enabled)
  This flag is set to 0 at the rising edge of the ninth SCL in the first byte.
- When the received slave address matches a value of (11110b + SARUy.SVA[1:0] bits) and the following address does not match the SARLy value with the ICSER.SARyE bit set to 1 (slave address y detection enabled)
  This flag is set to 0 at the rising edge of the ninth SCL in the second byte.

GCA Flag (General Call Address Detection Flag)

[Setting condition]

- When the received slave address matches the general call address (0000 000b + 0 (write)) with the ICSER.GCAE bit set to 1 (general call address detection is enabled)
  This flag is set to 1 at the rising edge of the ninth SCL in the first byte.

[Clearing conditions]

- When 0 is written to the GCA flag after reading GCA flag to be 1
- When a stop condition is detected
- When the received slave address does not match the general call address (0000 000b + 0 (write)) with the ICSER.GCAE bit set to 1 (general call address detection is enabled)
  This flag is set to 0 at the rising edge of the ninth SCL in the first byte.
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

DID Flag (Device-ID Address Detection Flag)

[Setting condition]

- When the first byte received immediately after a start condition or restart condition is detected matches a value of (device ID (1111 100b) + 0 (write)) with the ICSER.DIDE bit set to 1 (device-ID address detection is enabled)
  This flag is set to 1 at the rising edge of the ninth SCL in the first byte.

[Clearing conditions]

- When 0 is written to the DID flag after reading DID flag to be 1
- When a stop condition is detected
- When the first byte received immediately after a start condition or restart condition is detected does not match a value of (device ID (1111 100b)) with the ICSER.DIDE bit set to 1 (device-ID address detection is enabled)
  This flag is set to 0 at the rising edge of the ninth SCL in the first byte.
- When the first byte received immediately after a start condition or restart condition is detected matches a value of (device ID (1111 100b) + 0 (write)) and the second byte does not match any of slave addresses 0 to 2 with the ICSER.DIDE bit set to 1 (device-ID address detection is enabled)
  This flag is set to 0 at the rising edge of the ninth SCL in the second byte.
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

HOA Flag (Host Address Detection Flag)

[Setting condition]

- When the received slave address matches the host address (0001 000b) with the ICSER.HOAE bit set to 1 (host address detection is enabled)
  This flag is set to 1 at the rising edge of the ninth SCL in the first byte.

[Clearing conditions]

- When 0 is written to the HOA flag after reading HOA flag to be 1
- When a stop condition is detected
- When the received slave address does not match the host address (0001 000b) with the ICSER.HOAE bit set to 1

(host address detection is enabled)
This flag is set to 0 at the rising edge of the ninth SCL in the first byte.

- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

34.2.10 I2C-bus Status Register 2 (ICSR2)

Address(es): RIIC0.ICSR2 0008 8309h

	b7	b6	b5	b4	b3	b2	b1	b0
	TDRE	TEND	RDRF	NACKF	STOP	START	AL	TMOF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TMOF	Timeout Detection Flag	0: Timeout is not detected. 1: Timeout is detected.	R/(W) *1
b1	AL	Arbitration-Lost Flag	0: Arbitration is not lost. 1: Arbitration is lost.	R/(W) *1
b2	START	Start Condition Detection Flag	0: Start condition is not detected. 1: Start condition is detected.	R/(W) *1
b3	STOP	Stop Condition Detection Flag	0: Stop condition is not detected. 1: Stop condition is detected.	R/(W) *1
b4	NACKF	NACK Detection Flag	0: NACK is not detected. 1: NACK is detected.	R/(W) *1
b5	RDRF	Receive Data Full Flag	0: The ICDRR register contains no receive data. 1: The ICDRR register contains receive data.	R/(W) *1
b6	TEND	Transmission End Flag	0: Data is being transmitted. 1: Data has been transmitted.	R/(W) *1
b7	TDRE	Transmit Data Empty Flag	0: The ICDRT register contains transmit data. 1: The ICDRT register contains no transmit data.	R

Note 1. Only 0 can be written to clear the flag.

TMOF Flag (Timeout Detection Flag)

This flag is set to 1 when the RIIC recognizes timeout after the SCL0 line state remains unchanged for a certain period.
[Setting condition]

- When the SCL0 line state remains unchanged for the period specified by bits ICMR2.TMOH, TMOL, and TMOS while the ICFER.TMOE bit is 1 (the timeout function is enabled) in master mode or in slave mode and the received slave address matches.

[Clearing conditions]

- When 0 is written to the TMOF bit after reading TMOF = 1
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

AL Flag (Arbitration-Lost Flag)

This flag shows that bus mastership has been lost (loss in arbitration) due to a bus conflict or some other reason when a start condition is generated or an address and data are transmitted. The RIIC monitors the level on the SDA0 line during transmission and, if the level on the line does not match the value of the bit being output, sets the value of the AL flag to 1 to indicate that the bus is occupied by another device.
The RIIC can also detect loss of arbitration during NACK transmission in receive mode or during data transmission in slave mode.

[Setting conditions]

When master arbitration-lost detection is enabled: ICFER.MALE = 1

- When the internal SDA output state does not match the SDA0 line level at the rising edge of SCL except for the ACK period during data (including slave address) transmission in master transmit mode (when the SDA0 line is driven low while the internal SDA output is high (the SDA0 pin is in the high-impedance state))
- When a start condition is detected while the ICCR2.ST bit is 1 (requests to generate a start condition) or the internal SDA output state does not match the SDA0 line level
- When the ICCR2.ST bit is set to 1 (requests to generate a start condition) with the ICCR2.BBSY flag set to 1.

When NACK arbitration-lost detection is enabled: ICFER.NALE = 1

- When the internal SDA output state does not match the SDA0 line level at the rising edge of SCL in the ACK period during NACK transmission in receive mode

When slave arbitration-lost detection is enabled: ICFER.SALE = 1

- When the internal SDA output state does not match the SDA0 line level at the rising edge of SCL except for the ACK period during data transmission in slave transmit mode

[Clearing conditions]

- When 0 is written to the AL flag after reading AL = 1
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

Table 34.4 Relationship between Arbitration-Lost Generation Sources and Arbitration-Lost Enable Functions

ICFER			ICSR2		
MALE	NALE	SALE	AL	Error	Arbitration-Lost Generation Source
1	×	×	1	Start condition generation error	When internal SDA output state does not match SDA0 line level when a start condition is detected while the ICCR2.ST bit is 1
					When ICCR2.ST bit is set to 1 with ICCR2.BBSY flag set to 1
			1	Transmit data mismatch	When transmit data (including slave address) does not match the bus state in master transmit mode
×	1	×	1	NACK transmission mismatch	When ACK is detected during transmission of NACK in master receive mode or slave receive mode
×	×	1	1	Transmit data mismatch	When transmit data does not match the bus state in slave transmit mode

×: Don't care

START Flag (Start Condition Detection Flag)

[Setting condition]

- When a start condition (or a restart condition) is detected

[Clearing conditions]

- When 0 is written to the START bit after reading START = 1
- When a stop condition is detected
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

STOP Flag (Stop Condition Detection Flag)

[Setting condition]

- When a stop condition is detected

[Clearing conditions]

- When 0 is written to the STOP bit after reading STOP = 1
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

NACKF Flag (NACK Detection Flag)

[Setting condition]

- When ACK is not received (NACK is received) from the receiver in transmit mode with the ICFER.NACKE bit set to 1 (transfer suspension enabled)

[Clearing conditions]

- When 0 is written to the NACKF bit after reading NACKF = 1
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

Note: When the NACKF flag is set to 1, the RIIC suspends data transmission/reception. Writing to the ICDRT register in transmit mode or reading from the ICDRR register in receive mode with the NACKF flag set to 1 does not enable data transmit/receive operation. To restart data transmission/reception, set the NACKF flag to 0.

RDRF Flag (Receive Data Full Flag)

[Setting conditions]

- When receive data has been transferred from the ICDRS register to the ICDRR register
  This flag is set to 1 at the rising edge of the eighth or ninth SCL (selected by the ICMR3.RDRFS bit)
- When the received slave address matches after a start condition (or a restart condition) is detected with the ICCR2.TRS bit set to 0

[Clearing conditions]

- When 0 is written to the RDRF bit after reading RDRF = 1
- When data is read from the ICDRR register
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

TEND Flag (Transmission End Flag)

[Setting condition]

- At the rising edge of the ninth SCL while the TDRE flag is 1

[Clearing conditions]

- When 0 is written to the TEND bit after reading TEND = 1
- When data is written to the ICDRT register
- When a stop condition is detected
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

TDRE Flag (Transmit Data Empty Flag)

[Setting conditions]

- When data has been transferred from the ICDRT register to the ICDRS register and the ICDRT register becomes empty
- When the ICCR2.TRS bit is set to 1
- When the received slave address matches while the TRS bit is 1

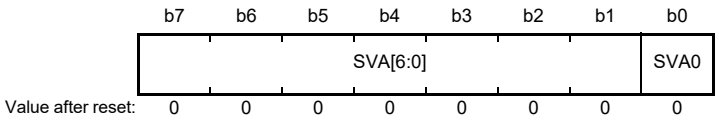
[Clearing conditions]

- When data is written to the ICDRT register
- When the ICCR2.TRS bit is set to 0
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

Note: The NACKF flag becoming 1 while the ICFER.NACKE bit is 1 suspends data transmission and reception by the RIIC. Even if the next data for transmission has already been written to the ICDRT register (the TDRE flag is 0), the data in the ICDRT register is retained but not transferred to the ICDRS register. At this point, the TDRE flag does not become 1.

34.2.11 Slave Address Register Ly (SARLy) (y = 0 to 2)

Address(es): RIIC0.SARL0 0008 830Ah, RIIC0.SARL1 0008 830Ch, RIIC0.SARL2 0008 830Eh



Bit	Symbol	Bit Name	Description	R/W
b0	SVA0	10-Bit Address LSB	Set a slave address	R/W
b7 to b1	SVA[6:0]	7-Bit Address/10-Bit Address Lower Bits	Set a slave address	R/W

SVA0 Bit (10-Bit Address LSB)

When the 10-bit address format is selected (SARUy.FS bit is 1), this bit functions as the LSB of a 10-bit address and forms the lower 8 bits of a 10-bit address in combination with the SVA[6:0] bits.
When the ICSER.SARyE bit is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 1, this bit is valid. While the SARUy.FS bit or SARyE bit is 0, the setting of this bit is ignored.

SVA[6:0] Bits (7-Bit Address/10-Bit Address Lower Bits)

When the 7-bit address format is selected (SARUy.FS bit is 0), these bits function as a 7-bit address. When the 10-bit address format is selected (SARUy.FS bit is 1), these bits function as the lower 8 bits of a 10-bit address in combination with the SVA0 bit.
While the ICSER.SARyE bit is 0, the setting of these bits is ignored.

34.2.12 Slave Address Register Uy (SARUy) (y = 0 to 2)

Address(es): RIIC0.SARU0 0008 830Bh, RIIC0.SARU1 0008 830Dh, RIIC0.SARU2 0008 830Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SVA[1:0]		FS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	FS	7-Bit/10-Bit Address Format Select	0: The 7-bit address format is selected. 1: The 10-bit address format is selected.	R/W
b2, b1	SVA[1:0]	10-Bit Address Upper Bits	Set a slave address	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

FS Bit (7-Bit/10-Bit Address Format Select)

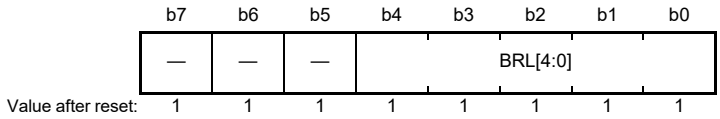
This bit is used to select 7-bit address or 10-bit address for slave address y (in registers SARLy and SARUy).
When the ICSER.SARyE bit is set to 1 (registers SARLy and SARUy enabled) and the SARUy.FS bit is 0, the 7-bit address format is selected for slave address y, the SARLy.SVA[6:0] bits setting is valid, and the settings of the SVA[1:0] bits and the SARLy.SVA0 bit are ignored.
When the ICSER.SARyE bit is set to 1 (registers SARLy and SARUy enabled) and the SARUy.FS bit is 1, the 10-bit address format is selected for slave address y and the settings of the SVA[1:0] bits and SARLy are valid.
While the ICSER.SARyE bit is 0 (registers SARLy and SARUy disabled), the setting of the SARUy.FS bit is invalid.

SVA[1:0] Bits (10-Bit Address Upper Bits)

When the 10-bit address format is selected (FS = 1), these bits function as the upper 2 bits of a 10-bit address.
When the ICSER.SARyE bit is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 1, these bits are valid.
While the SARUy.FS bit or SARyE bit is 0, the setting of these bits is ignored.

34.2.13 I2C-bus Bit Rate Low-Level Register (ICBRL)

Address(es): RIIC0.ICBRL 0008 8310h



Bit	Symbol	Bit Name	Description	R/W
b4 to b0	BRL[4:0]	Bit Rate Low Period	Low period of SCL	R/W
b7 to b5	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

ICBRL is a 5-bit register to set the low period of SCL.
It also works to generate the data setup time for automatic SCL low-hold operation (refer to section 34.8, Automatic Low-Hold Function for SCL); when the RIIC is used only in slave mode, this register needs to be set to a value longer than the data setup time*1.
ICBRL counts the low period with the internal reference clock (IICφ) specified by the ICMR1.CKS[2:0] bits.
If the digital noise filter is enabled (the ICFER.NFE bit is 1), set the ICBRL register to a value at least one greater than the number of stages in the noise filter. Regarding the number of stages in the noise filter, see the description of the ICMR3.NF[1:0] bits.

Note 1. Data setup time (tSU: DAT)
250 ns (up to 100 kbps: Standard-mode (Sm))
100 ns (up to 400 kbps: Fast-mode (Fm))

34.2.14 I2C-bus Bit Rate High-Level Register (ICBRH)

Address(es): RIIC0.ICBRH 0008 8311h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	BRH[4:0]				
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	BRH[4:0]	Bit Rate High Period	High period of SCL	R/W
b7 to b5	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

ICBRH is a 5-bit register to set the high period of SCL. ICBRH is valid in master mode. If the RIIC is used only in slave mode, this register need not to set the high period.
ICBRH counts the high period with the internal reference clock (IICφ) specified by the ICMR1.CKS[2:0] bits.
If the digital noise filter is enabled (the ICFER.NFE bit is 1), set the ICBRH register to a value at least one greater than the number of stages in the noise filter. Regarding the number of stages in the noise filter, see the description of the ICMR3.NF[1:0] bits.

The I2C transfer rate and the SCL duty are calculated using the following expression.
Transfer rate = 1 / {[(ICBRH + 1) + (ICBRL + 1)] / IICφ*1 + SCL0 line rise time [tr] + SCL0 line fall time [tf]}
Duty cycle = {SCL0 line rise time [tr]*2 + (ICBRH + 1) / IICφ} / {SCL0 line fall time [tf]*2 + (ICBRL + 1) / IICφ}

Note 1. IICφ = PCLK × Division ratio
Note 2. The SCL0 line rise time [tr] and SCL0 line fall time [tf] depend on the total bus line capacitance [Cb] and the pull-up resistor [Rp]. For details, refer to the I2C-bus specification from NXP Semiconductors.

Table 34.5 lists examples of ICBRH/ICBRL settings.

Table 34.5 Examples of ICBRH/ICBRL Settings for Transfer Rate

Transfer Rate (kbps)	Operating Frequency PCLK (MHz)								
	8			10			12.5		
	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL
10	100b	22 (F6h)	25 (F9h)	101b	13 (EDh)	15 (EFh)	101b	16 (F0h)	20 (F4h)
50	010b	16 (F0h)	19 (F3h)	010b	21 (F5h)	24 (F8h)	011b	12 (ECh)	15 (EFh)
100	001b	15 (EFh)	18 (F2h)	001b	19 (F3h)	23 (F7h)	001b	24 (F8h)	29 (FDh)
400	000b	4 (E4h)	10 (EAh)	000b	5 (E5h)	12 (ECh)	000b	7 (E7h)	16 (F0h)

Transfer Rate (kbps)	Operating Frequency PCLK (MHz)								
	16			20			25		
	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL
10	101b	22 (F6h)	25 (F9h)	110b	13 (EDh)	15 (EFh)	110b	16 (F0h)	20 (F4h)
50	011b	16 (F0h)	19 (F3h)	011b	21 (F5h)	24 (F8h)	100b	12 (ECh)	15 (EFh)
100	010b	15 (EFh)	18 (F2h)	010b	19 (F3h)	23 (F7h)	010b	24 (F8h)	29 (FDh)
400	000b	9 (E9h)	20 (F4h)	000b	11 (EBh)	25 (F9h)	001b	7 (E7h)	16 (F0h)

Transfer Rate (kbps)	Operating Frequency PCLK (MHz)								
	30			32			33		
	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL
10	110b	20 (F4h)	24 (F8h)	110b	22 (F6h)	25 (F9h)	110b	22 (F6h)	26 (FAh)
50	100b	15 (EFh)	18 (F2h)	100b	16 (F0h)	19 (F3h)	100b	17 (F1h)	20 (F4h)
100	011b	14 (EEh)	17 (F1h)	011b	15 (EFh)	18 (F2h)	011b	16 (F0h)	19 (F3h)
400	001b	8 (E8h)	19 (F3h)	001b	9 (E9h)	20 (F4h)	001b	9 (E9h)	21 (F5h)

Transfer Rate (kbps)	Operating Frequency PCLK (MHz)								
	40			50			60		
	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL
10	111b	13 (EDh)	15 (EFh)	111b	16 (F0h)	20 (F4h)	111b	20 (F4h)	24 (F8h)
50	100b	21 (F5h)	24 (F8h)	100b	26 (FAh)	31 (FFh)	101b	15 (EFh)	18 (F2h)
100	011b	19 (F3h)	23 (F7h)	011b	24 (F8h)	29 (FDh)	100b	14 (EEh)	17 (F1h)
400	001b	11 (EBh)	25 (F9h)	010b	7 (E7h)	16 (F0h)	010b	8 (E8h)	19 (F3h)

Note: ICBRH/ICBRL settings in these tables are calculated using the following values:
SCL0 line rise time (tr): 100 kbps or less (Sm): 1000 ns, 400 kbps or less (Fm): 300 ns
SCL0 line fall time (tf): 400 kbps or less (Sm/Fm): 300 ns
For the specified values of rise time (tr) and fall time (tf) of the SCL0 signal, refer to the I2C-bus specification from NXP Semiconductors.

34.2.15 I2C-bus Transmit Data Register (ICDRT)

Address(es): RIIC0.ICDRT 0008 8312h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1

When the ICDRT register detects a space in the I2C-bus shift register (ICDRS), it transfers the transmit data that has been written to the ICDRT register to the ICDRS register and starts transmitting data in transmit mode.
The double-buffer structure of the ICDRT register and the ICDRS register allows continuous transmit operation if the next transmit data has been written to the ICDRT register while the ICDRS register data is being transmitted.
The ICDRT register can always be read and written. Write transmit data to the ICDRT register once when a transmit data empty interrupt (TXI) request is generated.

34.2.16 I2C-bus Receive Data Register (ICDRR)

Address(es): RIIC0.ICDRR 0008 8313h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

When 1 byte of data has been received, the received data is transferred from the I2C-bus shift register (ICDRS) to the ICDRR register to enable the next data to be received.
The double-buffer structure of the ICDRS register and the ICDRR register allows continuous receive operation if the received data has been read from the ICDRR register while the ICDRS register is receiving data.
The ICDRR register cannot be written. Read data from the ICDRR register once when a receive data full interrupt (RXI) request is generated.
If the ICDRR register receives the next receive data before the current data is read from the ICDRR register (while the ICSR2.RDRF flag is 1), the RIIC automatically holds the SCL line low one cycle before the RDRF flag is set to 1 next.

34.2.17 I2C-bus Shift Register (ICDRS)

The ICDRS register is an 8-bit shift register to transmit and receive data.
During transmission, transmit data is transferred from the ICDRT register to the ICDRS register and is sent from the SDA0 pin. During reception, data is transferred from the ICDRS register to the ICDRR register after 1 byte of data has been received.
The ICDRS register cannot be accessed directly.

34.3 Operation

34.3.1 Communication Data Format

The I2C-bus format consists of 8-bit data and 1-bit acknowledgment. The first byte following a start condition or restart condition is an address byte used to specify a slave device with which the master device communicates. The specified slave is valid until a new slave is specified or a stop condition is generated.
Figure 34.2 shows the I2C-bus format, and Figure 34.3 shows the I2C-bus timing.

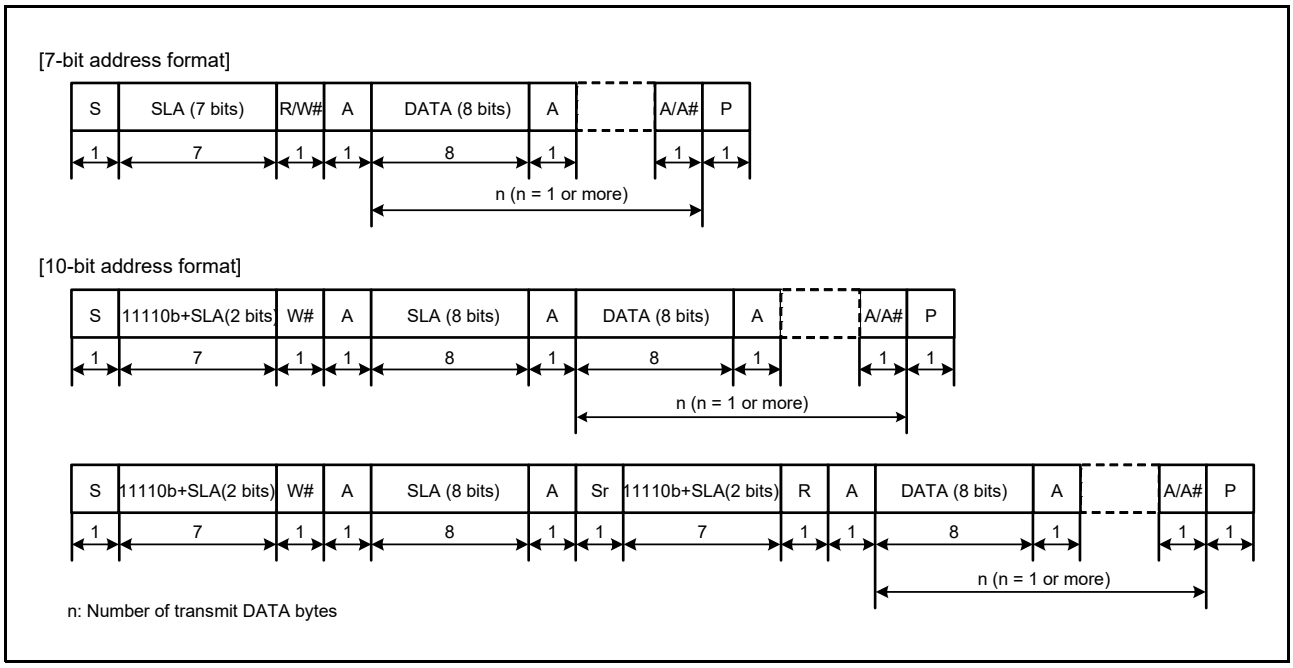


Figure 34.2 I2C-bus Format

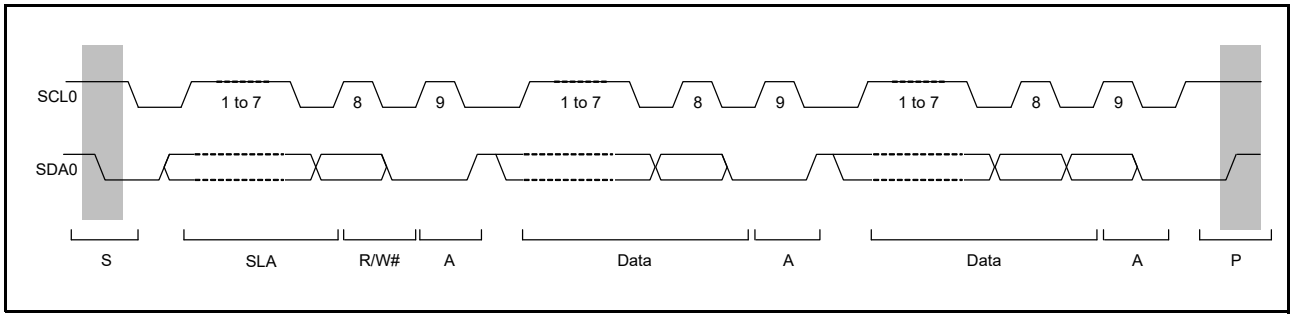


Figure 34.3 I2C-bus Timing (SLA = 7 Bits)

S: Start condition. The master device drives the SDA0 line low from high while the SCL0 line is high.

SLA: Slave address, by which the master device selects a slave device.

R/W#: Indicates the direction of data transfer: from the slave device to the master device when R/W is 1, or from the master device to the slave device when R/W is 0.

A: Acknowledge. The receiver drives the SDA0 line low. (In master transmit mode, the slave device returns acknowledge. In master receive mode, the master device returns acknowledge.)

A#: Not Acknowledge. The receiver drives the SDA0 line high.

Sr: Restart condition. The master device drives the SDA0 line low from high after the setup time has elapsed with the SCL0 line high.

DATA: Transmitted or received data

P: Stop condition. The master device drives the SDA0 line high from low while the SCL0 line is high.

34.3.2 Initial Settings

Before starting data transmission and reception, initialize the RIIC according to the procedure in Figure 34.4.
Set the ICCR1.ICE bit to 1 (internal reset) after setting the ICCR1.IICRST bit to 1 (RIIC reset) with the ICCR1.ICE bit set to 0 (SCL0 and SDA0 pins in inactive state). This initializes the various flags and internal state of the ICSR1 register. After that, set registers SARLy, SARUy, ICSER, ICMR1, ICBRH, and ICBRL (y = 0 to 2), and set the other registers as necessary (for initial settings of the RIIC, see Figure 34.4). When the necessary register settings have been completed, set the ICCR1.IICRST bit to 0 (releases the RIIC reset). This step is not necessary if initialization of the RIIC has already been completed.

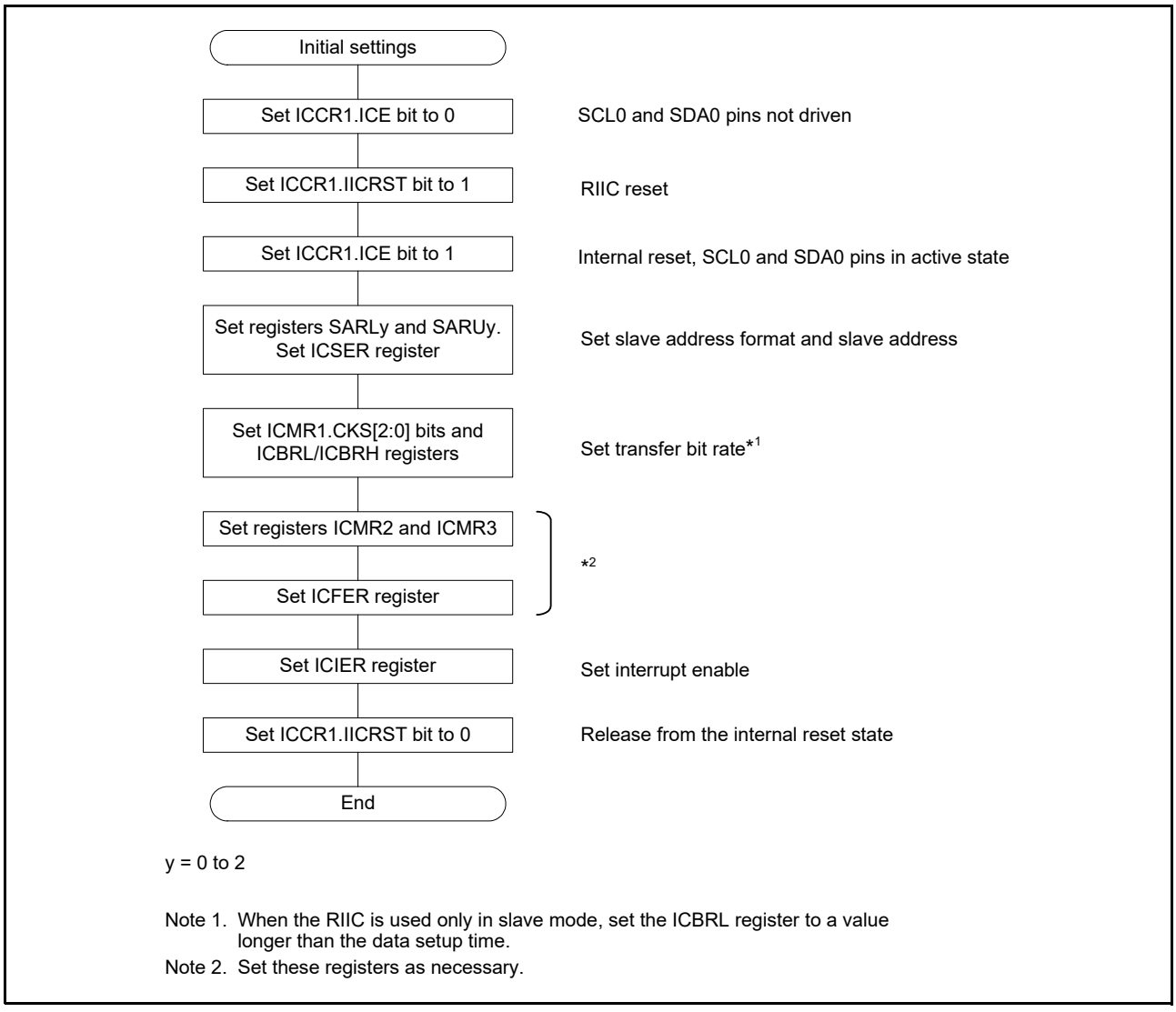


Figure 34.4 Example of RIIC Initialization Flowchart

34.3.3 Master Transmit Operation

In master transmit operation, the RIIC generates clock signals and sends data as the master device, and the slave device returns acknowledgments. Figure 34.5 shows an example of usage of master transmission and Figure 34.6 to Figure 34.8 show the timing of operations in master transmission.
The following describes the procedure and operations for master transmission.

(1) Initial settings. For details, refer to section 34.3.2, Initial Settings.
(2) Read the ICCR2.BBSY flag to check that the bus is open, and then set the ICCR2.ST bit to 1 (requests to generate a start condition). Upon receiving the request, the RIIC generates a start condition. At the same time, the BBSY flag and the ICSR2.START flag are automatically set to 1 and the ST bit is automatically set to 0. At this time, if the start condition is detected and the internal levels for the SDA output state and the levels on the SDA0 line have matched while the ST bit is 1, the RIIC recognizes that generating of the start condition as requested by the ST bit has been successfully completed, and bits MST and TRS in the ICCR2 register are automatically set to 1, placing the RIIC in master transmit mode. The ICSR2.TDRE flag is also automatically set to 1 in response to setting of the TRS bit to 1.
(3) Check that the ICSR2.TDRE flag is 1, and then write the value for transmission (the slave address and the R/W# bit) to the ICDRT register. Once the data for transmission are written to the ICDRT register, the TDRE flag is automatically set to 0, the data are transferred from the ICDRT register to the ICDRS register, and the TDRE flag is again set to 1. After the byte containing the slave address and R/W# bit has been transmitted, the value of the TRS bit is automatically updated to select master transmit or master receive mode in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 0, the RIIC continues in master transmit mode.
Because the ICSR2.NACKF flag being 1 at this time indicates that no slave device recognized the address or there was an error in communications, write 1 to the ICCR2.SP bit to generate a stop condition.
For data transmission with an address in the 10-bit format, start by writing 1111 0b, the 2 higher-order bits of the slave address, and W to the ICDRT register as the first address transmission. Then, as the second address transmission, write the 8 lower-order bits of the slave address to the ICDRT register.
(4) After confirming that the ICSR2.TDRE flag is 1, write the data for transmission to the ICDRT register. The RIIC automatically holds the SCL0 line low until the data for transmission are ready or a stop condition is generated.
(5) After all bytes of data for transmission have been written to the ICDRT register, wait until the ICSR2.NACKF or ICSR2.TEND flag becomes 1, and then set the ICCR2.SP bit to 1 (requests to generate a stop condition). Upon receiving a stop condition generation request, the RIIC generates the stop condition.
(6) Upon detecting the stop condition, the RIIC automatically sets bits MST and TRS in the ICCR2 register to 00b and enters slave receive mode. Furthermore, it automatically sets the TDRE and TEND flags to 0, and sets the ICSR2.STOP flag to 1.
(7) After checking that the ICSR2.STOP flag is 1, set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

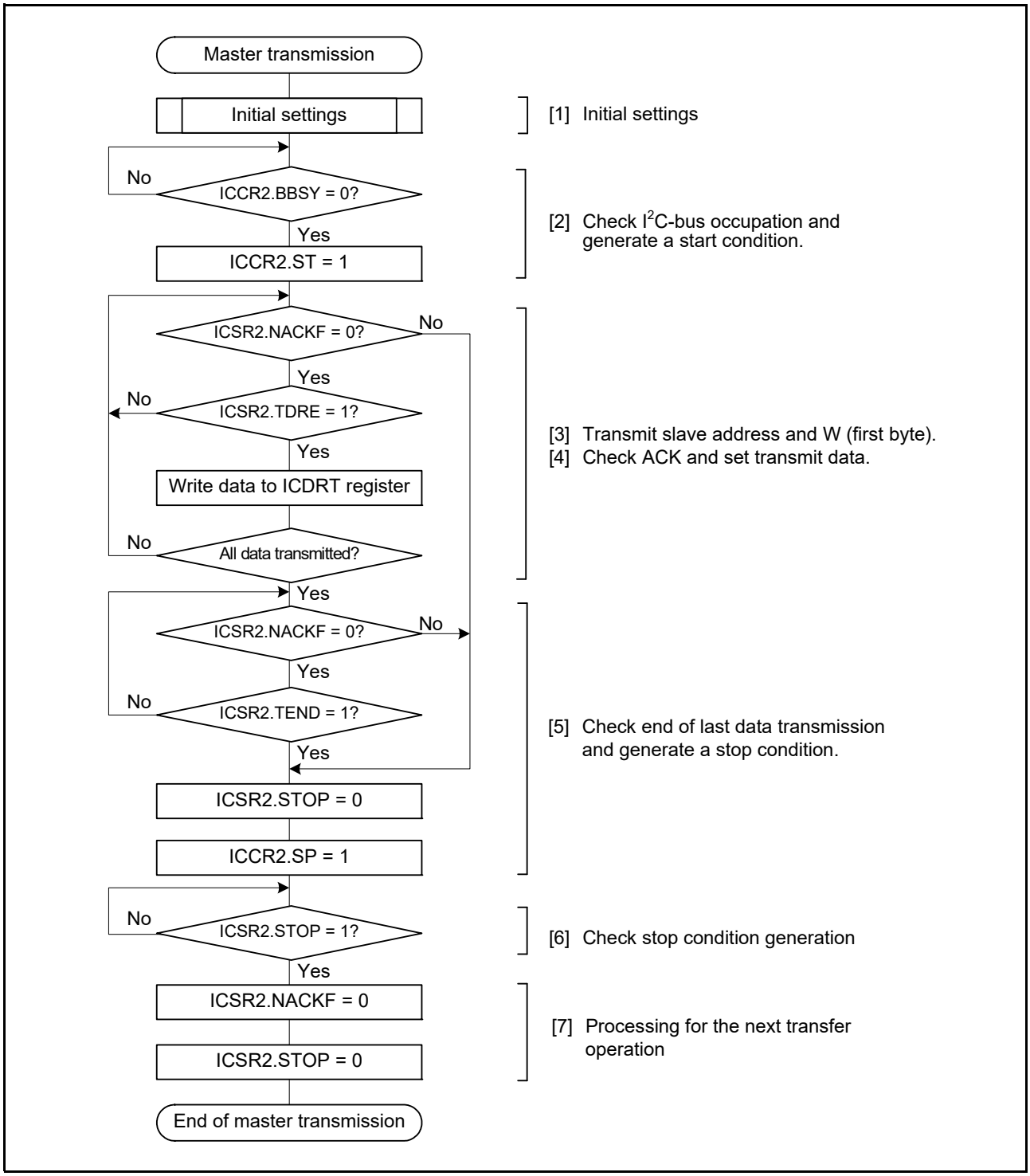


Figure 34.5 Example of Master Transmission Flowchart

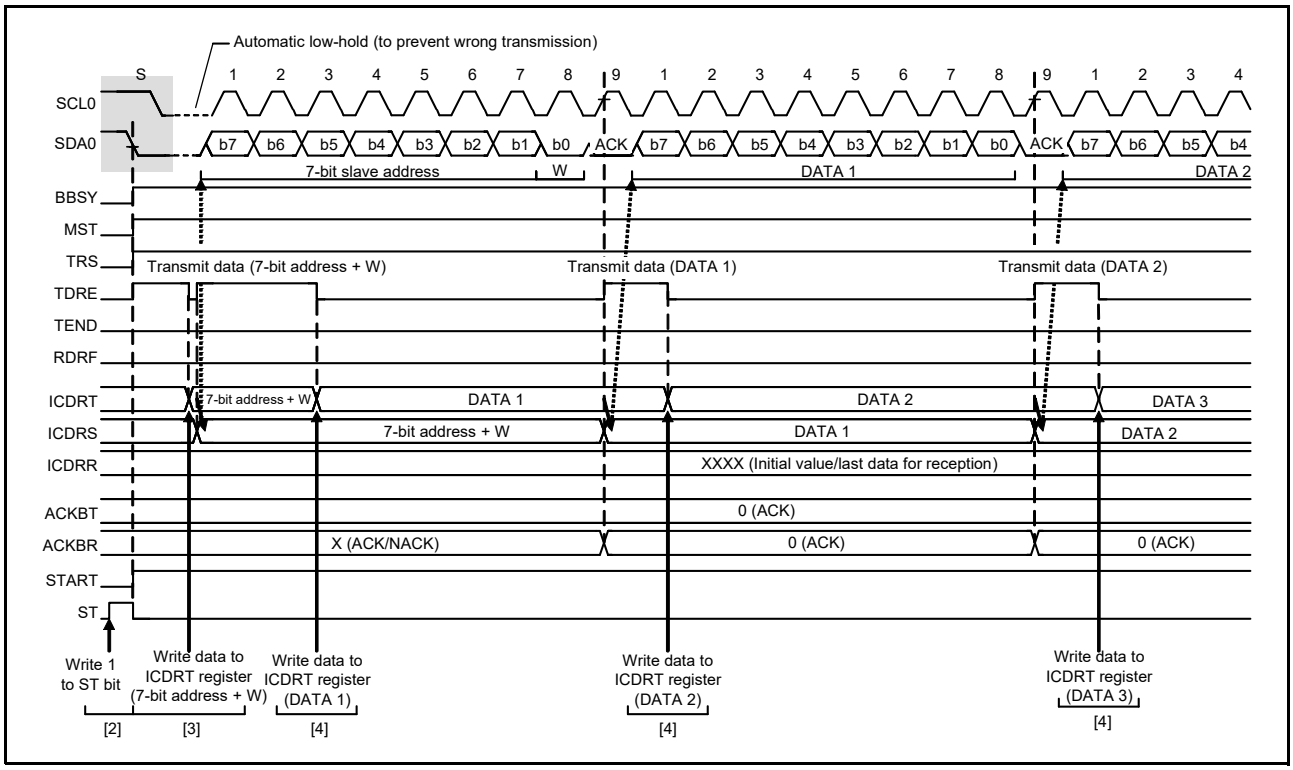


Figure 34.6 Master Transmit Operation Timing (1) (7-Bit Address Format)

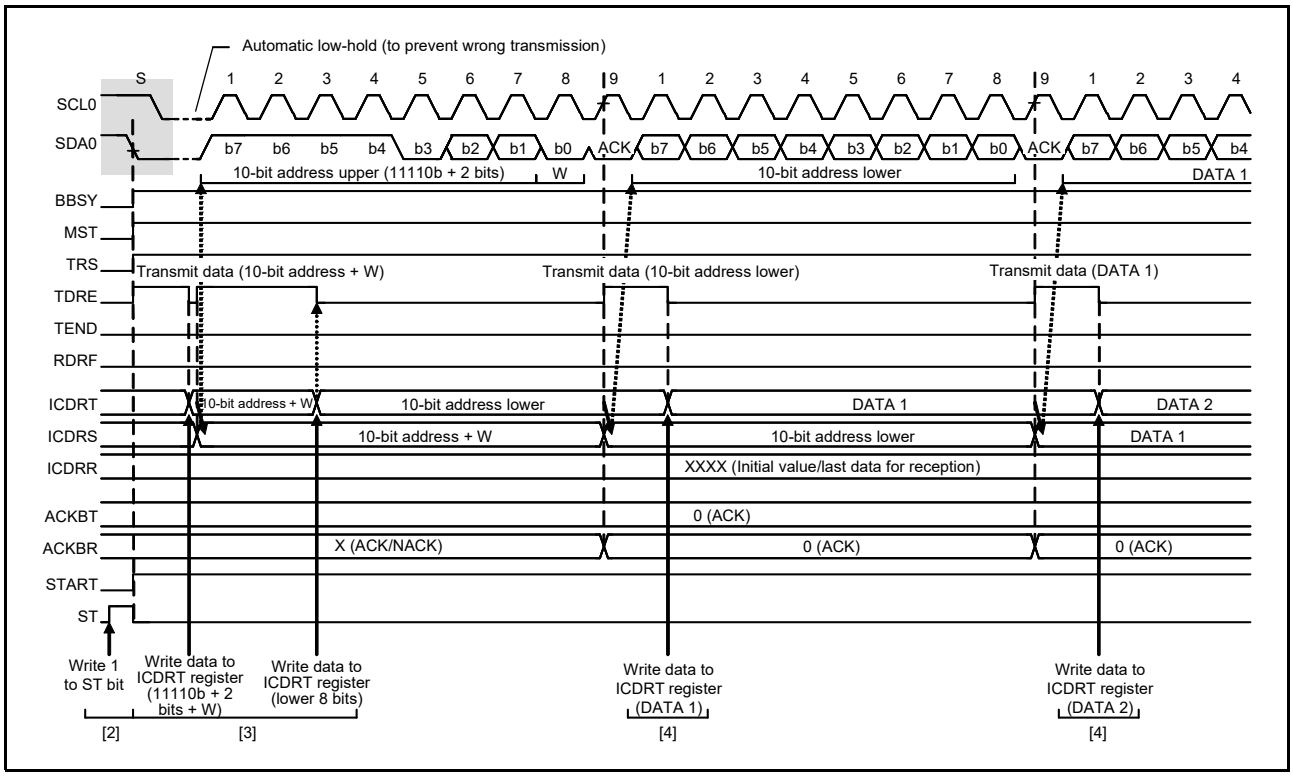


Figure 34.7 Master Transmit Operation Timing (2) (10-Bit Address Format)

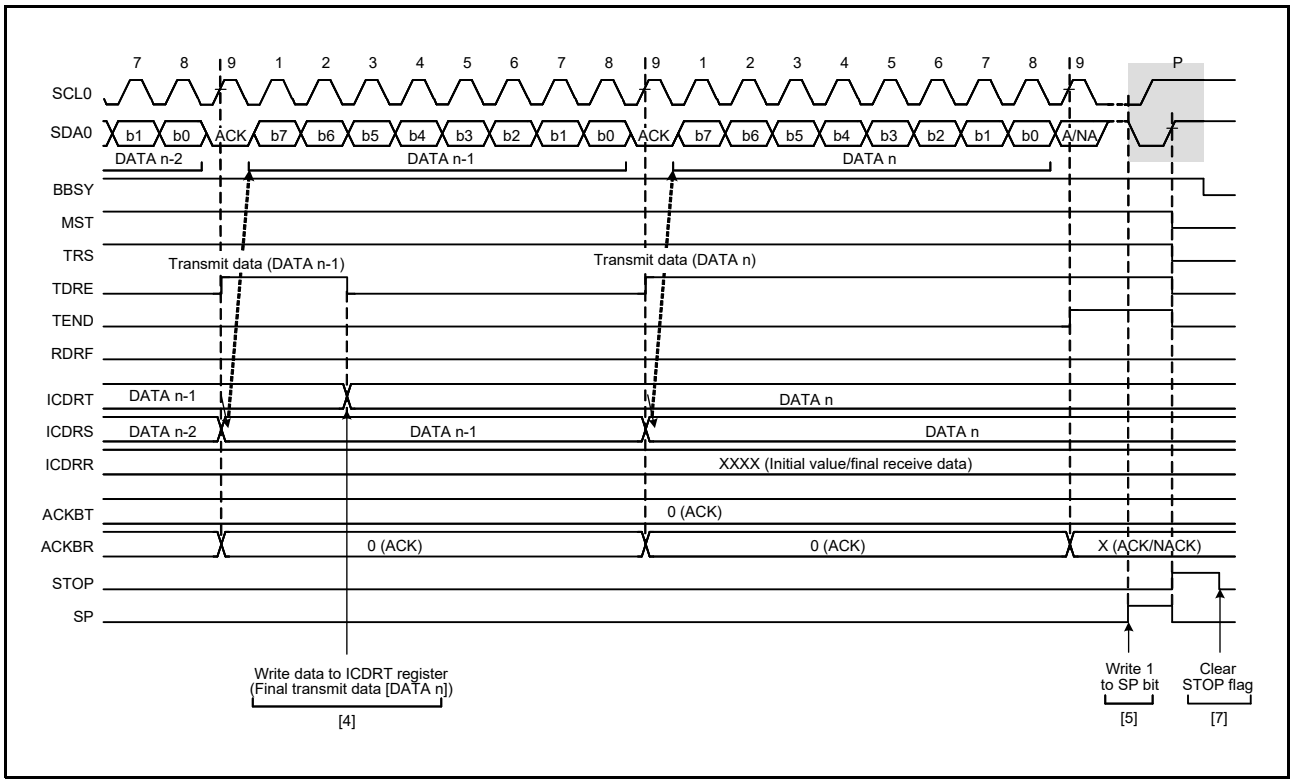


Figure 34.8 Master Transmit Operation Timing (3)

34.3.4 Master Receive Operation

In master receive operation, the RIIC as a master device generates clock signals, receives data from the slave device, and returns acknowledgments. Because the RIIC must start by sending a slave address to the corresponding slave device, this part of the procedure is performed in master transmit mode, but the subsequent steps are in master receive mode.
Figure 34.9 and Figure 34.10 show examples of usage of master reception (7-bit address format) and Figure 34.11 to Figure 34.13 show the timing of operations in master reception.
The following describes the procedure and operations for master reception.

(1) Initial settings. For details, refer to section 34.3.2, Initial Settings.

(2) Read the ICCR2.BBSY flag to check that the bus is open, and then set the ICCR2.ST bit to 1 (requests to generate a start condition). Upon receiving the request, the RIIC generates a start condition. When the RIIC detects the start condition, the BBSY flag and the ICSR2.START flag are automatically set to 1 and the ST bit is automatically set to 0. At this time, if the start condition is detected and the levels for the SDA output and the levels on the SDA0 line have matched while the ST bit is 1, the RIIC recognizes that generating of the start condition as requested by the ST bit has been successfully completed, and bits MST and TRS in the ICCR2 register are automatically set to 1, placing the RIIC in master transmit mode. The ICSR2.TDRE flag is also automatically set to 1 in response to setting of the TRS bit to 1.

(3) Check that the ICSR2.TDRE flag is 1, and then write the value for transmission (the first byte indicates the slave address and value of the R/W# bit) to the ICDRT register. Once the data for transmission are written to the ICDRT register, the TDRE flag is automatically set to 0, the data are transferred from the ICDRT register to the ICDRS register, and the TDRE flag is again set to 1. Once the byte containing the slave address and R/W# bit has been transmitted, the value of the ICCR2.TRS bit is automatically updated to select transmit or receive mode in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 1, the TRS bit is set to 0 on the rising edge of the ninth SCL, placing the RIIC in master receive mode. At this time, the TDRE flag is set to 0 and the ICSR2.RDRF flag is automatically set to 1.

Because the ICSR2.NACKF flag being 1 at this time indicates that no slave device recognized the address or there was an error in communications, write 1 to the ICCR2.SP bit to generate a stop condition.
For master reception from a device with a 10-bit address, start by using master transmission to send the 10-bit address, and then generate a restart condition. After that, transmitting 1111 0b, the two higher-order bits of the slave address, and the R bit places the RIIC in master receive mode.

(4) Dummy read the ICDRR register after confirming that the ICSR2.RDRF flag is 1; this makes the RIIC start output of the SCL and start data reception.

(5) After 1 byte of data has been received, the ICSR2.RDRF flag is set to 1 on the rising edge of the eighth or ninth SCL (the clock signal) as selected by the ICMR3.RDRFS bit. Reading the ICDRR register at this time will produce the received data, and the RDRF flag is automatically set to 0 at the same time. Furthermore, the value of the acknowledgment bit received during the ninth SCL is returned as the value set in the ICMR3.ACKBT bit. Furthermore, if the next byte to be received is the next to last byte, set the ICMR3.WAIT bit to 1 (for wait insertion) before reading the ICDRR register (containing the second byte from last). As well as enabling NACK output even in the case of delays in processing to set the ICMR3.ACKBT bit to 1 (NACK) in step (6), due to other interrupts, etc., this fixes the SCL0 line to low on the falling edge of the ninth clock pulse in reception of the last byte, so the state is such that generating a stop condition is possible.

(6) When the ICMR3.RDRFS bit is 0 and the slave device must be notified that it is to end transfer for data reception after transfer of the next (final) byte, set the ICMR3.ACKBT bit to 1 (NACK).

(7) After reading the byte before last from the ICDRR register, if the value of the ICSR2.RDRF flag is confirmed to be 1, write 1 to the ICCR2.SP bit (requests to generate a stop condition) and then read the last byte from the ICDRR register. When the ICDRR register is read, the RIIC is released from the wait state and generates the stop condition after low-level output in the ninth clock pulse is completed or the SCL0 line is released from the low-hold state.

(8) Upon detecting the stop condition, the RIIC automatically sets bits MST and TRS in the ICCR2 register to 00b and enters slave receive mode. Furthermore, detection of the stop condition leads to setting of the ICSR2.STOP flag to 1.

(9) After checking that the ICSR2.STOP flag is 1, set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

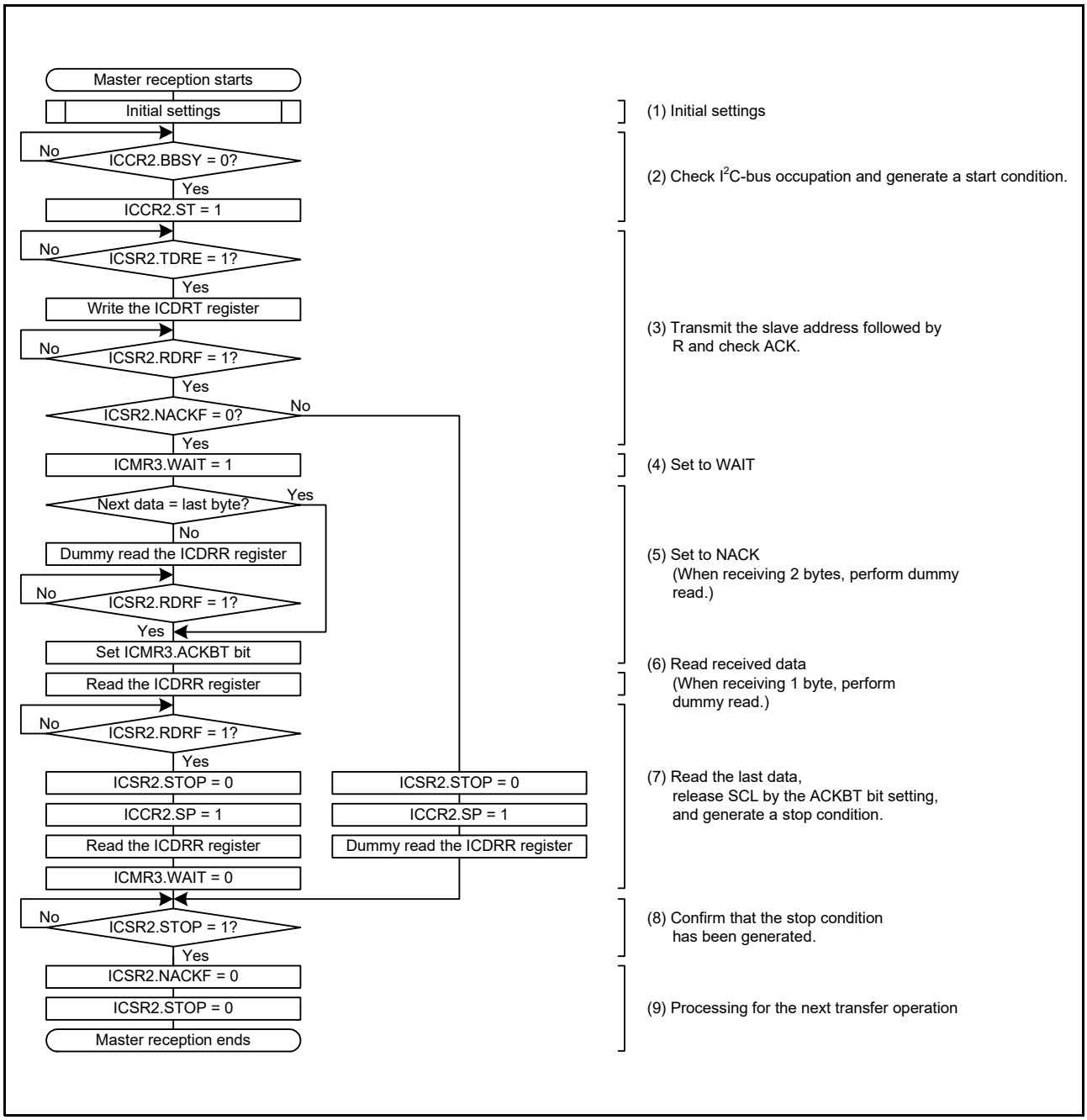


Figure 34.9 Example of Master Reception (7-Bit Address Format, 1 or 2 bytes)

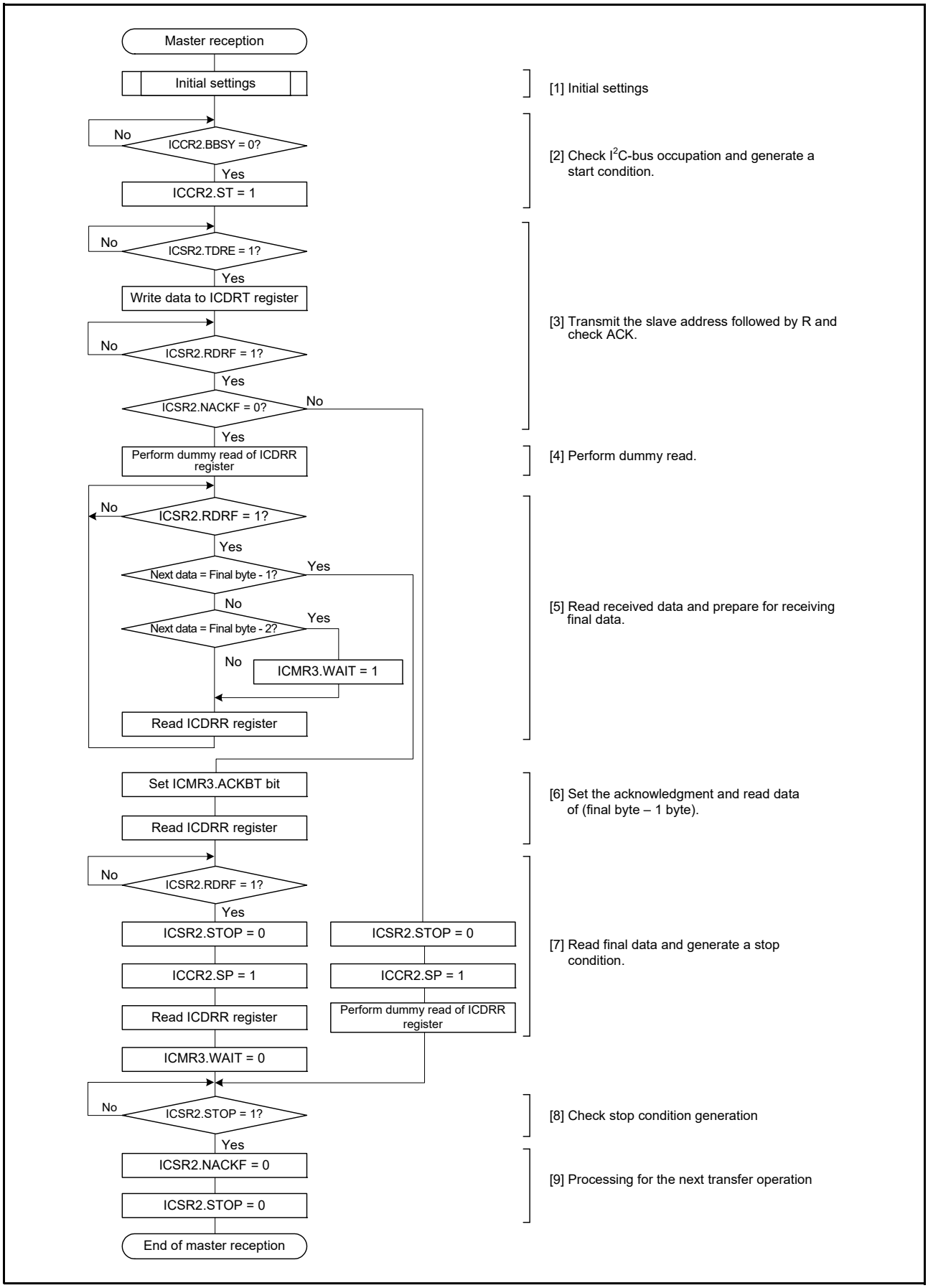


Figure 34.10 Example of Master Reception (7-Bit Address Format, 3 Bytes or More)

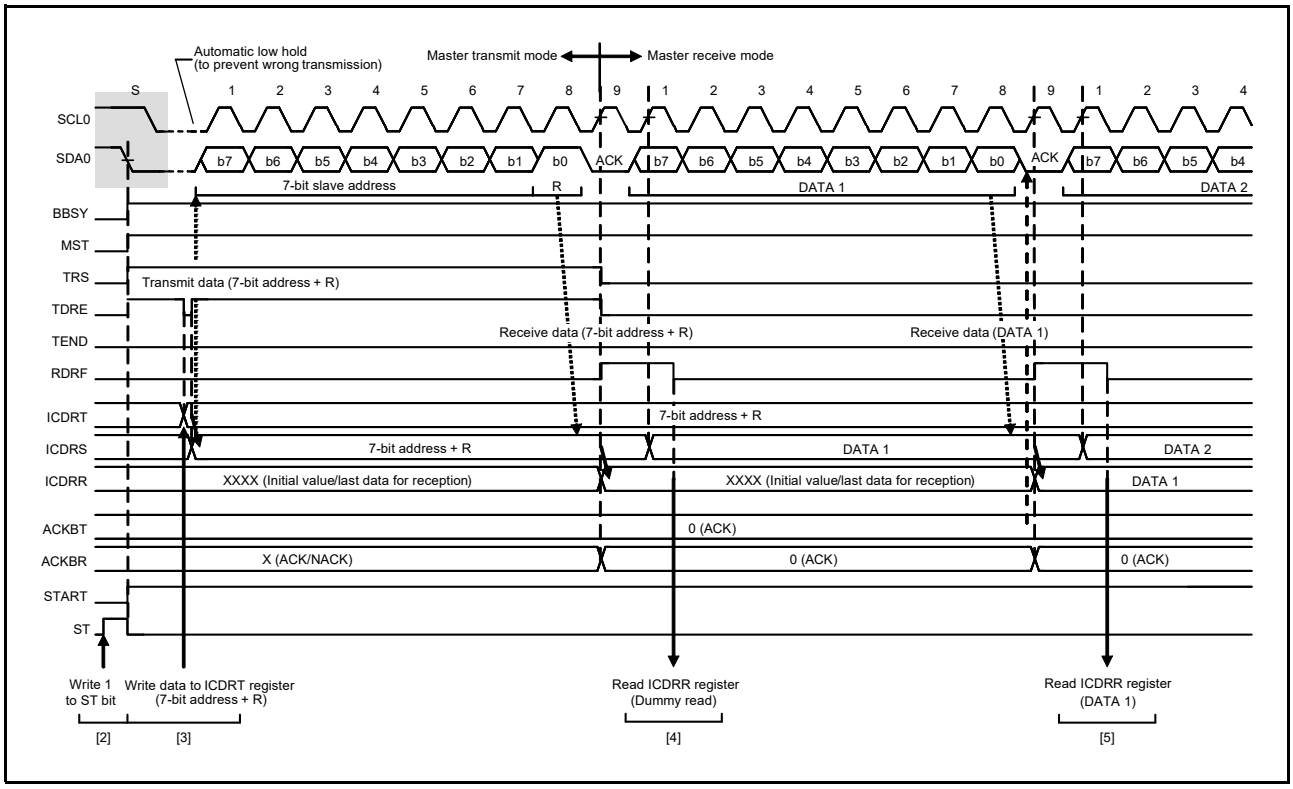


Figure 34.11 Master Receive Operation Timing (1) (7-Bit Address Format, When RDRFS bit is 0)

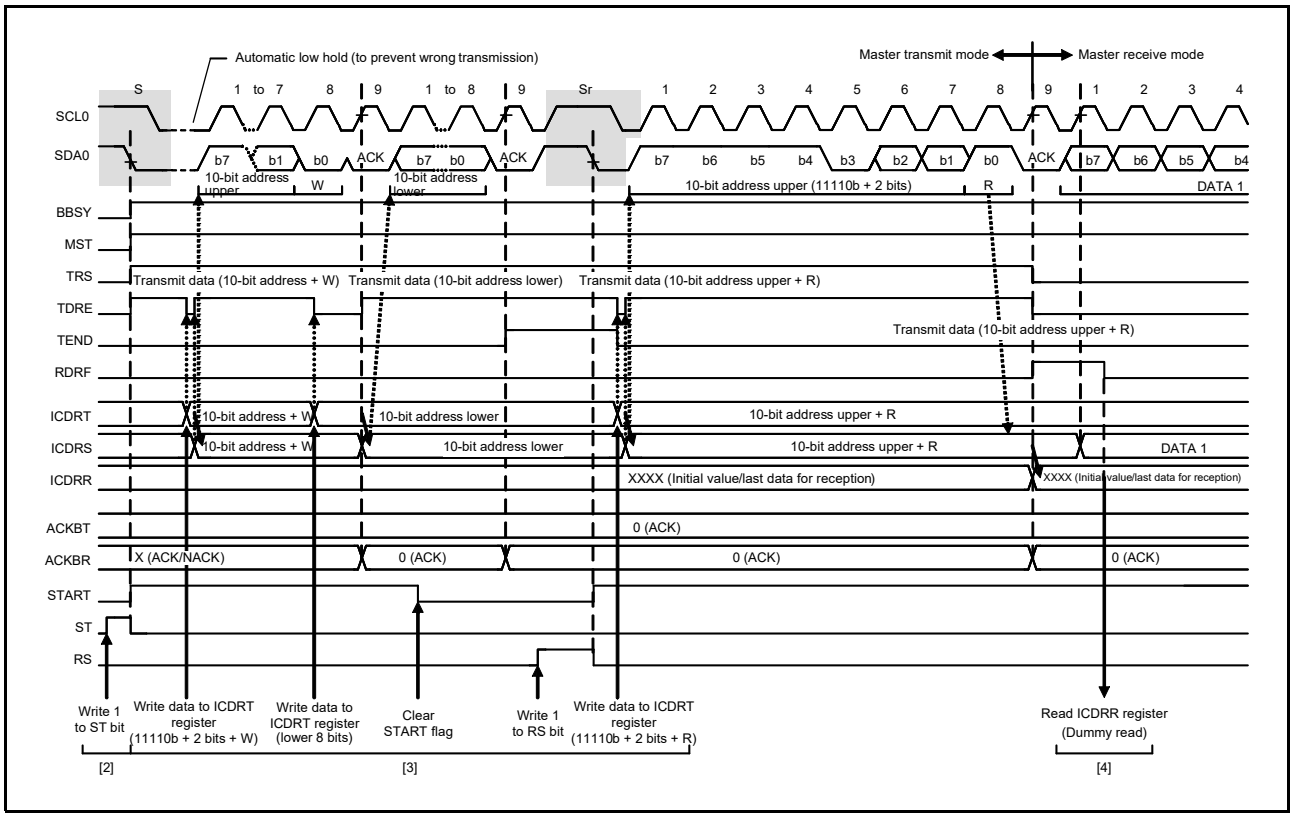


Figure 34.12 Master Receive Operation Timing (2) (10-Bit Address Format, When RDRFS bit is 0)

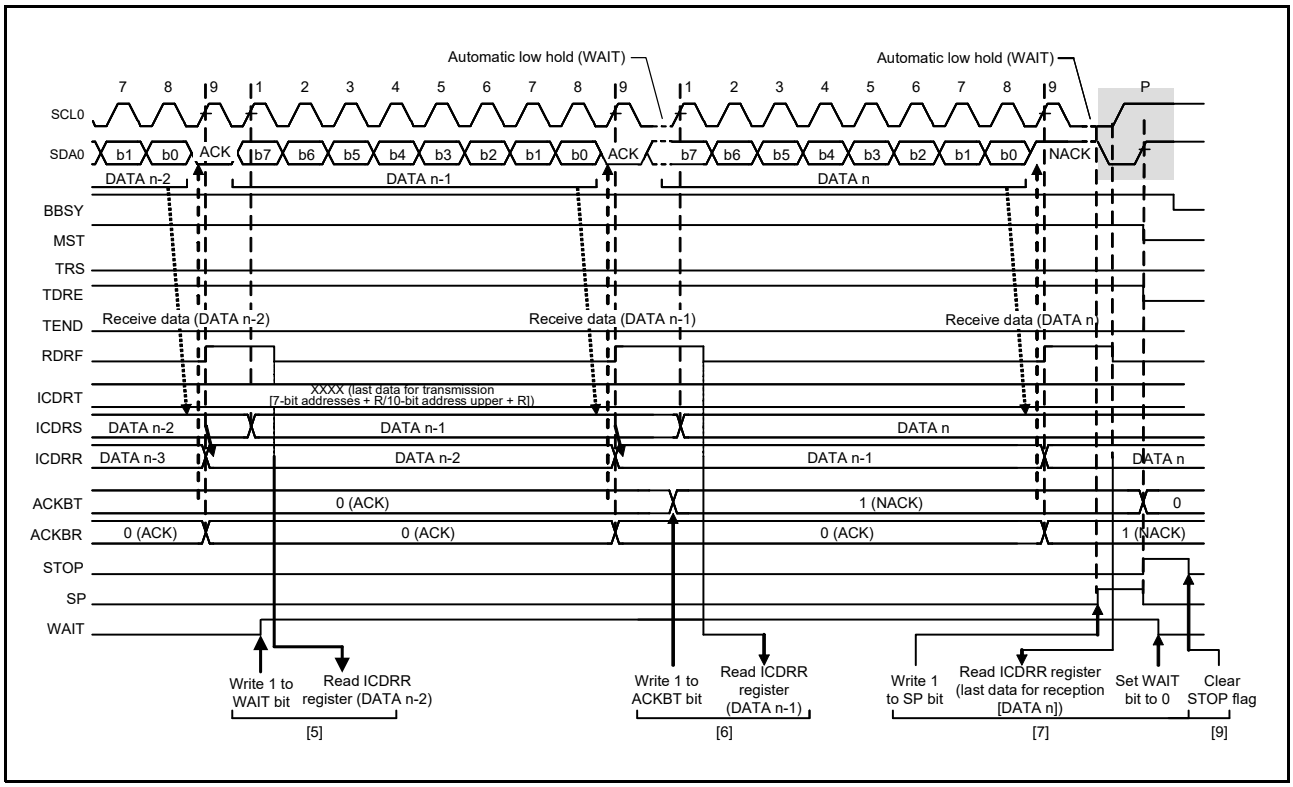


Figure 34.13 Master Receive Operation Timing (3) (When RDRFS bit is 0)

34.3.5 Slave Transmit Operation

In slave transmit operation, the master device outputs the SCL, the RIIC transmits data as a slave device, and the master device returns acknowledgments.

Figure 34.14 shows an example of usage of slave transmission and Figure 34.15 and Figure 34.16 show the timing of operations in slave transmission.

The following describes the procedure and operations for slave transmission.

(1) Initial settings. For details, refer to section 34.3.2, Initial Settings.
After initial settings, the RIIC will stay in the standby state until it receives a slave address that it matches.

(2) After receiving a matching slave address, the RIIC sets one of the corresponding bits ICSR1.HOA, GCA, and AASy (y = 0 to 2) to 1 on the rising edge of the ninth SCL (the clock signal) and outputs the value set in the ICMR3.ACKBT bit to the acknowledgment bit on the ninth SCL. If the value of the R/W# bit that was also received at this time is 1, the RIIC automatically places itself in slave transmit mode by setting both the ICCR2.TRS bit and the ICSR2.TDRE flag to 1.

(3) After the ICSR2.TDRE flag is confirmed to be 1, write the data for transmission to the ICDRT register. At this time, if the RIIC does not receive ACK from the master device (receives an NACK signal) while the ICFER.NACKE bit is 1, the RIIC suspends transfer of the next data.

(4) Wait until the ICSR2.TEND flag is set to 1 while the ICSR2.TDRE flag is 1, after the ICSR2.NACKF flag is set to 1 or the last byte for transmission is written to the ICDRT register. When the ICSR2.NACKF flag or the TEND flag is 1, the RIIC drives the SCL0 line low on the falling edge of ninth SCL.

(5) When the ICSR2.NACKF flag or the ICSR2.TEND flag is 1, dummy read the ICDRR register to complete the processing. This releases the SCL0 line.

(6) Upon detecting the stop condition, the RIIC automatically sets bits ICSR1.HOA, GCA, and AASy (y = 0 to 2), flags ICSR2.TDRE and TEND, and the ICCR2.TRS bit to 0, and enters slave receive mode.

(7) After checking that the ICSR2.STOP flag is 1, set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

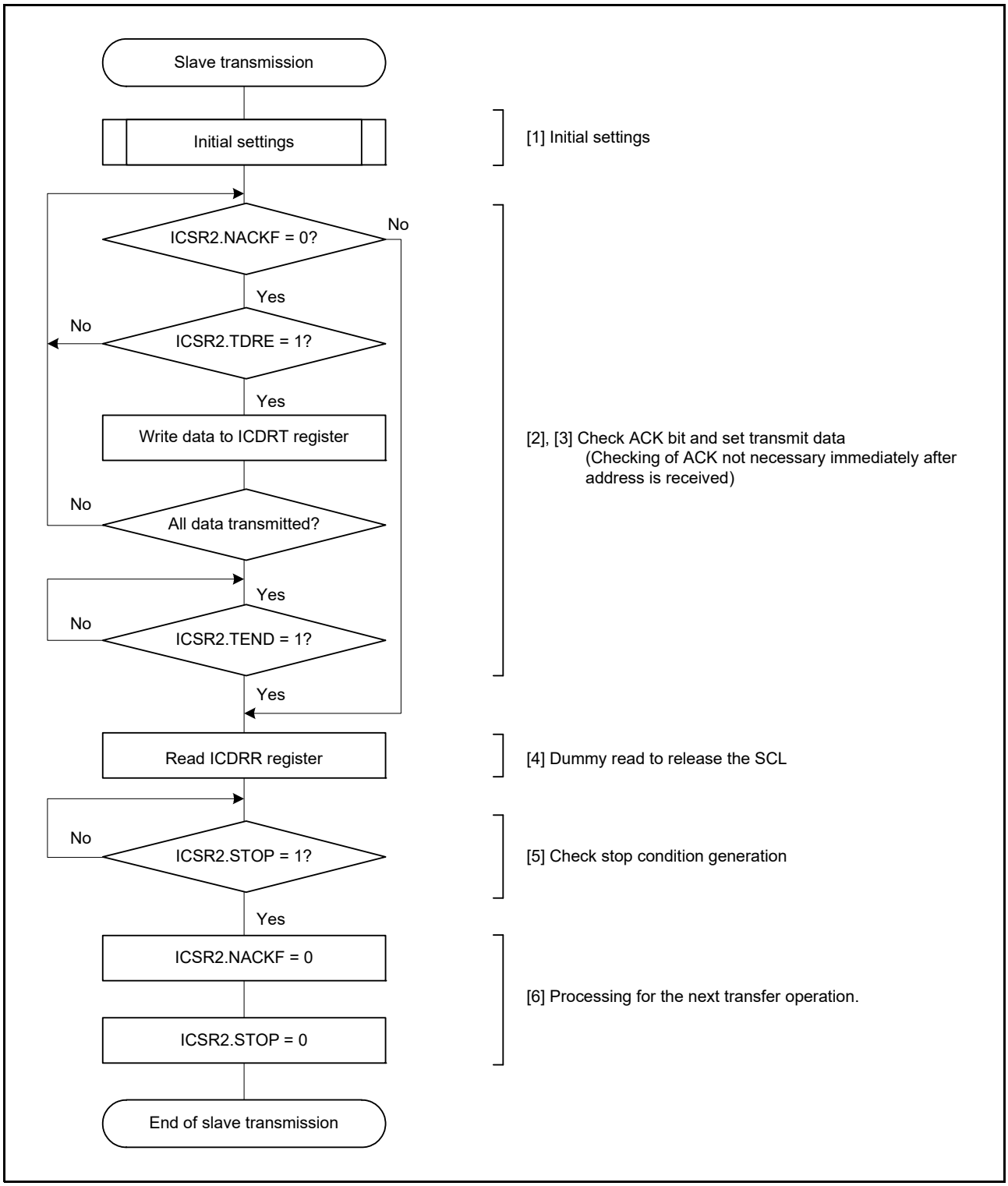


Figure 34.14 Example of Slave Transmission Flowchart

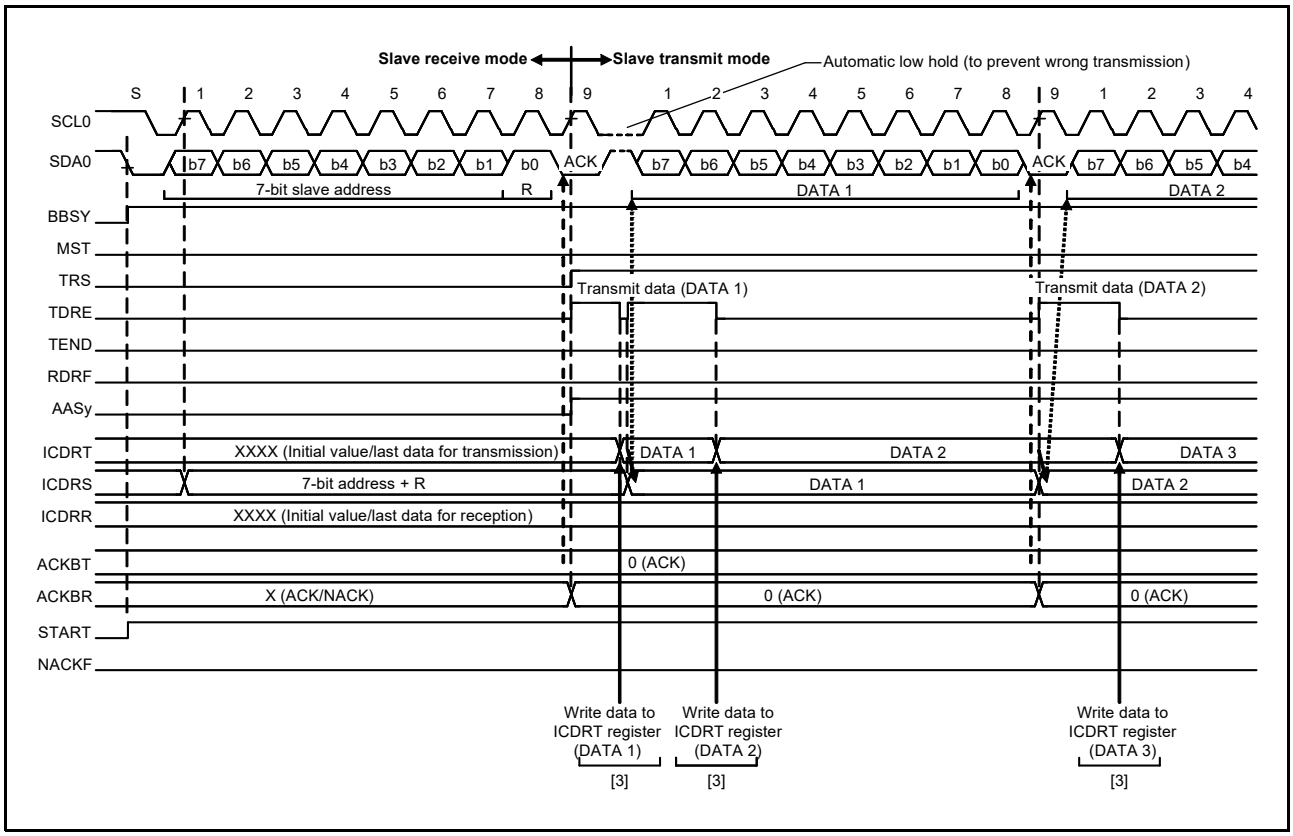


Figure 34.15 Slave Transmit Operation Timing (1) (7-Bit Address Format)

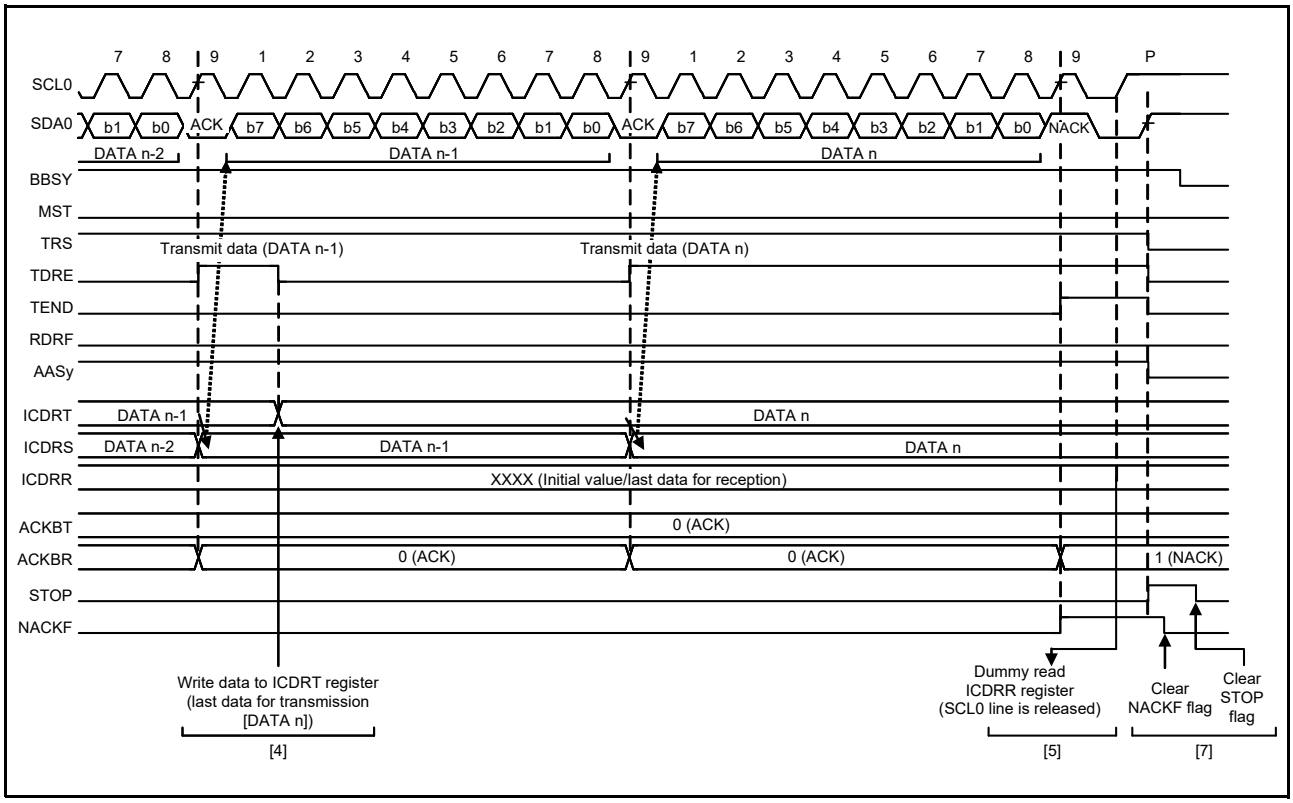


Figure 34.16 Slave Transmit Operation Timing (2)

34.3.6 Slave Receive Operation

In slave receive operation, the master device outputs the SCL and transmit data, and the RIIC returns acknowledgments as a slave device.

Figure 34.17 shows an example of usage of slave reception and Figure 34.18 and Figure 34.19 show the timing of operations in slave reception.

The following describes the procedure and operations for slave reception.

(1) Initial settings. For details, refer to section 34.3.2, Initial Settings.
After initial settings, the RIIC will stay in the standby state until it receives a slave address that it matches.

(2) After receiving a matching slave address, the RIIC sets one of the corresponding bits ICSR1.HOA, GCA, and AASy (y = 0 to 2) to 1 on the rising edge of the ninth SCL (the clock signal) and outputs the value set in the ICMR3.ACKBT bit to the acknowledgment bit on the ninth SCL. If the value of the R/W# bit that was also received at this time is 0, the RIIC continues to place itself in slave receive mode and sets the ICSR2.RDRF flag to 1.

(3) After the ICSR2.STOP flag is confirmed to be 0 and the ICSR2.RDRF flag to be 1, dummy read the ICDRR register (the dummy value consists of the slave address and R/W# bit when the 7-bit address format is selected, or the lower 8 bits when the 10-bit address format is selected).

(4) When the ICDRR register is read, the RIIC automatically sets the ICSR2.RDRF flag to 0. If reading of the ICDRR register is delayed and a next byte is received while the RDRF flag is still set to 1, the RIIC holds the SCL0 line low from one SCL cycle before the timing with which RDRF should be set. In this case, reading the ICDRR register releases the SCL0 line from being held low.
When the ICSR2.STOP flag is 1 and the ICSR2.RDRF flag is also 1 or when all the data is completely received, read the ICDRR register.

(5) Upon detecting the stop condition, the RIIC automatically clears bits ICSR1.HOA, GCA, and AASy (y = 0 to 2) to 0.

(6) After checking that the ICSR2.STOP flag is 1, set the ICSR2.STOP flag to 0 for the next transfer operation.

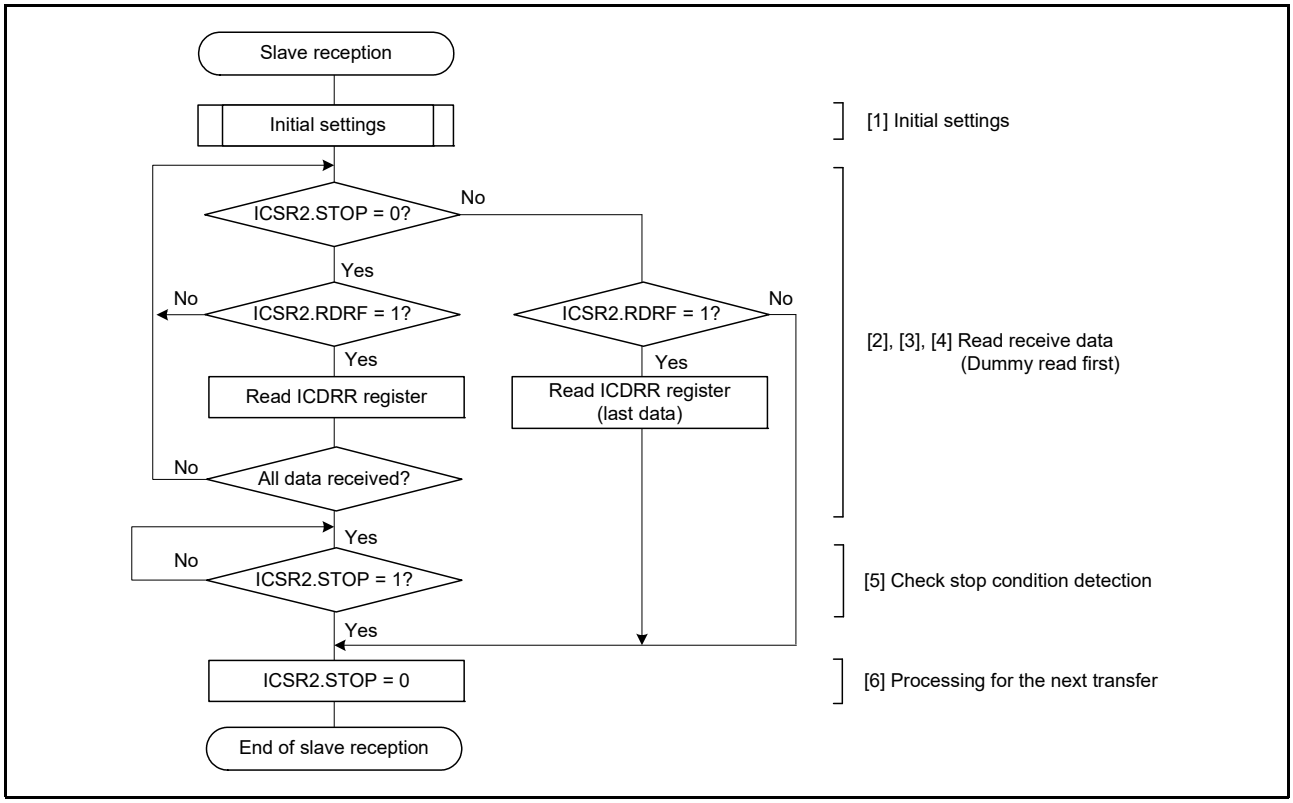


Figure 34.17 Example of Slave Reception Flowchart

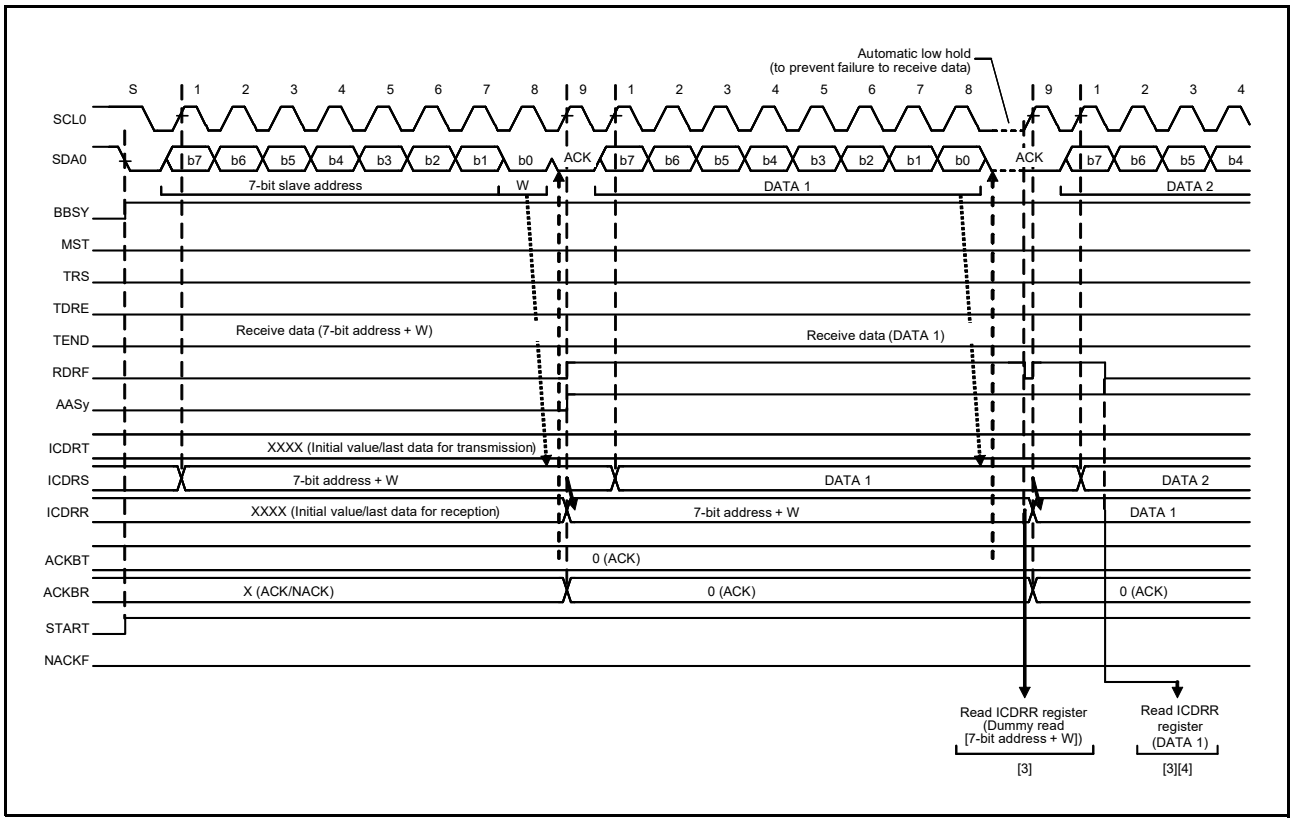


Figure 34.18 Slave Receive Operation Timing (1) (7-Bit Address Format, when RDRFS bit is 0)

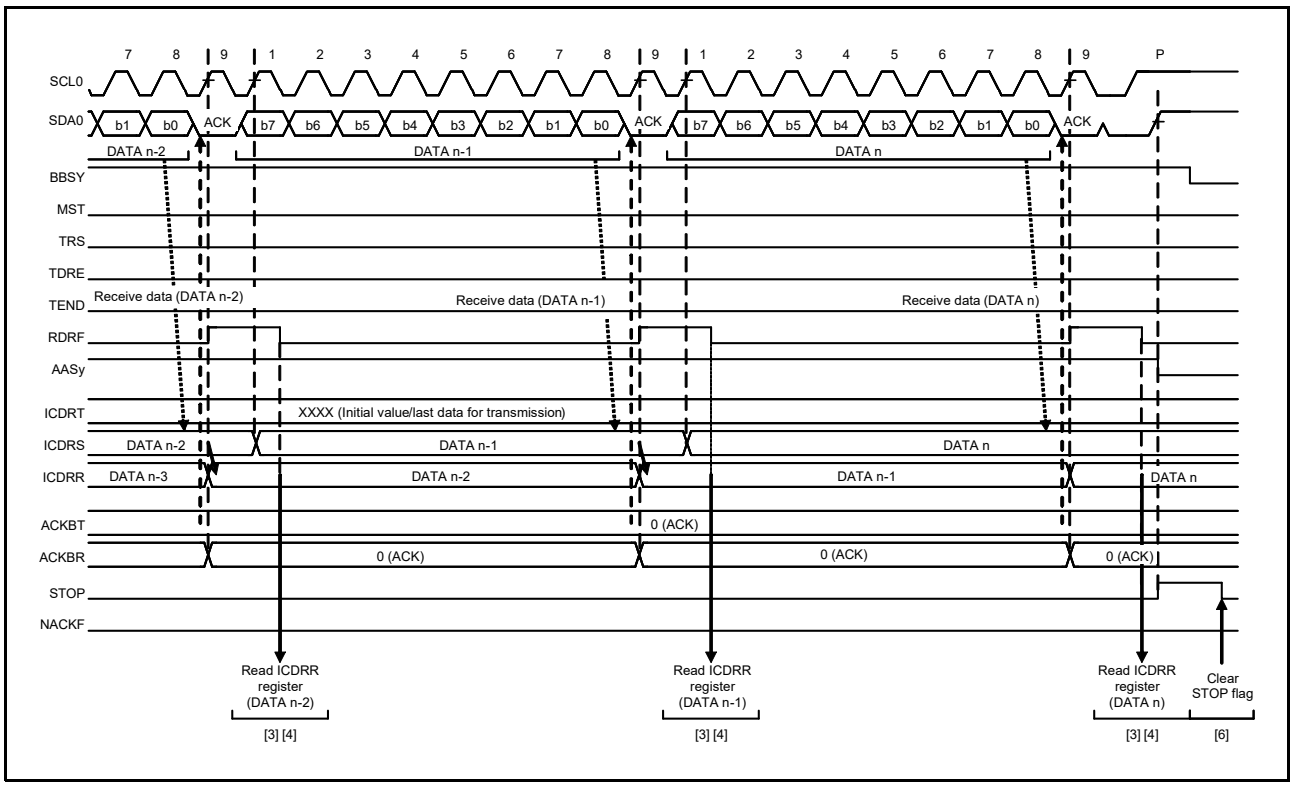


Figure 34.19 Slave Receive Operation Timing (2) (when RDRFS bit is 0)

34.4 SCL Synchronization Circuit

In generation of the SCL, the RIIC starts counting out the value for width at high level specified in the ICBRH register when it detects a rising edge on the SCL0 line and drives the SCL0 line low once counting of the width at high level is complete. When the RIIC detects the falling edge of the SCL0 line, it starts counting out the width at low period specified in the ICBRL register, and then stops driving the SCL0 line (releases the line) once counting of the width at low level is complete. The SCL is thus generated.

If multiple master devices are connected to the I2C-bus, a collision of SCL signals may arise due to contention with another master device. In such cases, the master devices have to synchronize their SCL signals. Since synchronization of the SCL signals must be handled bit by bit, the RIIC is equipped with a facility (the SCL synchronization circuit) for obtaining bit-by-bit synchronization of the SCL signals by monitoring the SCL0 line while in master mode.

When the RIIC has detected a rising edge on the SCL0 line and thus started counting out the width at high level specified in the ICBRH register, and the level on the SCL0 line falls because an SCL signal is being generated by another master device, the RIIC stops counting when it detects the falling edge, drives the level on the SCL0 line low, and starts counting out the width at low level specified in the ICBRL register. When the RIIC finishes counting out the width at low level, it stops driving the SCL0 line low (i.e. releases the line). At this time, if the width at low level of the SCL signal from the other master device is longer than the width at low level set in the RIIC, the width at low level of the SCL signal will be extended. Once the width at low level for the other master device has ended, the SCL signal rises because the SCL0 line has been released. When the RIIC finishes outputting the low period of the SCL, the SCL0 line is released and the SCL rises. That is, in cases of contention of SCL signals from more than one master, the width at high level of the SCL signal is synchronized with that of the clock having the narrower width, and the width at low level of the SCL signal is synchronized with that of the clock having the broader width. However, such synchronization of the SCL signal is only enabled when the ICFER.SCLE bit is set to 1.

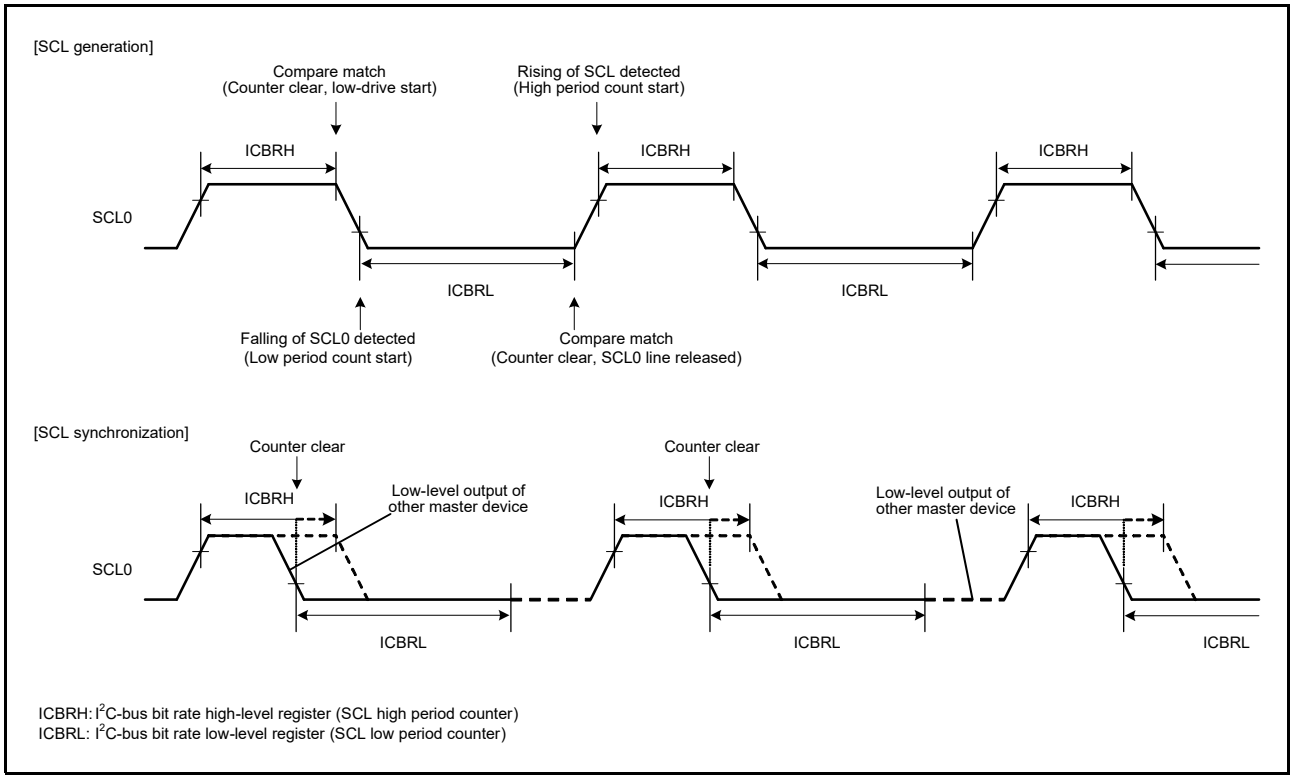


Figure 34.20 Generation and Synchronization of the SCL Signal from the RIIC

34.5 SDA Output Delay Function

The RIIC module incorporates a function for delaying output on the SDA line. The delay can be applied to all output (generation of the start, restart, and stop conditions, data, and the ACK and NACK signals) on the SDA line.

The SDA output delay function is used to delay the SDA output timing relative to falling edges of SCL to ensure that the SDA signal changes while the SCL is low and can be used to prevent erroneous operation of communications devices. This function is also used to satisfy the 300 ns (min.) data hold time prescribed by the SMBus specification.

The output delay function is enabled by setting the ICMR2.SDDL[2:0] bits to any value other than 000b, and disabled by setting the same bits to 000b.

When the SDA output delay function is enabled (the SDDL[2:0] bits are not “000b”), the SDA output delay counter counts the number of cycles set in the SDDL[2:0] bits of the count source selected by the ICMR2.DLCS bit (the internal reference clock (IICφ) or internal reference clock divided by 2 (IICφ/2)). On completion of counting of cycles of delay, the RIIC changes the bit being output as the SDA signal (generation of the start, restart, or stop condition, a new bit, or an ACK or NACK signal).

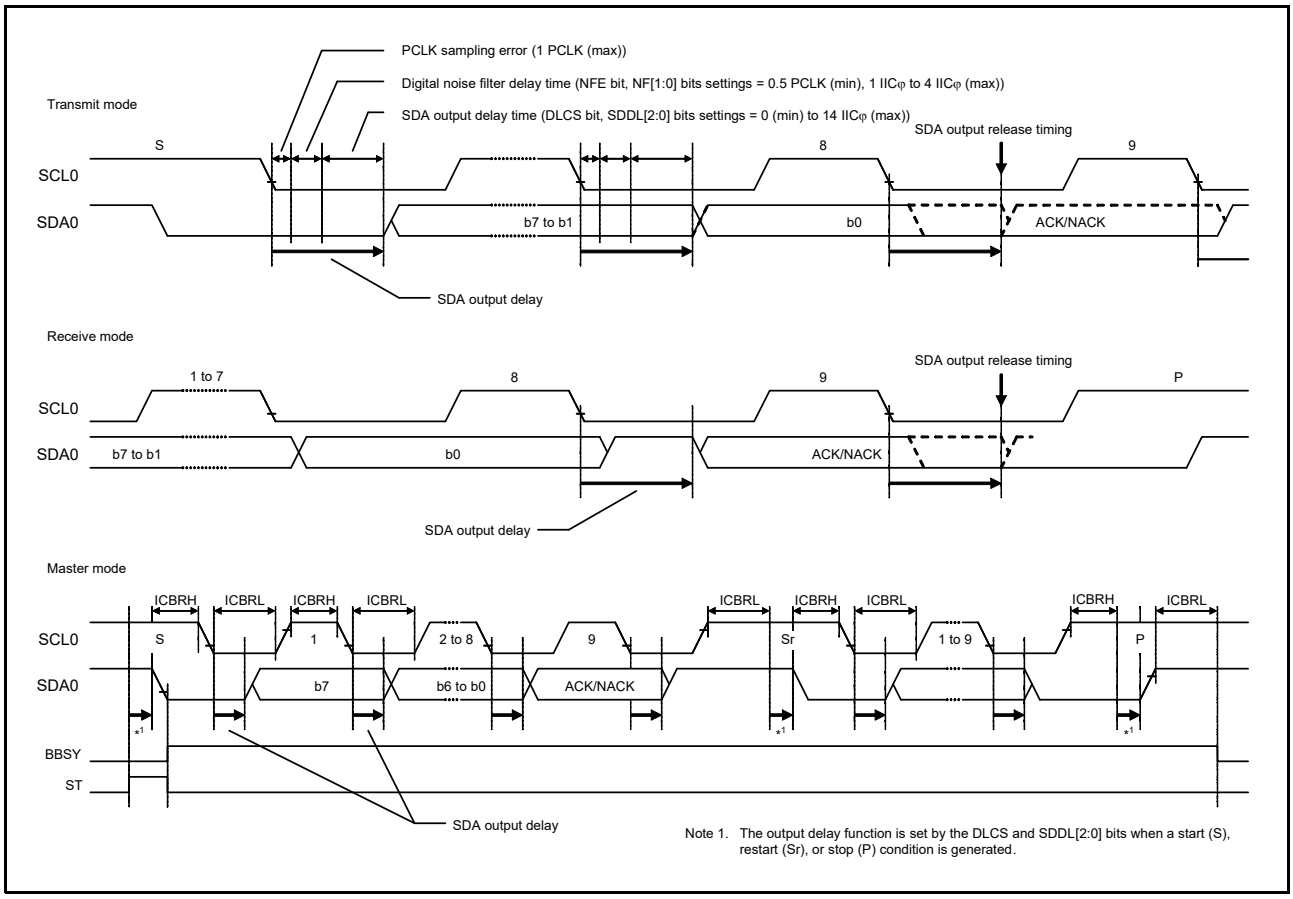


Figure 34.21 SDA Output Delay Function

34.6 Digital Noise Filters

The states of the SCL0 and SDA0 pins are conveyed to the internal circuitry through digital noise filters. Figure 34.22 is a block diagram of the digital noise filter.

The on-chip digital noise filter of each RIIC consists of four flip-flop circuit stages connected in series and a match detection circuit.

The number of effective stages in the digital noise filter is selected by the ICMR3.NF[1:0] bits. The selected number of effective stages determines the noise-filtering capability as a period from one to four cycles of IICφ.

The input signal to the SCL0 pin (or SDA0 pin) is sampled on falling edges of the IICφ signal. When the input signal level matches the output level for the number of effective flip-flop circuit stages selected by the ICMR3.NF[1:0] bits, the signal level is conveyed to the subsequent stages. If the signal levels do not match, the previous value is retained.

If the ratio between the frequency of the internal operating clock (PCLK) and the transfer rate is relatively small (e.g. data transfer at 400 kbps with PCLK = 4 MHz), the characteristics of the digital noise filter may lead to the elimination of needed signals as noise.

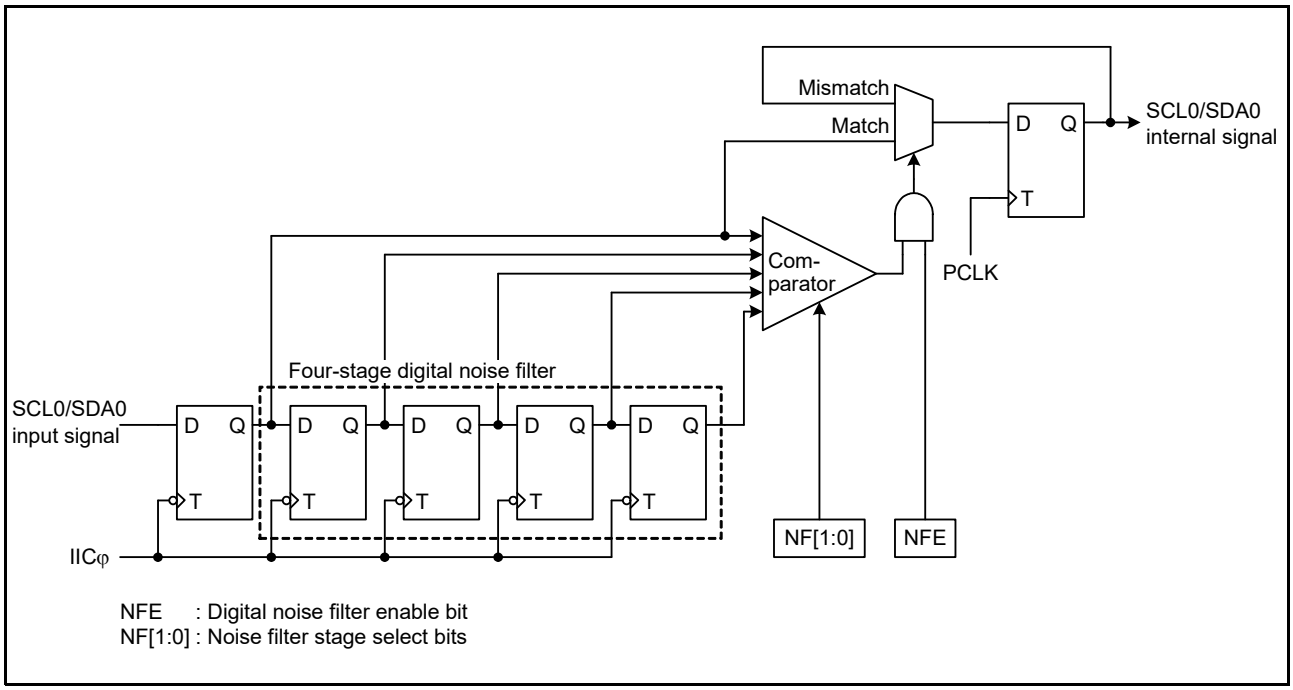


Figure 34.22 Block Diagram of the Digital Noise Filter

34.7 Address Match Detection

The RIIC can set three unique slave addresses in addition to the general call address and host address, and also can set 7-bit or 10-bit slave addresses.

34.7.1 Slave-Address Match Detection

The RIIC can set three unique slave addresses, and has a slave address detection function for each unique slave address. When the ICSER.SARyE bit (y = 0 to 2) is set to 1, the slave addresses set in registers SARUy and SARLy (y = 0 to 2) can be detected.

When the RIIC detects a match with its set slave address, the corresponding ICSR1.AASy flag (y = 0 to 2) is set to 1 on the rising edge of the ninth SCL, and the ICSR2.RDRF flag or the ICSR2.TDRE flag is set to 1 according to the level of the R/W# bit. This causes a receive data full interrupt (RXI) or transmit data empty interrupt (TXI) to be generated. The AASy flag is used to identify which slave address has been specified.

Figure 34.23 to Figure 34.25 show the AASy flag set timing in three cases.

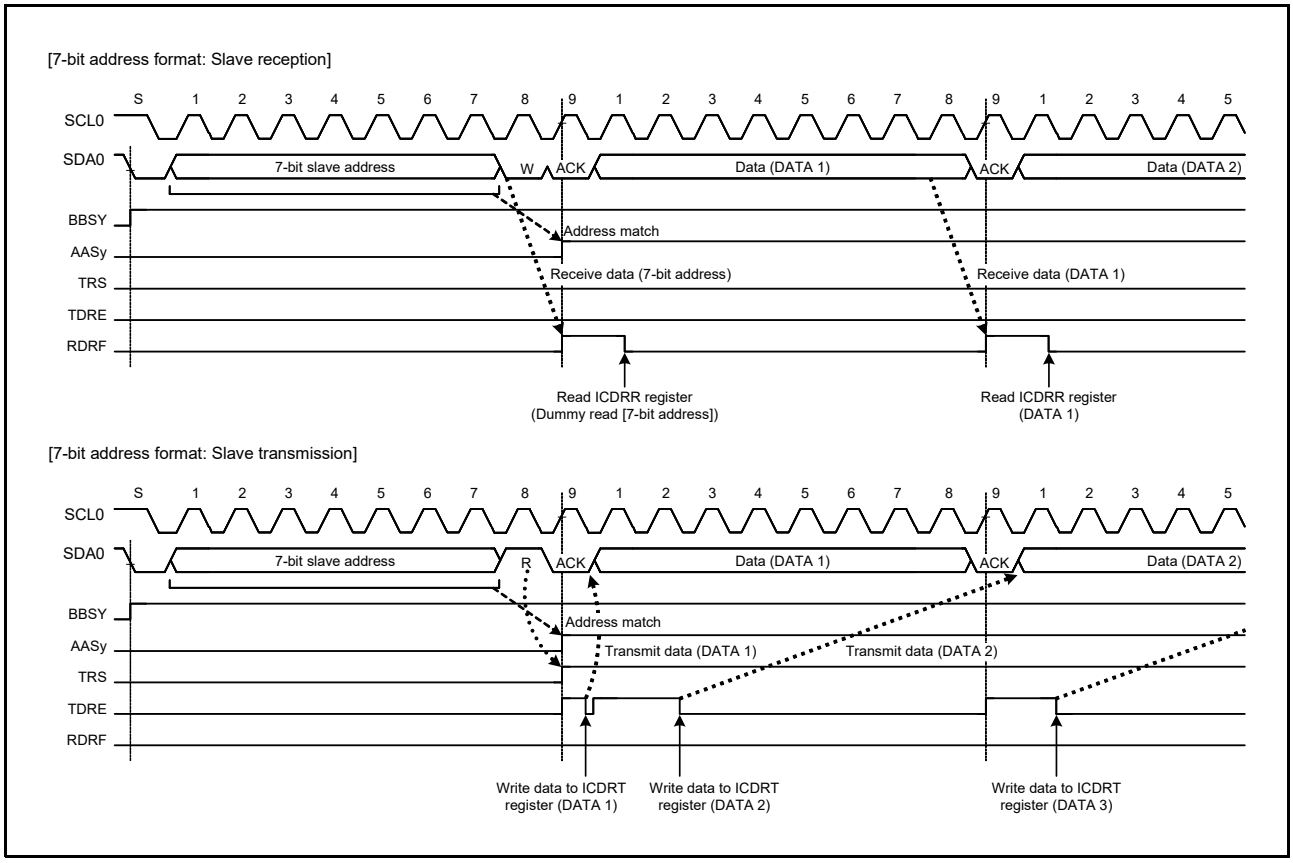


Figure 34.23 AASy Flag Set Timing with 7-Bit Address Format Selected

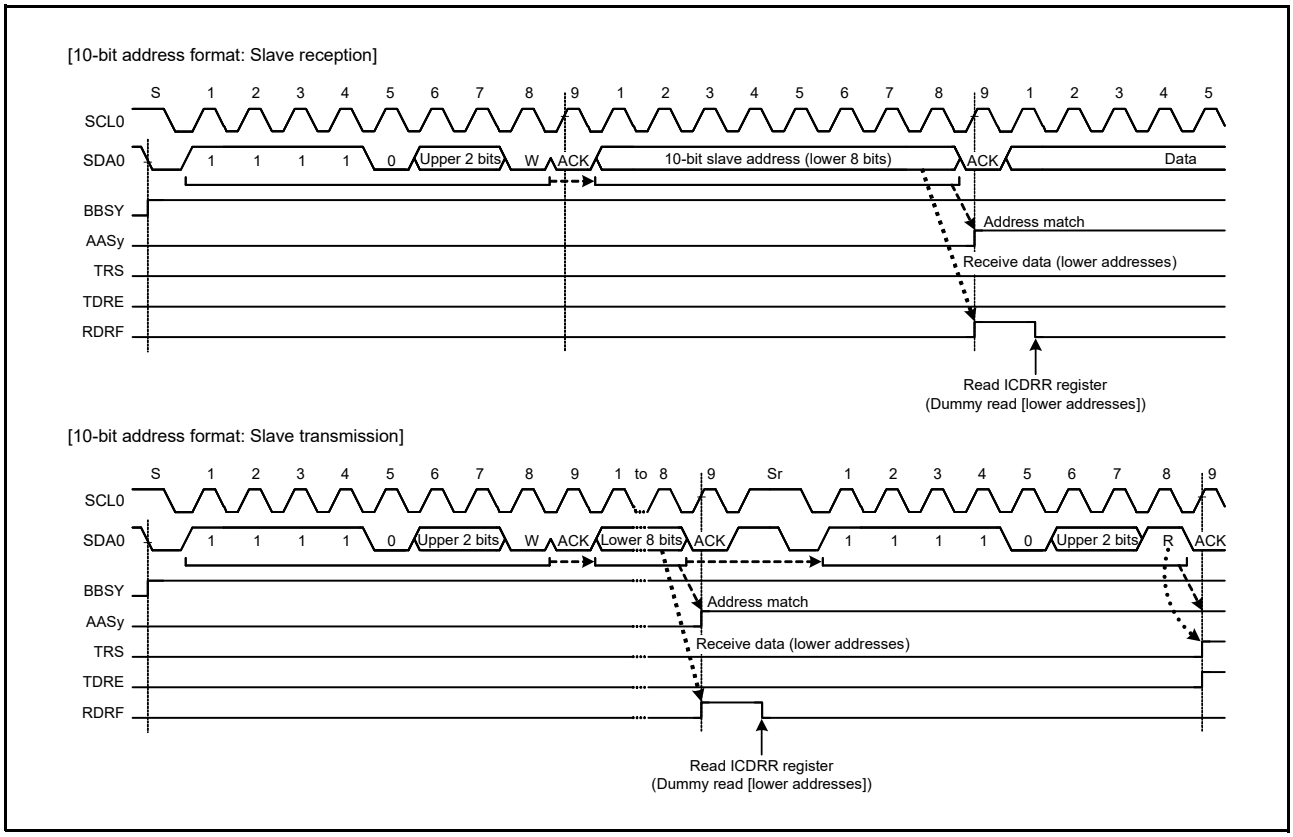


Figure 34.24 AASy Flag Set Timing with 10-Bit Address Format Selected

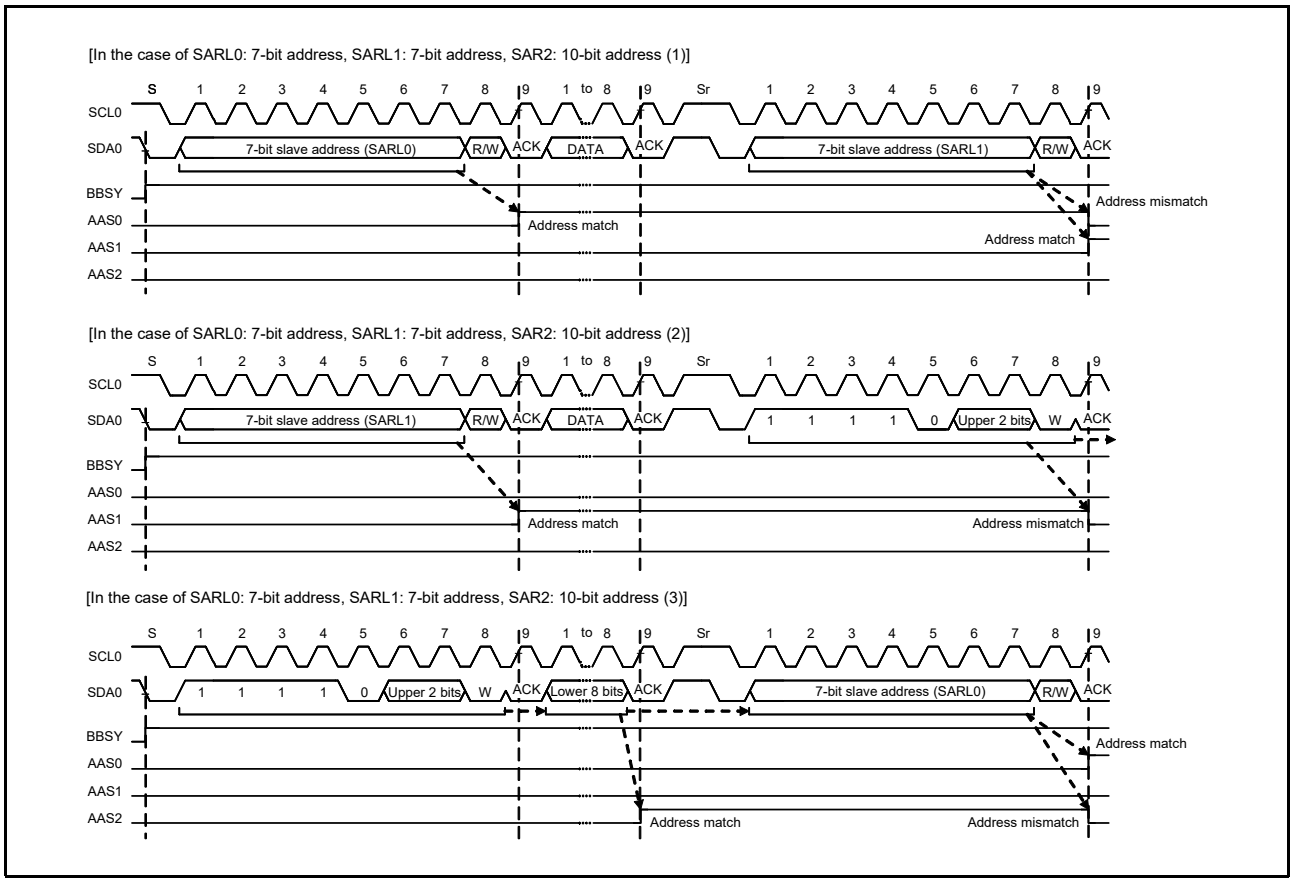


Figure 34.25 AASy Flag Set/Clear Timing with 7-Bit/10-Bit Address Formats Mixed

34.7.2 Detection of the General Call Address

The RIIC also has a facility for detecting the general call address (0000 000b + 0 (write)). This is enabled by setting the ICSER.GCAE bit to 1.

If the address following a start or restart condition is 0000 000b + 1 (read) (start byte), the RIIC recognizes this as the address of a slave device with an “all-zero” address but not as the general call address.

When the RIIC detects the general call address, both the ICSR1.GCA flag and the ICSR2.RDRF flag are set to 1 on the rising edge of the ninth SCL. This leads to the generation of a receive data full interrupt (RXI). The value of the GCA flag can be confirmed to recognize that the general call address has been transmitted.

Operation after detection of the general call address is the same as normal slave receive operation.

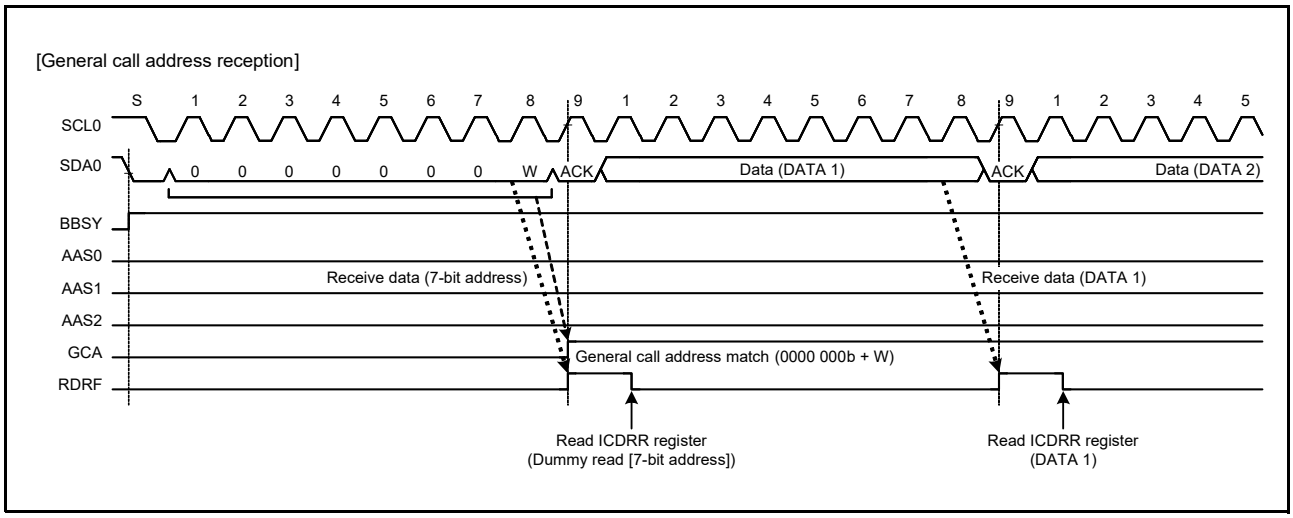


Figure 34.26 Timing of GCA Flag Setting during Reception of General Call Address

34.7.3 Device-ID Address Detection

The RIIC module has a function to detect device-ID addresses complying with the I2C-bus specification. When the RIIC receives 1111 100b as the first seven bits of the first byte following a start condition or a restart condition while the ICSER.DIDE bit is 1, the RIIC recognizes the address as a device-ID address, sets the ICSR1.DID flag to 1 on the rising edge of the ninth SCL when the following R/W# bit is 0, and then compares the second and following bytes with its own slave address. If the received address matches the value in the slave address register, the RIIC sets the corresponding ICSR1.AASy flag (y = 0 to 2) to 1.

After that, when the first byte received after a start or restart condition is generated matches the device ID address (1111 100b) again and the following R/W# bit is 1, the RIIC does not compare the second and subsequent bytes and sets the ICSR2.TDRE flag to 1.

In the device-ID address detection function, the RIIC sets the DID flag to 0 if a match with the RIIC's own slave address is not obtained or a match with the device ID address is not obtained after a match with the RIIC's own slave address and the detection of a restart condition. If the first byte after detection of a start or restart condition matches the device ID address (1111 100b) and the R/W# bit is 0, the RIIC sets the DID flag to 1 and compares the second and subsequent bytes with the RIIC's slave address. If the R/W# bit is 1, the DID flag holds the previous value and the RIIC does not compare the second and subsequent bytes. Therefore, the reception of a device-ID address can be checked by reading the DID flag after confirming that TDRE flag is 1.

Furthermore, prepare the device-ID fields (3 bytes: 12 bits indicating the manufacturer + 9 bits identifying the part + 3 bits indicating the revision) that must be sent to the host after reception of a continuous device-ID field as normal data for transmission. For details of the information that must be included in device-ID fields, contact NXP Semiconductors.

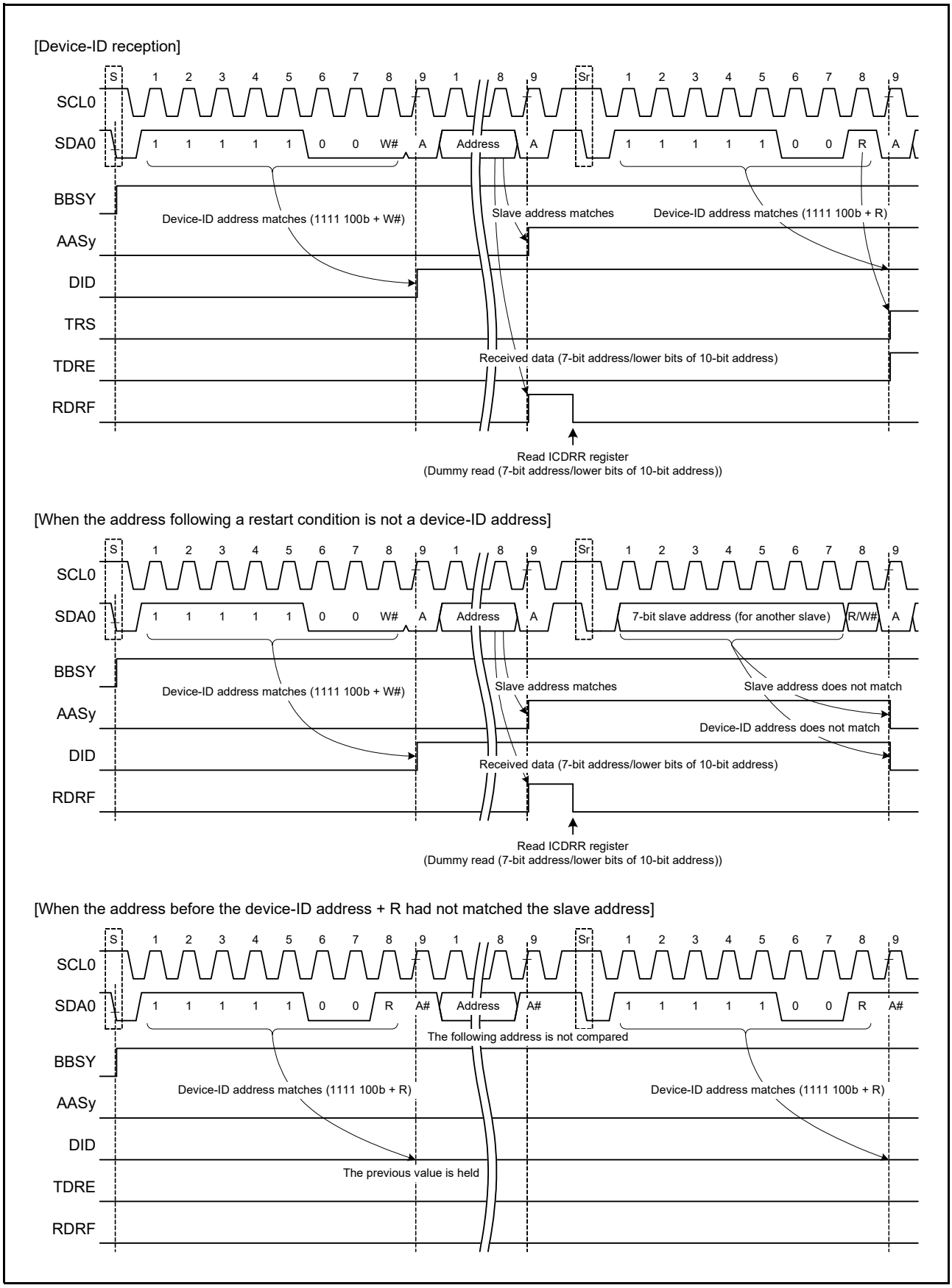


Figure 34.27 Set/Clear Timing of the AASy and DID Flags during Reception of Device-ID Address

34.7.4 Host Address Detection

The RIIC has a function to detect the host address while the SMBus is operating. When the ICSER.HOAE bit is set to 1 while the ICMR3.SMBS bit is 1, the RIIC can detect the host address (0001 000b) in slave receive mode (bits MST and TRS in the ICCR2 register are 00b).

When the RIIC detects the host address, the ICSR1.HOA flag is set to 1 at the rising edge of the ninth SCL, and at the same time, the ICSR2.RDRF flag is set to 1 when the R/W# bit is 0 (Wr bit). This causes a receive data full interrupt (RXI) to be generated. The HOA flag is used to recognize that the host address was sent from the smart battery or other devices.

If the bit following the host address (0001 000b) is an Rd bit (R/W# bit is 1), the RIIC can also detect the host address. After the host address is detected, the RIIC operates in the same manner as normal slave operation.

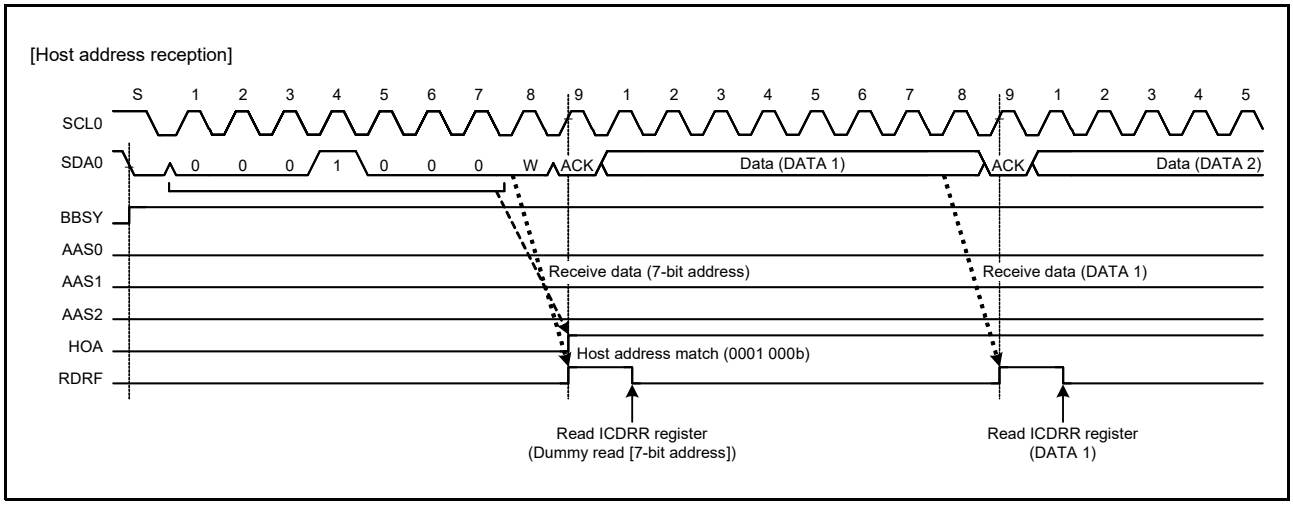


Figure 34.28 HOA Flag Set Timing during Reception of Host Address

34.8 Automatic Low-Hold Function for SCL

34.8.1 Function to Prevent Wrong Transmission of Transmit Data

If the shift register (ICDRS) is empty when data have not been written to the I2C-bus transmit data register (ICDRT) with the RIIC in transmission mode (ICCR2.TRS bit is 1), the SCL0 line is automatically held low over the intervals shown below. This low period is extended until data for transmission have been written, which prevents the unintended transmission of erroneous data.

Master transmit mode

- Low period after a start condition or restart condition is generated
- Low period between the ninth clock pulse of one transfer and the first clock pulse of the next

Slave transmit mode

- Low period between the ninth clock pulse of one transfer and the first clock pulse of the next

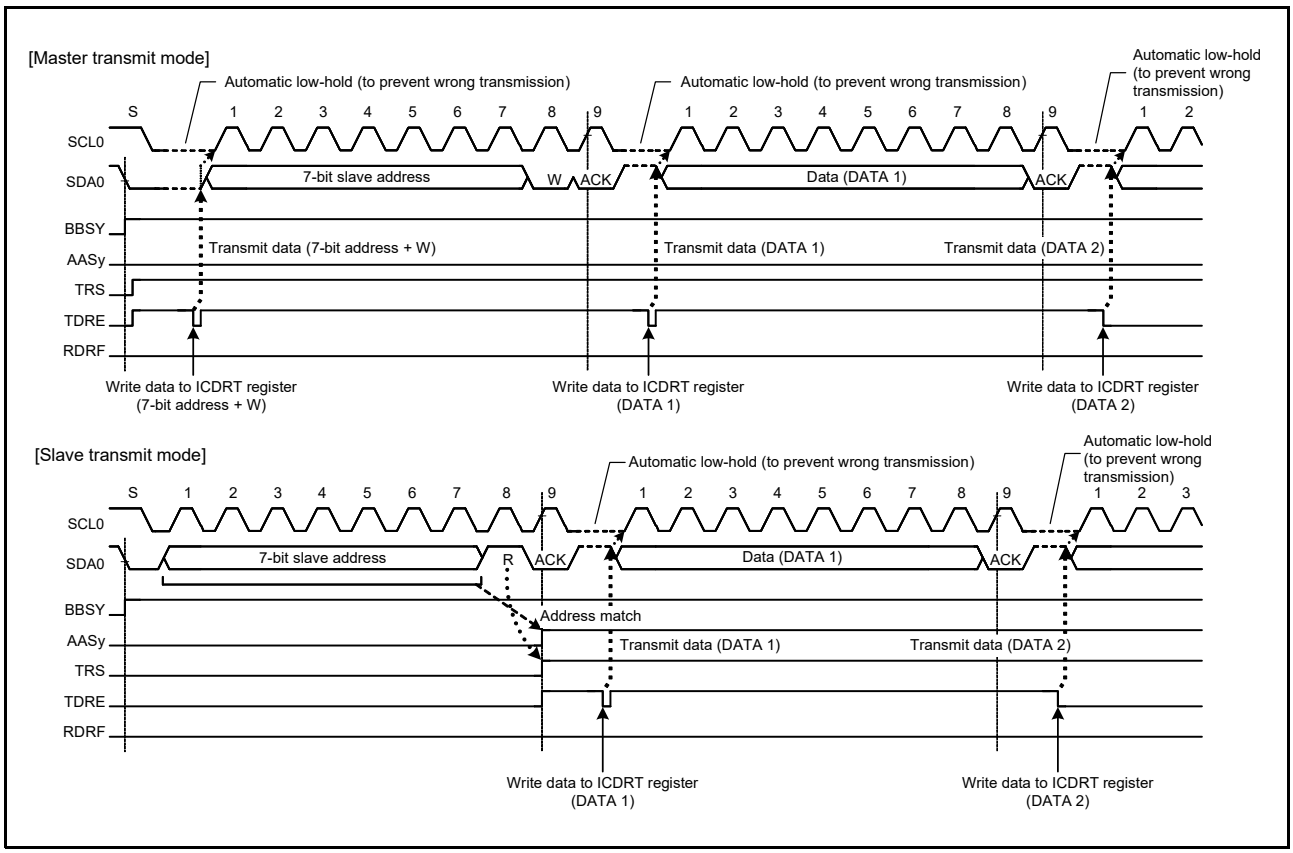


Figure 34.29 Automatic Low-Hold Operation in Transmit Mode

34.8.2 NACK Reception Transfer Suspension Function

The RIIC has a function to suspend transfer operation when NACK is received in transmit mode (ICCR2.TRS bit is 1). This function is enabled when the ICFER.NACKE bit is set to 1 (transfer suspension enabled). If the next transmit data has already been written (ICSR2.TDRE flag is 0) when NACK is received, next data transmission at the falling edge of the ninth SCL is automatically suspended. This prevents the SDA0 line output level from being held low when the MSB of the next transmit data is 0.

If the data transmission is suspended (ICSR2.NACKF flag is 1) by this function, the following data transmission and data reception are not started. To resume data transfer, set the NACKF flag to 0. In master transmit mode, restart data transfer by setting the NACKF flag to 0 after generating a restart condition, or restart data transfer from a start condition after generating a stop condition then setting the NACKF flag to 0.

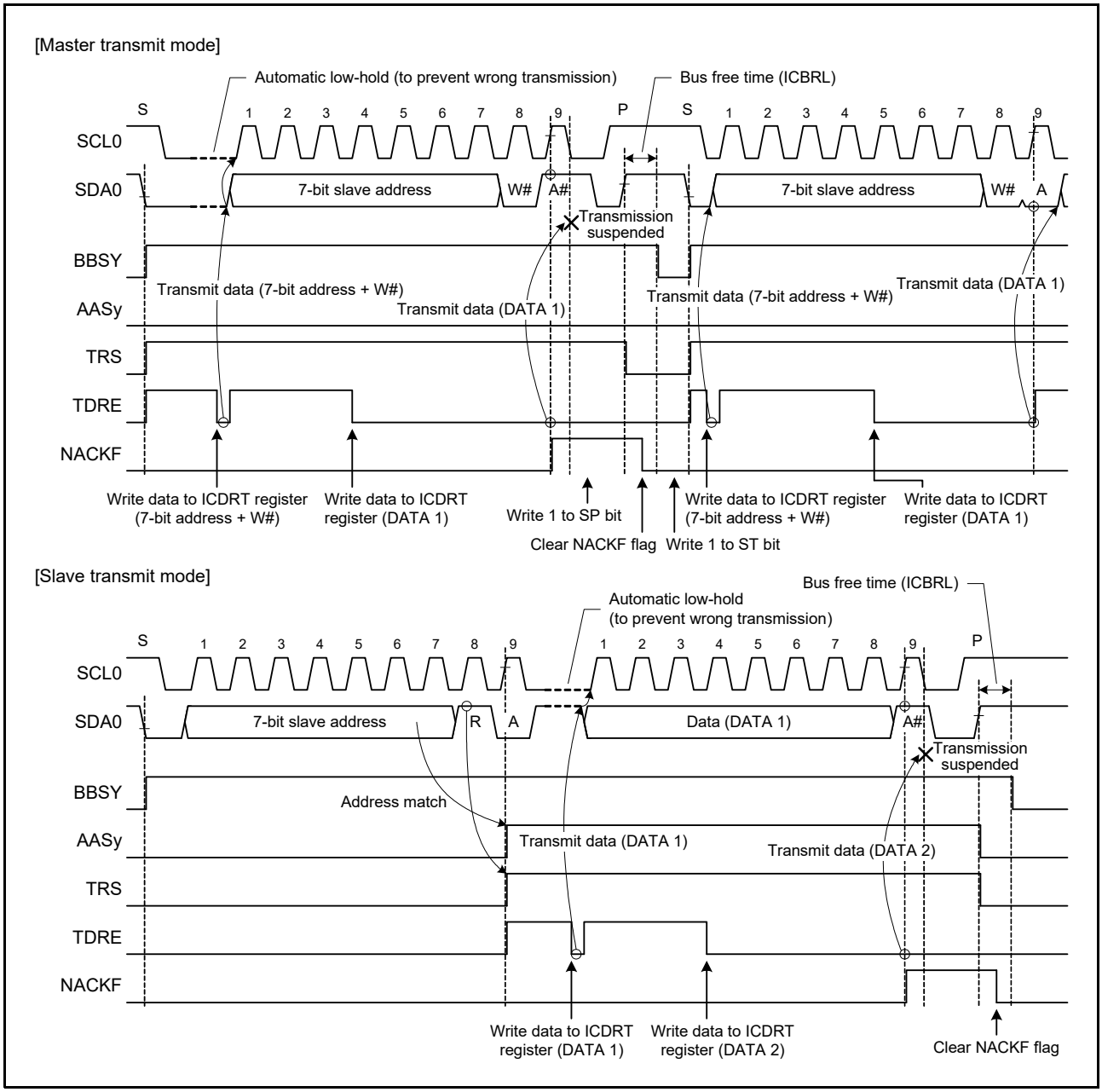


Figure 34.30 Suspension of Data Transmission When NACK is Received (NACKE = 1)

34.8.3 Function to Prevent Failure to Receive Data

If response processing is delayed when receive data (ICDRR) read is delayed for a period of one transfer byte or more with receive data full (ICSR2.RDRF flag is 1) in receive mode (ICCR2.TRS bit is 0), the RIIC holds the SCL0 line low automatically immediately before the next data is received to prevent failure to receive data.
This function to prevent failure to receive data using the automatic low-hold function is also enabled even if the read processing of the final receive data is delayed and, in the meantime, the RIIC's own slave address is designated after a stop condition is generated. This function does not disturb other communication because the RIIC does not hold the SCL0 line low when a mismatch with its own slave address occurs after a stop condition is generated.
Sections in which the SCL0 line is held low can be selected with a combination of the WAIT and RDRFS bits in the ICMR3 register.

(1) 1-Byte Receive Operation and Automatic Low-Hold Function Using the WAIT Bit

When the ICMR3.WAIT bit is set to 1, the RIIC performs 1-byte receive operation using the WAIT bit function. Furthermore, when the ICMR3.RDRFS bit is 0, the RIIC automatically sends the ICMR3.ACKBT bit value for the acknowledgment bit in the period from the falling edge of the eighth SCL to the falling edge of the ninth SCL, and automatically holds the SCL0 line low at the falling edge of the ninth SCL using the WAIT bit function. This low-hold is released by reading data from the ICDRR register, which enables bytewise receive operation.
The WAIT bit function is enabled for receive bytes after a match with the RIIC's own slave address (including the general call address and host address) is obtained in master receive mode or slave receive mode.

(2) 1-Byte Receive Operation (ACK/NACK Transmission Control) and Automatic Low-Hold Function Using the RDRFS Bit

When the ICMR3.RDRFS bit is set to 1, the RIIC performs 1-byte receive operation using the RDRFS bit function. When the RDRFS bit is set to 1, the ICSR2.RDRF flag (receive data full) is set to 1 at the rising edge of the eighth SCL, and the SCL0 line is automatically held low at the falling edge of the eighth SCL. This low-hold is released by writing a value to the ICMR3.ACKBT bit, but cannot be released by reading data from the ICDRR register, which enables receive operation by the ACK/NACK transmission control according to the data received in byte units.
The RDRFS bit function is enabled for receive bytes after a match with the RIIC's own slave address (including the general call address and host address) is obtained in master receive mode or slave receive mode.

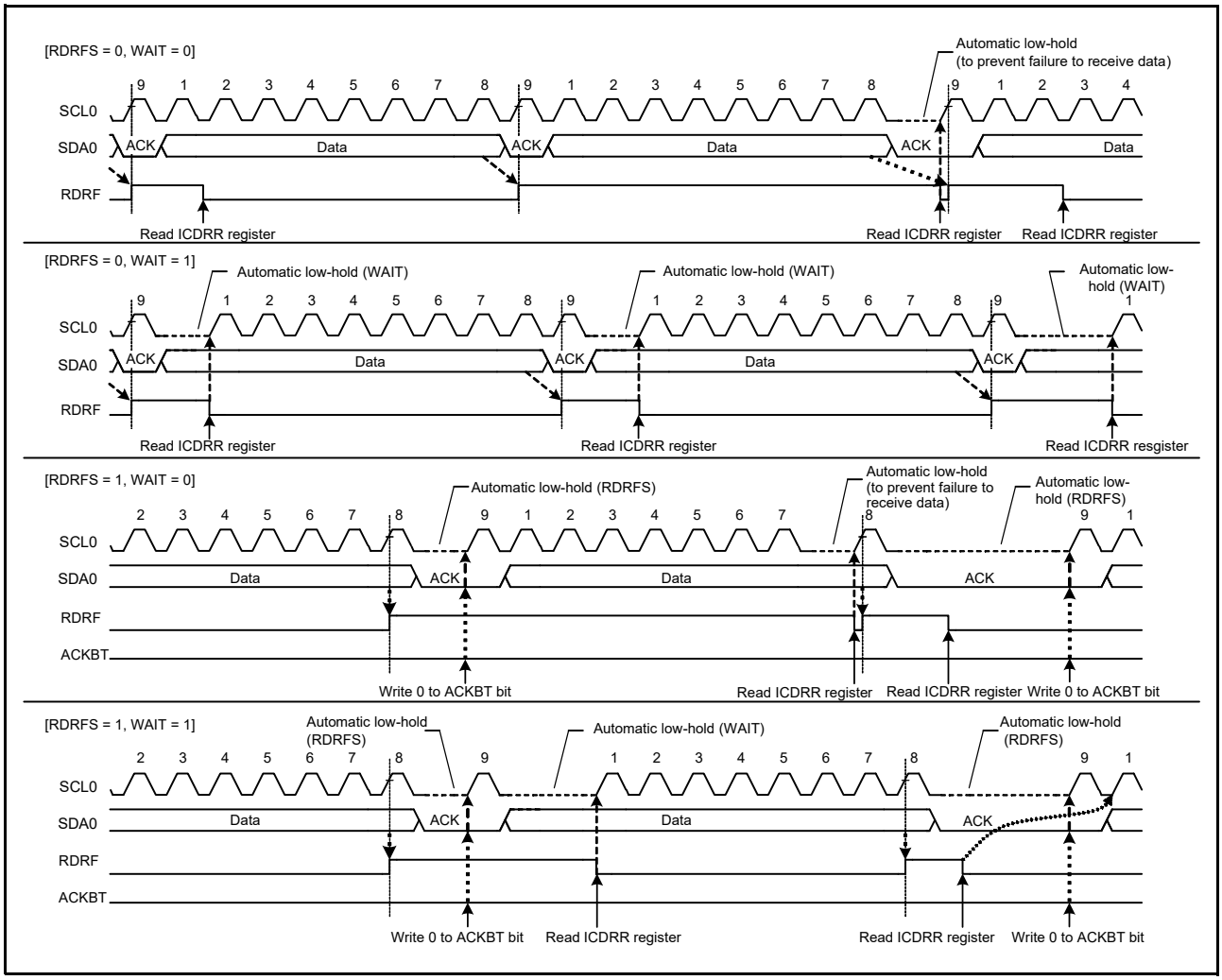


Figure 34.31 Automatic Low-Hold Operation in Receive Mode (Using RDRFS and WAIT Bits)

34.9 Arbitration-Lost Detection Functions

In addition to the normal arbitration-lost detection function defined by the I2C-bus specification, the RIIC has functions to prevent double-generation of a start condition, to detect arbitration-lost during transmission of NACK, and to detect arbitration-lost in slave transmit mode.

34.9.1 Master Arbitration-Lost Detection (MALE Bit)

The RIIC drives the SDA0 line low to generate a start condition. However, if the SDA0 line has already been driven low by another master device generating a start condition, the RIIC causes arbitration to be lost, so priority is given to transfer by the other master device. Similarly, if the ICCR2.ST bit is set to 1 while the ICCR2.BBSY flag is 1 (bus busy state), arbitration is lost, so priority is given to transfer by the other master device. No start condition is generated in this case.
When a start condition is generated successfully, if the data for transmission including the address bits (i.e. the internal SDA output level) and the level on the SDA0 line do not match (the high output as the internal SDA output; i.e. the SDA0 pin is in the high-impedance state) and the low is detected on the SDA0 line, the RIIC loses the arbitration.
After a master arbitration-lost is generated, the RIIC immediately enters slave receive mode. If a slave address (including the general call address) matches its own address at this time, the RIIC continues in slave operation.
A master arbitration-lost is detected when the following conditions are met while the ICFER.MALE bit is 1 (master arbitration-lost detection enabled).

Conditions for detecting master arbitration-lost

- Non-matching of the internal level for output on SDA and the level on the SDA0 line after a start condition was generated by setting the ICCR2.ST bit to 1 while the ICCR2.BBSY flag was set to 0 (erroneous generation of a start condition)
- Setting of the ICCR2.ST bit to 1 while the BBSY flag is set to 1 (start condition double-generation error)
- When the transmit data excluding acknowledgment bit (internal SDA output level) does not match the level on the SDA0 line in master transmit mode (bits MST and TRS in the ICCR2 register = 11b)

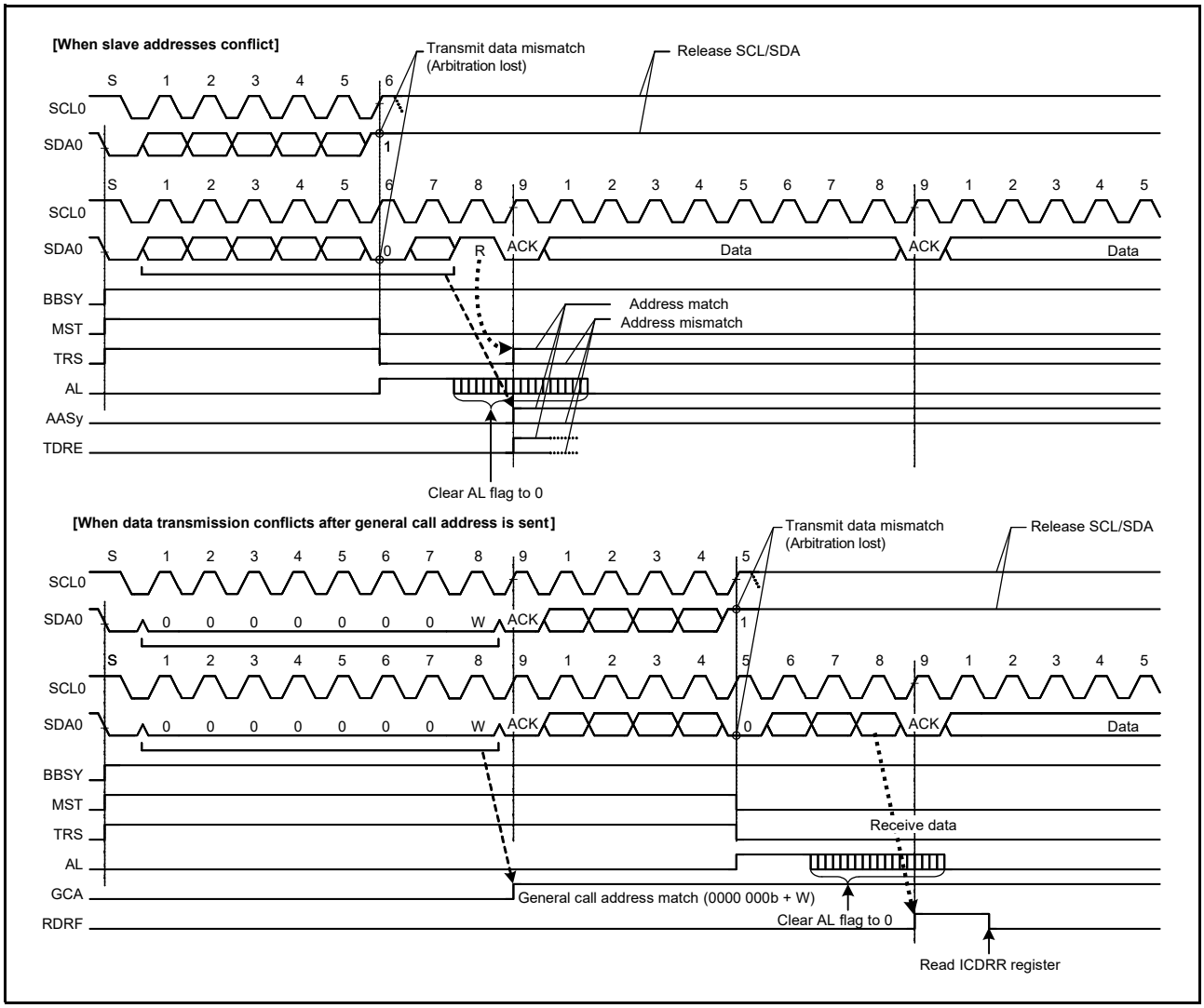


Figure 34.32 Examples of Master Arbitration-Lost Detection (MALE = 1)

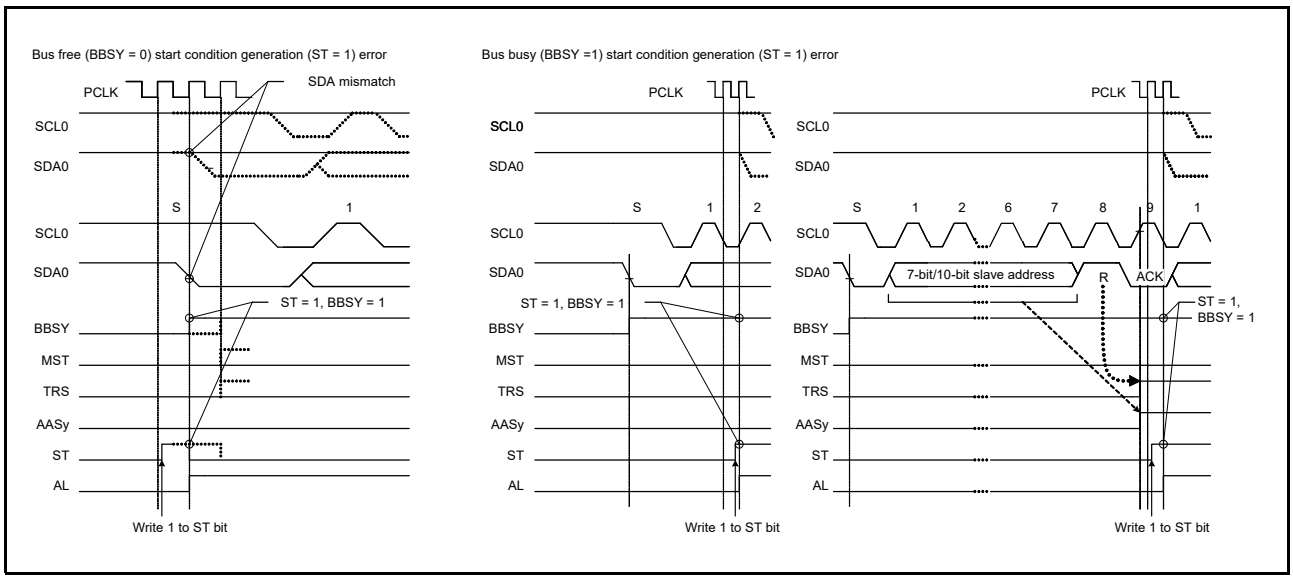


Figure 34.33 Arbitration-Lost When a Start Condition is Generated (MALE = 1)

34.9.2 NACK Transmission Arbitration-Lost Detection Function (NALE Bit)

The RIIC has a function to cause arbitration to be lost if the internal SDA output level does not match the level on the SDA0 line (the high output as the internal SDA output; i.e. the SDA0 pin is in the high-impedance state) and the low is detected on the SDA0 line during transmission of NACK in receive mode. Arbitration is lost due to a conflict of NACK transmission and ACK transmission when two or more master devices receive data from the same slave device simultaneously in a multi-master system. Such conflict occurs when multiple master devices send/receive the same information through a single slave device. Figure 34.34 shows an example of NACK transmission arbitration-lost detection.

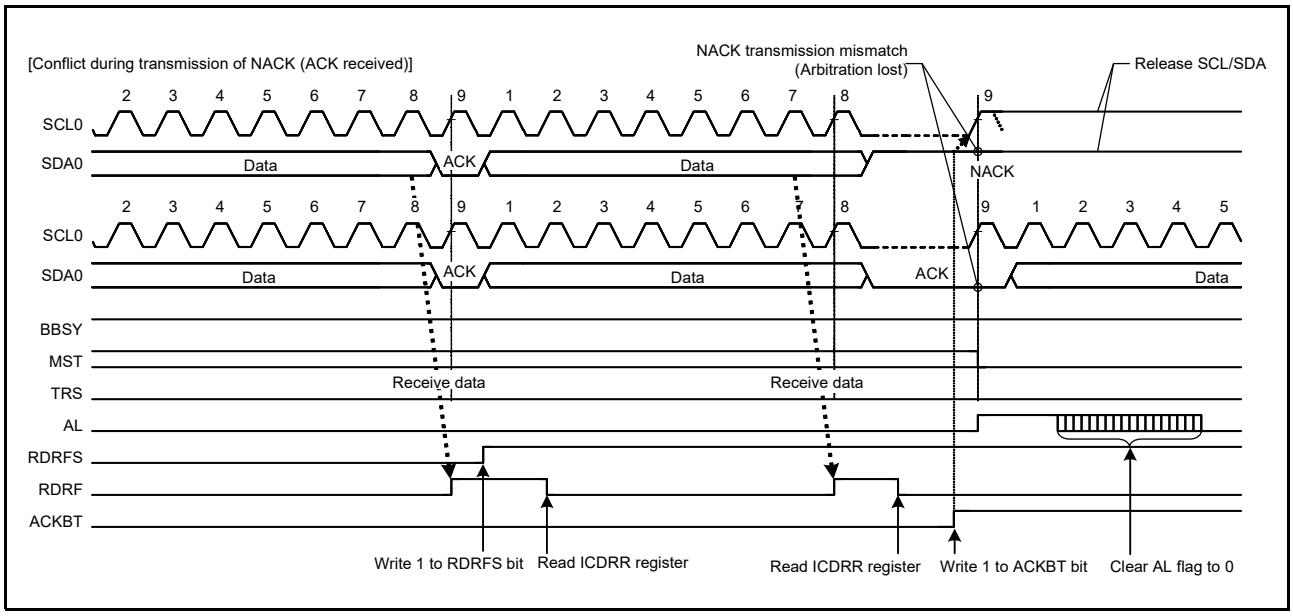


Figure 34.34 Example of NACK transmission arbitration-lost Detection (NALE = 1)

The following explains arbitration-lost detection using an example where two master devices (master A and master B) and a single slave device are connected through the bus. In this example, master A receives two bytes of data from the slave device, and master B receives four bytes of data from the slave device.

If master A and master B access the slave device simultaneously, because the slave address is identical, arbitration is not lost in both master A and master B during access to the slave device. Therefore, both master A and master B recognize that they have obtained the bus mastership and operate as such. Here, master A sends NACK when it has received second byte of data from the slave device. Meanwhile, master B sends ACK because it has not received necessary four bytes of data. At this time, the NACK transmission from master A and the ACK transmission from master B conflict. In general, if a conflict like this occurs, master A cannot detect ACK transmitted by master B and generates a stop condition. Therefore, the generation of the stop condition conflicts with the SCL output of master B, which disturbs communication.

When the RIIC receives ACK during transmission of NACK, it detects a defeat in conflict with other master devices and causes arbitration to be lost.

If a NACK transmission arbitration-lost occurs, the RIIC immediately cancels the slave address matched state and enters slave receive mode. This prevents a stop condition from being generated, preventing a communication failure on the bus. Also, in the ARP command processing of SMBus, it is possible to omit surplus processing (FFh transmission processing) after NACK transmission when the UDID (Unique Device Identifier) of “Assign Address” does not match and after the NACK transmission of the “Get UDID (General)” after the “Assign Address” is confirmed.

The RIIC detects NACK transmission arbitration-lost when the following condition is met with the ICFER.NALE bit set to 1 (NACK transmission arbitration-lost is enabled).

Condition for detecting NACK transmission arbitration-lost

- When the internal SDA output level does not match the SDA0 line (ACK is received) during transmission of NACK (ICMR3.ACKBT bit = 1)

34.9.3 Slave Arbitration-Lost Detection (SALE Bit)

The RIIC has a function to cause arbitration to be lost if the data for transmission (i.e. the internal SDA output level) and the level on the SDA0 line do not match (the high output as the internal SDA output; i.e. the SDA0 pin is in the high-impedance state and the low is detected on the SDA0 line in slave transmit mode). The slave arbitration-lost detection function is mainly used when transmitting a UDID (Unique Device Identifier) of SMBus.
When the slave arbitration-lost occurs, the RIIC is immediately released from the slave address matched state and enters slave receive mode. This function can detect a data conflict during UDID transmission of SMBus and omit surplus processing (FFh transmission processing) after the data conflict.
The RIIC detects slave arbitration-lost when the following condition is met with the ICFER.SALE bit set to 1 (slave arbitration-lost detection enabled).

Condition for detecting slave arbitration-lost

- When transmit data excluding acknowledgment bit (internal SDA output level) does not match the SDA0 line in slave transmit mode (bits MST and TRS in the ICCR2 register are 01b)

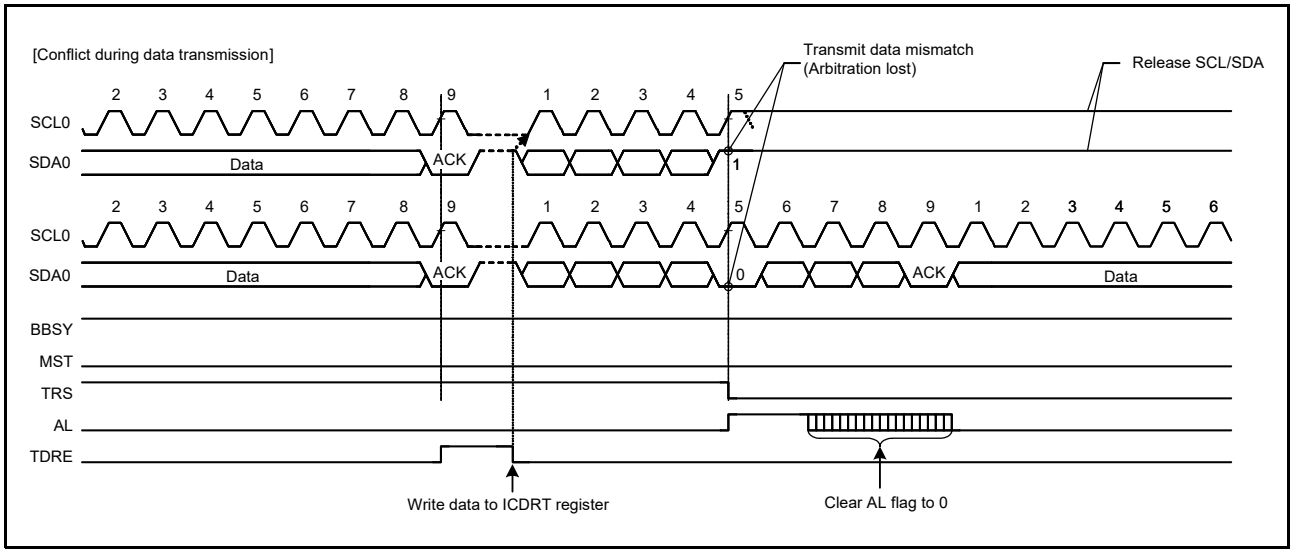


Figure 34.35 Example of Slave Arbitration-Lost Detection (SALE = 1)

34.10 Start Condition/Restart Condition/Stop Condition Generating Function

34.10.1 Generating a Start Condition

The RIIC generates a start condition when the ICCR2.ST bit is set to 1.
When the ST bit is set to 1, a start condition generation request is made and the RIIC generates a start condition when the ICCR2.BBSY flag is 0 (bus free state). When a start condition is generated normally, the RIIC automatically shifts to the master transmit mode.
A start condition is generated in the following sequence.

Start condition generation

(1) Drive the SDA0 line low (high to low).
(2) Secure the time set in the ICBRH register and the start condition hold time.
(3) Drive the SCL0 line low (high to low).
(4) Detect the low level on the SCL0 line and secure the low period of the signal on the SCL0 line set in the ICBRL register.

34.10.2 Generating a Restart Condition

The RIIC generates a restart condition when the ICCR2.RS bit is set to 1.
When the RS bit is set to 1, a restart condition generation request is made and the RIIC generates a restart condition when the ICCR2.BBSY flag is 1 (bus busy state) and the ICCR2.MST bit is 1 (master mode).
A restart condition is generated in the following sequence.

Restart condition generation

(1) Release the SDA0 line.
(2) Secure the low period of the signal on the SCL0 line set in the ICBRL register.
(3) Release the SCL0 line (low to high).
(4) Detect the high level on the SCL0 line and secure the time set in the ICBRL register and the restart condition setup time.
(5) Drive the SDA0 line low (high to low).
(6) Secure the time set in the ICBRH register and the restart condition hold time.
(7) Drive the SCL0 line low (high to low).
(8) Detect the low level on the SCL0 line and secure the low period of the signal on the SCL0 line set in the ICBRL register.

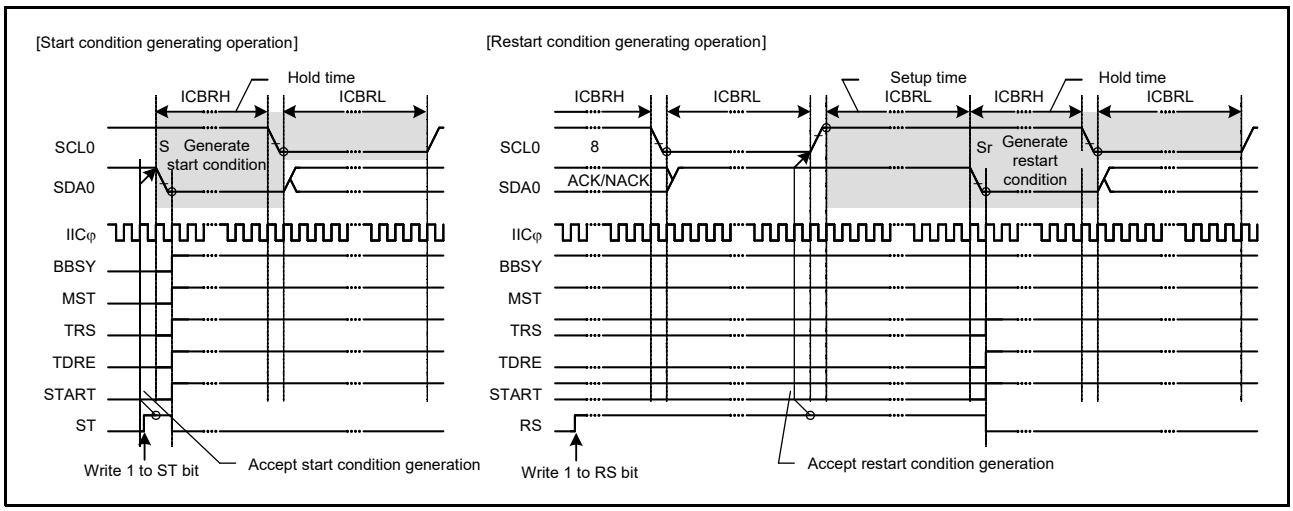


Figure 34.36 Start Condition/Restart Condition Generation Timing (ST and RS Bits)

34.10.3 Generating a Stop Condition

The RIIC generates a stop condition when the ICCR2.SP bit is set to 1.
When the SP bit is set to 1, a stop condition generation request is made and the RIIC generates a stop condition when the ICCR2.BBSY flag is 1 (bus busy state) and the ICCR2.MST bit is 1 (master mode).
A stop condition is generated in the following sequence.

Stop condition generation

(1) Drive the SDA0 line low (high to low).
(2) Secure the low period of the signal on the SCL0 line set in the ICBRL register.
(3) Release the SCL0 line (low to high).
(4) Detect the high level on the SCL0 line and secure the time set in the ICBRH register and the stop condition setup time.
(5) Release the SDA0 line (low to high).
(6) Secure the time set in the ICBRL register and the bus free time.
(7) Set the BBSY flag to 0 (to release the bus mastership).

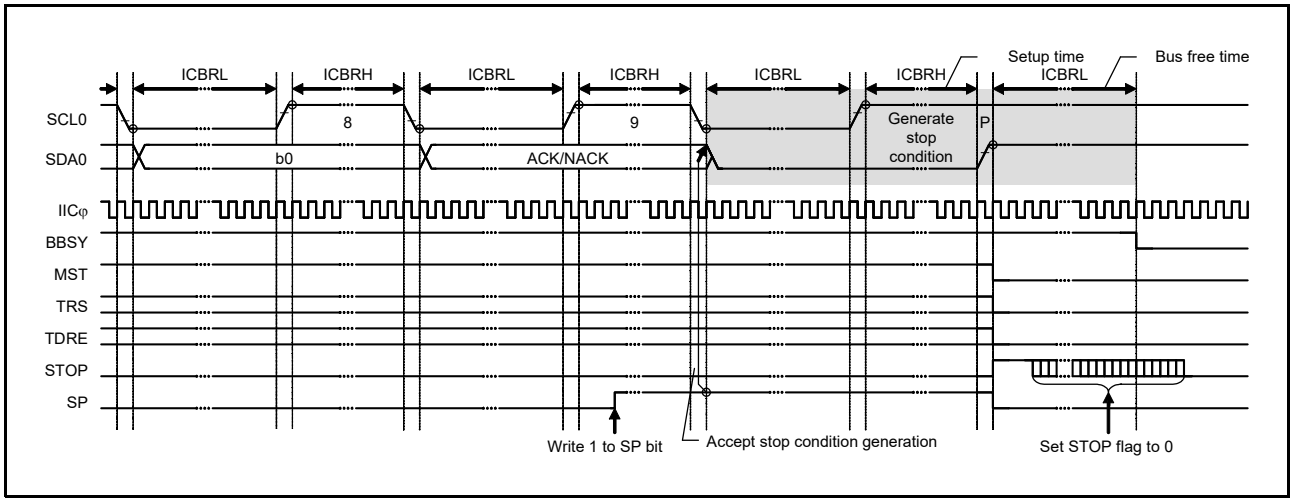


Figure 34.37 Stop Condition Generation Timing (SP Bit)

34.11 Bus Hanging

If the clock signals from the master and slave devices go out of synchronization due to noise or other factors, the I2C-bus might hang with a fixed level on the SCL0 line and/or SDA0 line.
As measures against the bus hanging, the RIIC has a timeout function to detect hanging by monitoring the SCL0 line, a function for the output of an additional SCL to release the bus from a hung state due to clock signals being out of synchronization, the RIIC reset function, and internal reset function.
By checking bits SCLO, SDAO, SCLI, and SDAI in the ICCR1 register, it is possible to see whether the RIIC or its partner in communications is placing the low level on the SCL0 or SDA0 lines.

34.11.1 Timeout Function

The RIIC includes a timeout function for detecting when the SCL0 line has been stuck longer than the predetermined time. The RIIC can detect an abnormal bus state by monitoring that the SCL0 line is stuck low or high for a predetermined time.
The timeout function monitors the SCL0 line state and counts the low period or high period using the internal counter. The timeout function resets the internal counter each time the SCL0 line changes (rising or falling), but continues to count unless the SCL0 line changes. If the internal counter overflows due to no SCL0 line change, the RIIC can detect the timeout and report the bus hung state.
This timeout function is enabled when the ICFER.TMOE bit is 1. It detects a hung state that the SCL0 line is stuck low or high during the following conditions:

- The bus is busy (ICCR2.BBSY flag is 1) in master mode (ICCR2.MST bit is 1).
- The RIIC's own slave address is detected (ICSR1 register is not 00h) and the bus is busy (ICCR2.BBSY flag is 1) in slave mode (ICCR2.MST bit is 0).
- The bus is free (ICCR2.BBSY flag is 0) while generation of a start condition is requested (ICCR2.ST bit is 1).

The internal counter of the timeout function works using the internal reference clock (IICφ) set by the ICMR1.CKS[2:0] bits as a count source. It functions as a 16-bit counter when long mode is selected (ICMR2.TMOS bit is 0) or a 14-bit counter when short mode is selected (TMOS bit is 1).
The SCL0 line level (low/high or both levels) during which this counter is activated can be selected by the setting of bits TMOH and TMOL in the ICMR2 register. If both bits TMOL and TMOH are set to 0, the internal counter does not work.

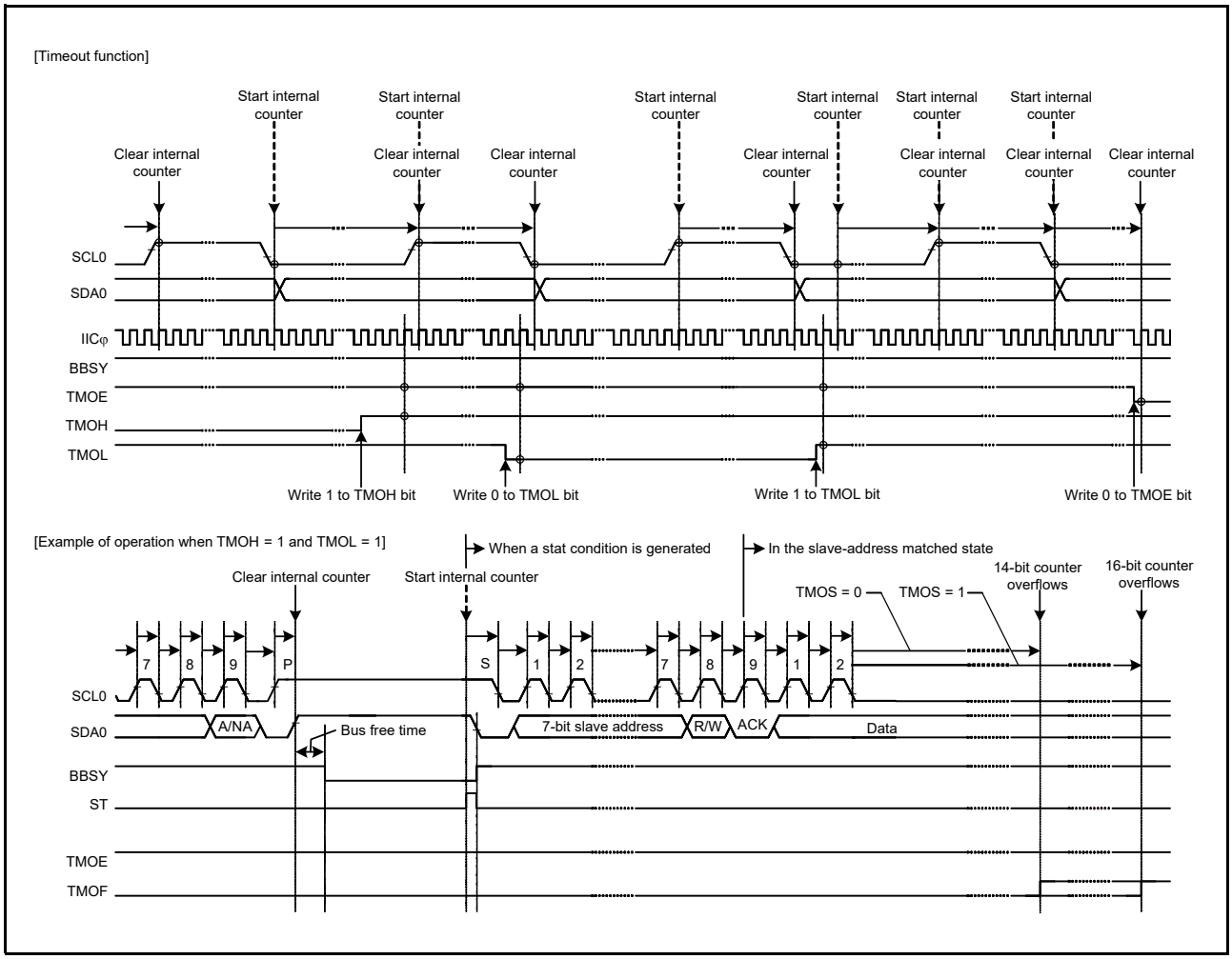


Figure 34.38 Timeout Function

34.11.2 Additional SCL Output Function

In master mode, the RIIC module has a facility for the output of additional SCL to release the SDA0 line from being held low by the slave device due to the master being out of synchronization with the slave device.
This function is mainly used in master mode to release the SDA0 line from the state of being stuck low by including additional SCL output from the RIIC with single cycles of the SCL as the unit if the RIIC cannot generate a stop condition because the slave device is holding the SDA0 line low. Do not use this function in normal situations. Using it when communications are proceeding correctly will lead to malfunctions.
When the ICCR1.CLO bit is set to 1, an additional clock pulse at the frequency set by the ICMR1.CKS[2:0] bits and the ICBRH and ICBRL registers is output from the SCL0 pin. After output of this clock pulse, the CLO bit automatically becomes 0. The SCL0 pin is held low when the ICCR2.BBSY flag is 1 and held high when the BBSY flag is 0.
Consecutive additional clock pulses can be output by writing 1 to the CLO bit after confirming the CLO bit to be 0.
When the RIIC module is in master mode and the slave device is holding the SDA0 line low because synchronization with the slave device has been lost due to the effects of noise, etc., the output of a stop condition is not possible. The additional SCL output function can be used to output additional clock pulses one by one to make the slave device release the SDA0 line from being held low, thus recovering the bus from an unusable state. Release of the SDA0 line by the slave device can be monitored by reading the ICCR1.SDAI bit. After confirming release of the SDA0 line by the slave device, complete communications by regenerating a stop condition.
Use this function with the ICFER.MALE bit set to 0 (master arbitration-lost detection disabled).

Conditions for using the ICCR1.CLO bit

- When the bus is free (ICCR2.BBSY flag is 0) or in master mode (ICCR2.MST bit is 1 and ICCR2.BBSY flag is 1)
- When the communication device does not hold the SCL0 line low

Figure 34.39 shows the operation timing of the additional SCL output function (CLO bit).

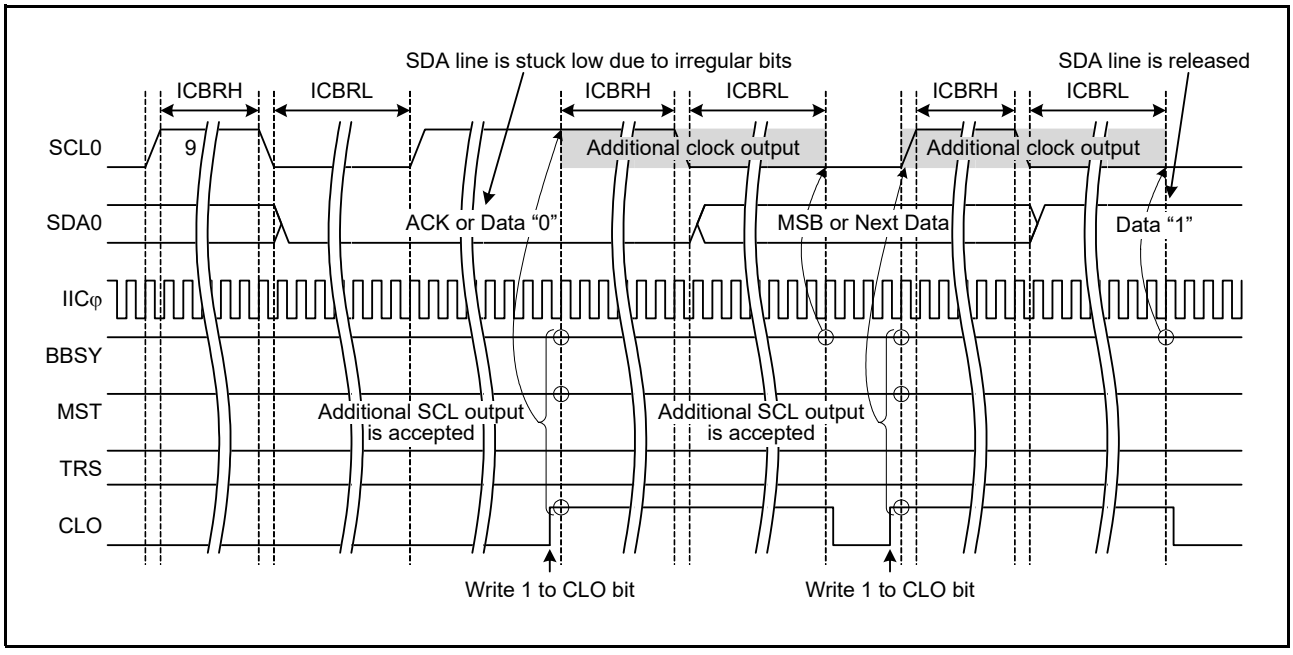


Figure 34.39 Additional SCL Output Function (CLO Bit)

34.11.3 RIIC Reset and Internal Reset

The RIIC module incorporates a function for resetting itself. There are two types of reset. One is referred to as an RIIC reset; this initializes all registers including the ICCR2.BBSY flag. The other is referred to as an internal reset; this releases the RIIC from the slave-address matched state and initializes the internal counter while retaining other settings. After applying a reset, be sure to set the ICCR1.IICRST bit to 0.

Both types of reset are effective for release from bus-hung states because both restore the output state of the SCL0 and SDA0 pins to the high-impedance state.

Applying a reset during slave operation may lead to a loss of synchronization between the master device clock and the slave device clock, so avoided this where possible. Note that monitoring of the bus state, such as for the presence of a start condition, is not possible during an RIIC reset (bits ICE and IICRST in the ICCR1 register are 01b).

For a detailed description of the RIIC and internal resets, refer to section 34.14, Initialization of Registers and Functions When a Reset is Applied or a Condition is Detected.

34.12 SMBus Operation

The RIIC is available for data communication conforming to the SMBus (Version 2.0). To perform SMBus communication, set the ICMR3.SMBS bit to 1. To use the transfer rate within a range of 10 kbps to 100 kbps of the SMBus specification, set the ICMR1.CKS[2:0] bits, the ICBRH register, and the ICBRL register. In addition, determine the values of the ICMR2.DLCS bit and the ICMR2.SDDL[2:0] bits to meet the data hold time (300 ns (min.)). If the RIIC is used only as a slave device, the transfer rate setting is not necessary, whereas the ICBRL register needs to be set to a value longer than the data setup time (250 ns).
For the SMBus device default address (1100 001b), use one of the slave address registers L0 to L2 (registers SARL0, SARL1, and SARL2), and set the corresponding SARUy.FS bit (7-bit/10-bit address format select) (y = 0 to 2) to 0 (7-bit address format).
When transmitting the UDID (Unique Device Identifier), set the ICFER.SALE bit to 1 to enable the slave arbitration-lost detection function.

34.12.1 SMBus Timeout Measurement

(1) Measuring timeout of slave device

The following period (timeout interval: $T_{LOW:SEXT}$) must be measured for slave devices in SMBus communication.

- From start condition to stop condition

To measure timeout for slave devices, measure the period from start condition detection to stop condition detection with the MTU or TMR timer using a start condition detection interrupt (STI) and stop condition detection interrupt (SPI) of the RIIC. The measured timeout period must be within the cumulative clock low extend time (slave device) ($T_{LOW:SEXT}$: 25 ms (max.)) of the SMBus specification.
If the time measured with the MTU or TMR exceeds the detect clock low timeout ($T_{TIMEOUT}$: 25 ms (min.)) of the SMBus specification, the slave device must release the bus by writing 1 to the ICCR1.IICRST bit to apply an internal reset of the RIIC. When an internal reset is applied to the RIIC, it stops driving the SCL0 and SDA0 pins of the bus and makes the SCL0/SDA0 pin outputs high-impedance, thus releasing the bus.

(2) Measuring timeout of master device

The following periods (timeout interval: $T_{LOW:MEXT}$) must be measured for master devices in SMBus communication.

- From start condition to acknowledgment bit
- Between acknowledgment bits
- From acknowledgment bit to stop condition

To measure timeout for master devices, measure these periods with the MTU or TMR timer using a start condition detection interrupt (STI), stop condition detection interrupt (SPI), and transmission end interrupt (TEI) or receive data full interrupt (RXI) of the RIIC. The measured timeout period must be within the cumulative clock low extend time (master device) ($T_{LOW:MEXT}$: 10 ms (max.)) of the SMBus specification, and the total of all $T_{LOW:MEXT}$ from start condition to stop condition must be within $T_{LOW:SEXT}$: 25 ms (max.).
For the ACK receive timing (rising edge of the ninth SCL), monitor the ICSR2.TEND flag in master transmit mode (master transmitter) and the ICSR2.RDRF flag in master receive mode (master receiver). For this reason, perform bytewise transmit operation in master transmit mode, and hold the ICMR3.RDRFS bit 0 until the byte just before reception of the final byte in master receive mode. While the RDRFS bit is 0, the RDRF flag is set to 1 at the rising edge of the ninth SCL.
If the period measured with the MTU or TMR exceeds the cumulative clock low extend time (master device) ($T_{LOW:MEXT}$: 10 ms (max.)) of the SMBus specification or the total of measured periods exceeds the detect clock low timeout ($T_{TIMEOUT}$: 25 ms (min.)) of the SMBus specification, the master device must stop the transaction by generating a stop condition. In master transmit mode, immediately stop the transmit operation (writing data to the ICDRT register).

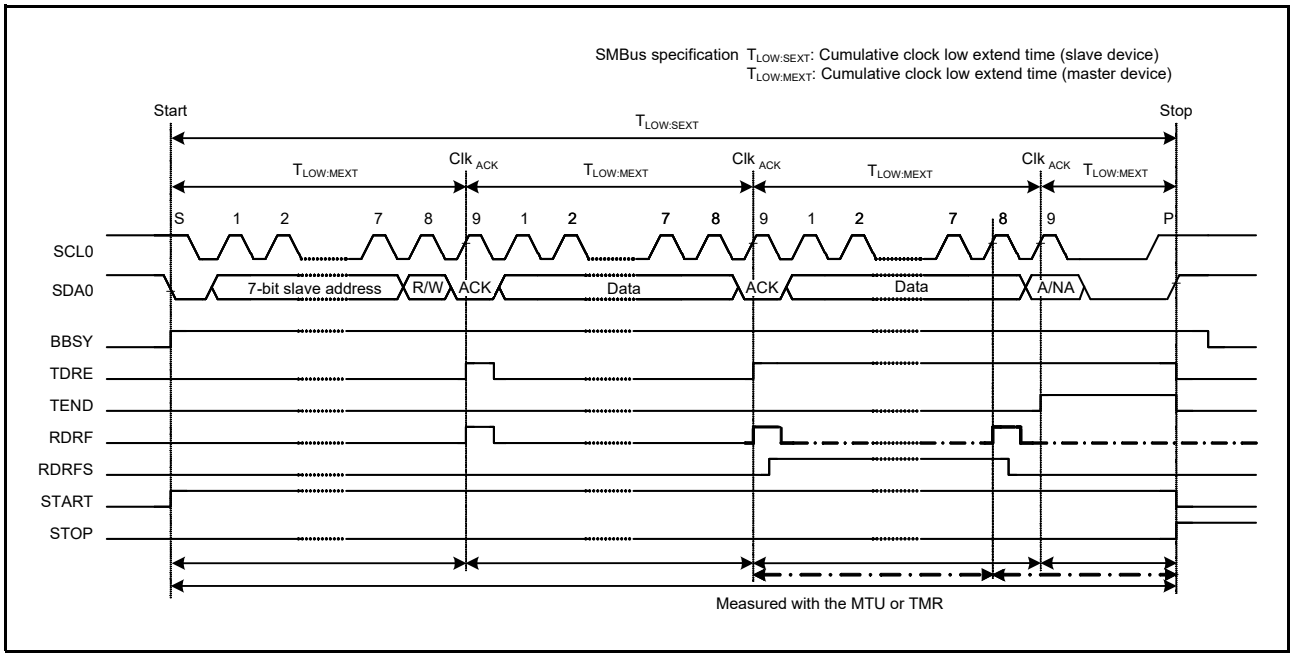


Figure 34.40 SMBus Timeout Measurement

34.12.2 Packet Error Code (PEC)

This MCU incorporates a CRC calculator. The CRC calculator enables transmission of a packet error code (PEC) or checking the received data of the SMBus in data communication of the RIIC. For the CRC generating polynomials of the CRC calculator, refer to section 39, CRC Calculator (CRCA).

The PEC data in master transmit mode can be generated by writing all transmit data to the CRC data input register (CRCDIR) in the CRC calculator.

The PEC data in master receive mode can be checked by writing all receive data to CRCDIR in the CRC calculator and comparing the obtained value in the CRC data output register (CRCDOR) with the received PEC data.

To send ACK or NACK according to the match or mismatch result when the final byte is received as a result of the PEC code check, set the ICMR3.RDRFS bit to 1 before the rising edge of the eighth SCL during reception of the final byte, and hold the SCL0 line low at the falling edge of the eighth clock pulse.

34.12.3 SMBus Host Notification Protocol (Notify ARP Master Command)

In communications over an SMBus, a slave device can temporarily act as a master device to notify the SMBus host (or ARP master) of its own slave address or to request its own slave address from the SMBus host.

For a product of this MCU to operate as an SMBus host (or ARP master), the host address (0001 000b) sent from the slave device must be detected as a slave address, so the RIIC has a function for detecting the host address. To detect the host address as a slave address, set the ICMR3.SMBS bit and the ICSER.HOAE bit to 1. Operation after the host address has been detected is the same as normal slave operation.

34.13 Interrupt Sources

The RIIC generates four types of interrupt requests: communication error or communication event (arbitration-lost detection, NACK detection, timeout detection, start condition detection, and stop condition detection), receive data full, transmit data empty, and transmission end.

Table 34.6 lists details of the several interrupt requests. The receive data full and transmit data empty interrupt requests allow the DTC or DMAC to start data transfer.

Table 34.6 Interrupt Sources

Symbol	Interrupt Source	Interrupt Flag	Start DTC/DMA Transfer	Interrupt Generation Condition
EEI	Communication error/ communication event	AL	Not possible	AL = 1 and ALIE = 1
		NACKF		NACKF = 1 and NAKIE = 1
		TMOF		TMOF = 1 and TMOIE = 1
		START		START = 1 and STIE = 1
		STOP		STOP = 1 and SPIE = 1
RXI*2	Receive data full	RDRF	Possible	RDRF = 1 and RIE = 1
TXI*1	Transmit data empty	TDRE	Possible	TDRE = 1 and TIE = 1
TEI*3	Transmission end	TEND	Not possible	TEND = 1 and TEIE = 1

Note: There is a delay time between the execution of a write instruction for a peripheral module by the CPU and actual writing to the module. Thus, when an interrupt flag has been cleared or an interrupt request has been masked, read the relevant flag again to check whether clearing or masking has been completed, and then return from interrupt handling. Returning from interrupt handling without checking that writing to the module has been completed creates a possibility of repeated processing of the same interrupt.

Note 1. Because TXI is an edge-detected interrupted, it does not require clearing. Furthermore, the ICSR2.TDRE flag (a condition for TXI) is automatically set to 0 when data for transmission are written to the ICDRT register or a stop condition is detected (ICSR2.STOP flag is 1).

Note 2. Because RXI is an edge-detected interrupted, it does not require clearing. Furthermore, the ICSR2.RDRF flag (a condition for RXI) is automatically set to 0 when data are read from the ICDRR register.

Note 3. When using the TEI interrupt, clear the ICSR2.TEND flag in the TEI interrupt handling.
Note that the ICSR2.TEND flag is automatically set to 0 when data for transmission are written to the ICDRT register or a stop condition is detected (ICSR2.STOP flag is 1).

Clear the each flag or mask the interrupt request during interrupt handling.

34.13.1 Buffer Operation for TXI and RXI Interrupts

If the conditions for generating a TXI and RXI interrupt are satisfied while the corresponding ICU.IRn.IR flag is 1, the interrupt request is not output for the ICU but retained internally (the capacity for internal retention is one request per source).

An interrupt request that was being retained internally is output to the ICU when the value of the IR flag becomes 0.

Internally retained interrupt requests are automatically cleared under normal conditions of usage.

Internally retained interrupt requests can also be cleared by writing 0 to the corresponding interrupt enable bit in the ICIER register.

34.14 Initialization of Registers and Functions When a Reset is Applied or a Condition is Detected

The RIIC can be reset by MCU reset, RIIC reset, and internal reset functions. Table 34.7 lists the reset states of registers and functions when a reset is applied or a condition is detected.

Table 34.7 Reset States of Registers and Functions When a Reset is Applied or a Condition is Detected

		MCU Reset	RIIC Reset (ICE = 0, IICRST = 1)	Internal Reset (ICE = 1, IICRST = 1)	Start Condition/ Restart Condition Detection	Stop Condition Detection
ICCR1	SDAO, SCLO	To be reset	To be reset	To be reset	Retained	Retained
	IICRST, ICE		Retained	Retained		
	Others		To be reset			
ICCR2	ST, RS	To be reset	To be reset	To be reset	To be reset	Retained
	SP					To be reset
	TRS				See note 1	
	MST				See note 1	
	BBSY			Retained	Becomes 1	
ICMR1	BC[2:0]	To be reset	To be reset	To be reset	To be reset	Retained
	Others			Retained	Retained	
ICMR2		To be reset	To be reset	Retained	Retained	Retained
ICMR3	ACKBT	To be reset	To be reset	Retained	Retained	To be reset
	Others					Retained
ICFER		To be reset	To be reset	Retained	Retained	Retained
ICSER		To be reset	To be reset	Retained	Retained	Retained
ICIER		To be reset	To be reset	Retained	Retained	Retained
ICSR1		To be reset	To be reset	To be reset	Retained	To be reset
ICSR2	START	To be reset	To be reset	To be reset	Becomes 1	To be reset
	STOP				Retained	Becomes 1
	TEND					To be reset
	TDRE				See note 1	
	Others				Retained	Retained
SARL0, SARL1, SARL2, SARU0, SARU1, SARU2		To be reset	To be reset	Retained	Retained	Retained
ICBRH, ICBRL		To be reset	To be reset	Retained	Retained	Retained
ICDRT		To be reset	To be reset	Retained	Retained	Retained
ICDRR		To be reset	To be reset	Retained	Retained	Retained
ICDRS		To be reset	To be reset	To be reset	Retained	Retained
Timeout function		To be reset	To be reset	To be reset	Operation	Operation
Bus free time measurement		To be reset	To be reset	Operation	Operation	Operation

Note 1. This bit is not reset. This bit becomes 0 or 1 in accordance with the conditions.

34.15 Event Link Function (Output)

The RIIC0 handles event output for the event link controller (ELC) corresponding to the following sources.

- Communication error/communication event
- Receive data full
- Transmit data empty
- Transmission end

34.15.1 Interrupt Handling and Event Linking

The RIIC has four types of interrupts: communication error or communication event (arbitration-lost detection, NACK detection, timeout detection, start condition detection, or stop condition detection), receive data full, transmit data empty, and transmission end. Each of these has an enable bit to control enabling and disabling of the interrupt signal. An interrupt request signal is output for the ICU when an interrupt source condition is satisfied while the setting of the corresponding enable bit is enabled.
The corresponding event signals are sent to other modules via the ELC when the interrupt source conditions are satisfied, regardless of the settings of the interrupt enable bits.
For details on interrupt sources, see Table 34.6.

34.16 Usage Notes

34.16.1 Setting Module Stop Function

Module stop state can be entered or released using module stop control register B (MSTPCRB). The initial setting is for operation of the RIIC to be stopped. RIIC register access is enabled by releasing the module stop state.
For details on module stop control register B, refer to section 11, Low Power Consumption.

34.16.2 Notes on Starting Transfer

If the IR flag corresponding to the RIIC interrupt is 1 when transfer is started (ICCR1.ICE bit is 1), follow the procedure below to clear interrupts before enabling operations. Starting transfer with the IR flag set to 1 while the ICCR1.ICE bit is 1 leads to an interrupt request being internally retained after transfer starts, and this can lead to unanticipated behavior of the IR flag.

1. Confirm that the ICCR1.ICE bit is 0.
2. Set the relevant interrupt enable bits (ICIER.TIE, etc.) on the peripheral function side to 0.
3. Read the relevant interrupt enable bits (ICIER.TIE, etc.) on the peripheral function side and confirm that its value is 0.
4. Set the IR flag to 0.

35. I3C Bus Interface (RI3C)

This MCU has a single-channel I3C bus interface (RI3C0).
The RI3C module conforms with the MIPI Alliance I3C bus interface and provides a subset of its functions.

35.1 Overview

Table 35.1 lists the specifications of the RI3C and Figure 35.1 shows a block diagram of the RI3C. Table 35.2 lists the I/O pins of the RI3C.

Table 35.1 RI3C Specifications

Item	Specification
Operation mode	Controller (Primarily Controller/Secondary Controller) mode and Target mode selectable
Data handler	• Controller: - FIFO buffer transfer • Target: - FIFO buffer transfer
Communication protocol	• SDR (I3C single data rate) mode - Private message - Broadcast message (common command code (CCC)) - Direct message (common command code (CCC)) • Legacy I^2C message - Fast-mode (Fm): 0 to 400 kbps - Fast-mode Plus (Fm+): 0 to 1 Mbps
IBI	• Target Interrupt Request (TIR) • Controller Role Request (CRR) (Secondary Controller only) • Hot-Join event
Address format	7-bit address
Address detection	• Target address (Static Address or Dynamic Address) • Broadcast Address (7Eh)
Clock stalling	The I3C Controller can stall the I3C bus while the SCL line is low.
Interrupt source	Seven sources: • Response queue full (RESPI) • Command queue empty (CMDI) • IBI queue empty/full (IBII) • Receive status queue full (RCVI) • Receive data full (RXI) • Transmit data empty (TXI) • Communication error or communication event (EEI) - START condition detection (including repeated START condition) - STOP condition detection - HDR Exit Pattern detection - Timeout detection - Buffer access error detection - Data transfer abort - Data transfer error
Error detection	• Buffer access error • Address header error • Address NACKed or Dynamic Address Assignment NACKed • Receive overflow or transmit underflow error • Transfer abort • NACK is received during an I2C write data transfer • Timeout error
Event link output	• Receive data full event • Transmit data empty event • Communication error or communication event

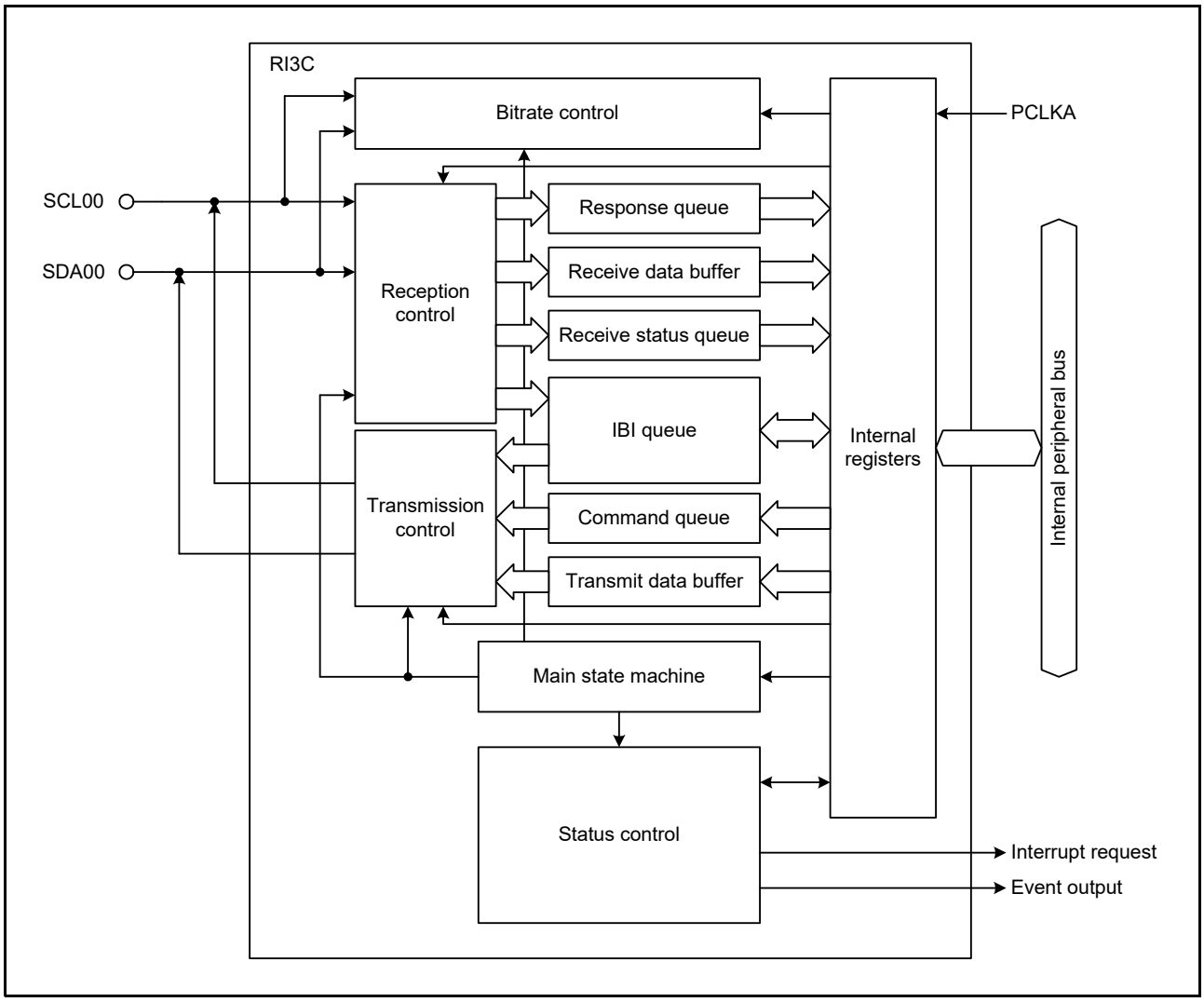


Figure 35.1 RI3C Block Diagram

Table 35.2 RI3C Pin Configuration

Channel	Pin Name	I/O	Function
RI3C0	SCL00	I/O	Serial clock I/O pin
	SDA00	I/O	Serial data I/O pin

35.2 Register Descriptions

35.2.1 Mode Register (ICMR)

Address(es): RI3C0.ICMR 000E C000h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OMS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	OMS	Operating Mode Select	Set this bit to 0.	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R

OMS Bit (Operating Mode Select)

Set this bit to 0.

35.2.2 Control Register (ICCR)

Address(es): RI3C0.ICCR 000E C014h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ICE	RESUME	ABORT	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	HJC	—	—	—	—	—	—	—	IBAINC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IBAINC	I3C Broadcast Address Include *1	0: Do not include I3C Broadcast Address for private transfers 1: Include I3C Broadcast Address for private transfers	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	HJC	Hot-Join Control*1	0: ACK the Hot-Join Request 1: NACK and send Broadcast CCC to disable Hot-Join	R/W
b28 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R
b29	ABORT	Host Controller Abort*1	Writing 1 to this bit aborts the currently issued transfer. Write 0 to this bit to clear the Aborted state. Values when read: 0: Aborted state is cleared. 1: RI3C remains in Aborted state.	R/W
b30	RESUME	Host Controller Resume	Writing 1 to this bit while RI3C is in the Halt state can resume the operation. Values when read: 0: RI3C is running. 1: RI3C is suspended.	R/W
b31	ICE	Bus Interface Enable	0: The I3C bus operation is disabled. (SCL00 and SDA00 pins in inactive state) 1: The I3C bus operation is enabled. (SCL00 and SDA00 pins in active state)	R/W

Note 1. This bit is used in I3C Primarily Controller and I3C Secondary Controller modes.

IBAINC Bit (I3C Broadcast Address Include)

This bit controls whether the I3C Broadcast Address (7Eh) is included for private transfers.

If the I3C Broadcast Address is not included for private transfers, then IBIs driven from Target devices might not win the arbitration, potentially delaying acceptance of the IBIs.

HJC Bit (Hot-Join Control)

This bit acts as global control to either ACK or NACK all Hot-Join Requests arriving from the devices on the I3C bus. If set to NACK, then the NACK will be followed by the Broadcast CCC to disable Hot-Join.

ABORT Bit (Host Controller Abort)

When 1 is written to this bit, RI3C starts the abort operation which terminates the currently issued transfer.

In response to an Abort request, RI3C issues the STOP condition to terminate the current transaction on the I3C bus after the complete data byte is transferred or received. At this time, this bit becomes 1.

To resume operation, write 0 to this bit.

If this bit is set to 1 to abort the transaction, ignore the ERR_STATUS field in the Response Descriptor.

RESUME Bit (Host Controller Resume)

This bit is used to resume I3C operation following the Halt state.
RI3C enters the Halt state as a result of any type of error occurring in a transfer. The error type is indicated by the ERR_STATUS field in Response Descriptor.
After RI3C has entered the Halt state, the application must write 1 to this bit to resume operation. This bit is automatically set to 0 when RI3C restarts the transfer (starting the next command).

ICE Bit (Bus Interface Enable)

Enables or disables the operation on the I3C bus.
When using RI3C, set this bit to 1. When this bit is 1, the SCL00 and SDA00 pins are active.
Set this bit to 0 when RI3C is not used. When the ICE bit is 0, the SCL00 and SDA00 pins are inactive.
When this bit is read as 0, it indicates that I3C bus operation is prohibited.

35.2.3 Controller Device Address Register (ICCAR)

Address(es): RI3C0.ICCAR 000E C018h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DAV	—	—	—	—	—	—	—	—	DADR[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b22 to b16	DADR[6:0]	Dynamic Address	Set the Dynamic Address of RI3C.	R/W
b30 to b23	—	Reserved	These bits are read as 0. The write value should be 0.	R
b31	DAV	Dynamic Address Valid	0: The DADR[6:0] bits are not valid. 1: The DADR[6:0] bits are valid.	R/W

Note: This register is used in I3C Primarily Controller mode.

This register is used to store the Dynamic Address of RI3C.
When the ICCR.ICE bit is set to 1 after setting this register, the device will act as Primary Controller.
If the ICCR.ICE bit is set to 1 after setting the ICDCTR.ROLE[1:0] bits to 00b (I3C Target) or the ICTDCTRm.ROLE[1:0] bits to 01b (I3C Controller capable) without setting this register, the device will act as Target.

DADR[6:0] Bits (Dynamic Address)

These bits are used to program RI3C Controller Dynamic Address. RI3C uses this address to respond to Controller transactions in I3C interface mode (Target or Secondary Controller mode).
In I3C Active Controller mode, the software shall program the Dynamic Address as it self-assigns its Dynamic Address.

DAV Bit (Dynamic Address Valid)

This bit indicates whether or not the value in the DADR[6:0] bits is valid.
In I3C Active Controller mode, set this bit to 1 as it self-assigns its Dynamic Address.

35.2.4 Reset Control Register (ICRCR)

Address(es): RI3C0.ICRCR 000E C020h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ISRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	SQRST	IQRST	RBRST	TBRST	RQRST	CQRST	MRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MRST	Module Reset*1	Writing 1 to this bit resets all registers and internal circuits. This bit automatically returns to 0 when the reset is complete.	R/W
b1	CQRST	Command Queue Reset	Writing 1 to this bit flushes the command queue. This bit automatically returns to 0 when the command queue has been flushed.	R/W
b2	RQRST	Response Queue Reset	Writing 1 to this bit flushes the response queue. This bit automatically returns to 0 when the response queue has been flushed.	R/W
b3	TBRST	Transmit Data Buffer Reset	Writing 1 to this bit flushes the transmit data buffer. This bit automatically returns to 0 when the transmit data buffer has been flushed.	R/W
b4	RBRST	Receive Data Buffer Reset	Writing 1 to this bit flushes the receive data buffer. This bit automatically returns to 0 when the receive data buffer has been flushed.	R/W
b5	IQRST	IBI Queue Reset	Writing 1 to this bit flushes the IBI queue. This bit automatically returns to 0 when the IBI queue has been flushed.	R/W
b6	SQRST	Receive Status Queue Reset*2	Writing 1 to this bit flushes the receive status queue. This bit automatically returns to 0 when the receive status queue has been flushed.	R/W
b15 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	ISRST	Internal Status Reset	0: Some registers and internal circuits are released from reset. 1: Some registers and internal circuits are reset.	R/W
b31 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Do not write any value to this register when this bit is 1.
Note 2. This bit is used in I3C Secondary Controller and I3C Target modes.

ISRST Bit (Internal Status Reset)

When this bit is set to 1, some registers and internal circuits are reset. Refer to section 35.8, Reset Description for detail on the registers.
Set this bit to 0 to release from the reset.

35.2.5 Operating Mode Monitor Register (ICMMR)

Address(es): RI3C0.ICMMR 000E C024h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	WP	—	—	—	—	ACF	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b2	ACF	Active Controller Flag*2	0: RI3C is not the Active Controller. 1: RI3C is the Active Controller.	R/W*1
b6 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R
b7	WP	Write Protect*2	0: Writing to the ACF flag is disabled. 1: Writing to the ACF flag is enabled (when writing simultaneously with the value of the target bit).	W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. When setting a value to the ACF flag, set the WP bit to 1 at the same time.
Note 2. This bit is used in I3C Primarily Controller and I3C Secondary Controller modes.

ACF Flag (Active Controller Flag)

This flag indicates whether the RI3C is presently the Active Controller.
[Setting conditions]

- When 1 is written to the WP bit and the ACF flag at the same time.
- When 1 is written to the ICCAR.DAV bit. (in I3C Primarily Controller mode)
- When GETACCCR reception is successfully completed by issuing STOP condition, after the ACK is responded to the Controller Role Request transmitted to the Secondary Controller. (in I3C Primarily Controller mode)
- When GETACCCR reception is successfully completed by issuing STOP condition, after the ACK is responded to the Controller Role Request transmitted to the Active Controller. (in I3C Secondary Controller mode)

[Clearing conditions]

- When the RI3C is reset by setting the ICRCR.MRST bit to 1.
- When the internal status is reset by setting the ICRCR.ISRST bit to 1.
- When 1 is written to the WP bit and 0 is written to the ACF flag at the same time.
- When 0 written to the ICCAR.DAV bit. (in I3C Primarily Controller mode)
- When GETACCCR transmission is successfully completed by issuing STOP condition, after responding ACK to the Controller Role Request received from the Secondary Controller. (in I3C Primarily Controller mode)
- When GETACCCR transmission is successfully completed by issuing STOP condition, after responding ACK to the Controller Role Request received from another Controller-capable device. (in I3C Secondary Controller mode)

WP Bit (Write Protect)

This bit is used to rewrite the ACF flag.
When setting a value to the ACF flag, set this bit to 1 at the same time.
This bit is automatically set to 0. The read value is 0.

35.2.6 Internal Status Register (ICISR)

Address(es): RI3C0.ICISR 000E C030h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	BERF	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b10	BERF	Buffer Access Error Flag	0: Buffer access error has not been detected. 1: Buffer access error was detected.	R/W*1
b31 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. To clear this flag, confirm that the flag is 1 and then set it to 0.

BERF Flag (Buffer Access Error Flag)

This flag indicates that an access error to data buffers or queue buffers has been detected.

[Setting conditions]

When any of the following conditions is satisfied while the ICISER.BERDE bit is 1.

- When a transmit data is written while the transmit data buffer is completely full.
- When a received data is read while the receive data buffer is completely empty.
- When a Command Descriptor is written while the command queue is completely full.
- When a Response Descriptor is read while the response queue is completely empty.
- When a Receive Status Descriptor is read while the receive status queue is completely empty.
- When an IBI Status Descriptor is read while the ICMMR.ACF flag is 1 (Active Controller) and the IBI queue is completely empty.
- When an IBI data is written while the ICMMR.ACF flag is 0 (not Active Controller) and the IBI queue is completely full.
- When the response queue, IBI queue, or receive status queue overflows.

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1.

35.2.7 Internal Status Detection Enable Register (ICISER)

Address(es): RI3C0.ICISER 000E C034h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	BERDE	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b10	BERDE	Buffer Access Error Detection Enable	0: Buffer access error detection is disabled. 1: Buffer access error detection is enabled.	R/W
b31 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R

35.2.8 Internal Status Interrupt Enable Register (ICISIER)

Address(es): RI3C0.ICISIER 000E C038h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	BERIE	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b10	BERIE	Buffer Access Error Detection Interrupt Enable	0: Buffer access error detection interrupt is disabled. 1: Buffer access error detection interrupt is enabled.	R/W
b31 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R

BERIE Bit (Buffer Access Error Detection Interrupt Enable)

When the BERF flag becomes 1 while this bit is 1, an interrupt is generated.

35.2.9 Device Characteristics Table Index Register (ICDCTIR)

Address(es): RI3C0.ICDCTIR 000E C044h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	INDEX[4:0]					—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b18 to b0	—	Reserved	These bits are read as 0.	R
b23 to b19	INDEX[4:0]	DCT Table Index	Current index of the DCT, which is used as the starting index for the I3C ENTDAA CCC.	R
b31 to b24	—	Reserved	These bits are read as 0.	R

Note: This register is used in I3C Primarily Controller and I3C Secondary Controller modes.

INDEX[4:0] Bits (DCT Table Index)

Once the complete characteristics of device that won the arbitration are written to the Device Characteristics Table (DCT) during ENTDAA using address assignment command, this index is incremented by 1.

35.2.10 IBI Notify Control Register (ICINCR)

Address(es): RI3C0.ICINCR 000E C058h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	RTIRN	—	RCRRN	RHJRN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RHJRN	Rejected Hot-Join Request Notify	0: Do not pass the rejected IBI status to the IBI queue, if a Hot-Join Request is rejected. 1: Pass the rejected IBI status to the IBI queue, if a Hot-Join Request is rejected.	R/W
b1	RCRRN	Rejected Controller Role Request Notify	0: Do not pass the rejected IBI status to the IBI queue, if a Controller Role Request is rejected. 1: Pass the rejected IBI status to the IBI queue, if a Controller Role Request is rejected.	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R
b3	RTIRN	Rejected Target Interrupt Request Notify	0: Do not pass the rejected IBI status to the IBI queue, if a Target Interrupt Request is rejected. 1: Pass the rejected IBI status to the IBI queue, if a Target Interrupt Request is rejected.	R/W
b31 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This register is used in I3C Primarily Controller and I3C Secondary Controller modes.

RHJRN Bit (Rejected Hot-Join Request Notify)

This bit enables or disables reporting rejection of individual Hot-Join Requests, if the incoming Hot-Join Request is NACKed and is auto-disabled based on the setting of the ICCR.HJC bit.

RCRRN Bit (Rejected Controller Role Request Notify)

This bit enables or disables reporting rejection of individual Controller Role Requests (CRR), if the incoming CRR is NACKed and is auto-disabled based on the setting of the relevant ICTDATRm.CRRRJ bit (m = 0 to 3).

RTIRN Bit (Rejected Target Interrupt Request Notify)

This bit enables or disables reporting rejection of individual Target Interrupt Requests (TIR), if the incoming TIR is NACKed and is auto-disabled based on the setting of the relevant ICTDATRm.TIRRJ bit (m = 0 to 3).

35.2.11 Target Mode Control Register (ICTCR)

Address(es): RI3C0.ICTCR 000E C064h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TA0DE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	TA0DE	Target Address 0 Detection Enable*1	0: The value of the ICDAMR0.TADR[9:0] bits is disabled. 1: The value of the ICDAMR0.TADR[9:0] bits is enabled.	R/W
b31 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. These bits are used in I3C Secondary Controller and I3C Target modes.

TA0DE Bit (Target Address 0 Detection Enable)

This bit is used to enable or disable the Target address set in ICDAMR0.TADR[9:0] bits.

When this bit is set to 1, the Target address set in ICDAMR0.TADR[9:0] bits is enabled and is compared with the received Target address.

When this bit is set to 0, the Target address set in ICDAMR0.TADR[9:0] bits is disabled and is ignored even if it matches the received Target address.

35.2.12 Standard Bitrate Register (ICSBR)

Address(es): RI3C0.ICSBR 000E C074h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ODDBL	—	PPHW[5:0]						—	—	PPLW[5:0]					
Value after reset:	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ODHW[7:0]								ODLW[7:0]							
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	ODLW[7:0]	Open-Drain Low Width Setting*1	Count value for the low period of SCL in Open-Drain mode.	R/W
b15 to b8	ODHW[7:0]	Open-Drain High Width Setting*1	Count value for the high period of SCL in Open-Drain mode.	R/W
b21 to b16	PPLW[5:0]	Push-Pull Low Width Setting*1	Count value for the low period of SCL in Push-Pull mode.	R/W
b23, b22	—	Reserved	These bits are read as 0. The write value should be 0.	R
b29 to b24	PPHW[5:0]	Push-Pull High Width Setting*1	Count value for the high period of SCL in Push-Pull mode.	R/W
b30	—	Reserved	These bits are read as 0. The write value should be 0.	R
b31	ODDBL	Open-Drain High/Low Width Doubling*1	0: Set the high and low widths to the set value in Open-Drain mode. 1: Double the high and low widths of the set values in Open-Drain mode.	R/W

Note 1. These bits are used in I3C Primarily Controller and I3C Secondary Controller modes.

This register is used to set the bit rate when the I3C SDR0 mode is specified in the MODE field of the Command Descriptor. When the I3C SDR2 mode is specified, a bit rate of 1/2 the bit rate set in this register is used.

ODLW[7:0] Bits (Open-Drain Low Width Setting)

The ODLW[7:0] bits are used to set the low period of SCL in Open-Drain mode.
The RI3C counts the low period with PCLKA.

ODHW[7:0] Bits (Open-Drain High Width Setting)

The ODHW[7:0] bits are used to set the high period of SCL in Open-Drain mode.
These bits are valid in Controller mode.
The RI3C counts the high period with PCLKA.

PPLW[5:0] Bits (Push-Pull Low Width Setting)

The PPLW[5:0] bits are used to set the low period of SCL in Push-Pull mode.
RI3C counts the low period with PCLKA.

PPHW[5:0] Bits (Push-Pull High Width Setting)

The PPHW[5:0] bits are used to set the high period of SCL in Push-Pull mode.
These bits are valid in Controller mode.
RI3C counts the high period with PCLKA.

The transfer rate and the duty cycle are calculated using the following expression.

t_{LOW} = ODLW[7:0] / PCLKA
t_{HIGH} = ODHW[7:0] / PCLKA
Transfer rate = 1 / (t_{LOW} + t_{HIGH} + t_r + t_f)
Duty cycle = (t_r + t_{HIGH}) / (t_{LOW} + t_{HIGH} + t_r + t_f)

t_{LOW}: Low period of the SCL
t_{HIGH}: high period of the SCL
t_r: Rise time of the SCL signal*1
t_f: Fall time of the SCL signal*1

Note 1. The SCL line rise time [t_r] and SCL line fall time [t_f] depend on the total bus line capacitance [C_b] and the pull-up resistor [R_p]. For details, refer to the I3C bus standard from MIPI Alliance.

35.2.13 Extended Bitrate Register (ICEBR)

Address(es): RI3C0.ICEBR 000E C078h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	PPHW[5:0]						—	—	PPLW[5:0]					
Value after reset:	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ODHW[7:0]								ODLW[7:0]							
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	ODLW[7:0]	Open-Drain Low Width Setting*1	Count value for the low period of SCL in Open-Drain mode.	R/W
b15 to b8	ODHW[7:0]	Open-Drain High Width Setting*1	Count value for the high period of SCL in Open-Drain mode.	R/W
b21 to b16	PPLW[5:0]	Push-Pull Low Width Setting*1	Count value for the low period of SCL in Push-Pull mode.	R/W
b23, b22	—	Reserved	These bits are read as 0. The write value should be 0.	R
b29 to b24	PPHW[5:0]	Push-Pull Low Width Setting*1	Count value for the high period of SCL in Push-Pull mode.	R/W
b31, b30	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. These bits are used in I3C Primarily Controller and I3C Secondary Controller modes.

This register is used to set the bit rate when the I3C SDR1 mode is specified in the MODE field of the Command Descriptor. When the I3C SDR3 mode is specified, a bit rate of 1/2 the bit rate set in this register is used. When the I3C SDR4 mode is specified, a bit rate of 1/4 the bit rate set in this register is used.

ODLW[7:0] Bits (Open-Drain Low Width Setting)

The ODLW[7:0] bits are used to set the low period of SCL in Open-Drain mode.
The RI3C counts the low period with PCLKA.

ODHW[7:0] Bits (Open-Drain High Width Setting)

The ODHW[7:0] bits are used to set the high period of SCL in Open-Drain mode.
These bits are valid in Controller mode.
The RI3C counts the high period with PCLKA.

PPLW[5:0] Bits (Push-Pull Low Width Setting)

The PPLW[5:0] bits are used to set the low period of SCL in Push-Pull mode.
The RI3C counts the low period with the PCLKA.

PPHW[5:0] Bits (Push-Pull Low Width Setting)

The PPHW[5:0] bits are used to set the high period of SCL in Push-Pull mode.
These bits are valid in Controller mode.
The RI3C counts the high period with the PCLKA.

The transfer rate and the duty cycle are calculated using the following expression.

t_{LOW} = ODLW[7:0] / PCLKA
t_{HIGH} = ODHW[7:0] / PCLKA
Transfer rate = 1 / (t_{LOW} + t_{HIGH} + t_r + t_f)
Duty cycle = (t_r + t_{HIGH}) / (t_{LOW} + t_{HIGH} + t_r + t_f)

t_{LOW}: Low period of the SCL
t_{HIGH}: high period of the SCL
t_r: Rise time of the SCL signal*1
t_f: Fall time of the SCL signal*1

Note 1. The SCL line rise time [t_r] and SCL line fall time [t_f] depend on the total bus line capacitance [C_b] and the pull-up resistor [R_p]. For details, refer to the I3C bus standard from MIPI Alliance.

35.2.14 Bus Free Time Setting Register (ICBFTR)

Address(es): RI3C0.ICBFTR 000E C07Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is used to specify the time from when the STOP condition is detected until the ICBSR.BFREE flag is set to 1. The upper 23 bits are reserved. These bits are read as 0 and the write value should be 0.
The number set in this register is counted using PCLKA as the count source.

35.2.15 Bus Available Time Setting Register (ICBATR)

Address(es): RI3C0.ICBATR 000E C080h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is used to specify the time from when the STOP condition is detected until the ICBSR.BAVL flag is set to 1. The upper 23 bits are reserved. These bits are read as 0 and the write value should be 0.
The number set in this register is counted using PCLKA as the count source.

35.2.16 Bus Idle Time Setting Register (ICBITR)

Address(es): RI3C0.ICBITR 000E C084h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is used to specify the time from when the STOP condition is detected until the ICBSR.BIDL flag is set to 1.
The upper 14 bits are reserved. These bits are read as 0 and the write value should be 0.
The number set in this register is counted using PCLKA as the count source.

35.2.17 Output Signal Control Register (ICOCR)

Address(es): RI3C0.ICOCR 000E C088h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	SOWP	SCLO	SDAO
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	SDAO	SDA Output Control*1	0: The RI3C drives the SDA00 pin low. 1: The RI3C releases the SDA00 pin.	R/W
b1	SCLO	SCL Output Control*1	0: The RI3C drives the SCL00 pin low. 1: The RI3C releases the SCL00 pin.	R/W
b2	SOWP	SCLO/SDAO Write Protect*1	0: Writing to the SCLO and SDAO bits is disabled. 1: Writing to the SCLO and SDAO bits is enabled (when writing simultaneously with the value of the target bit). This bit is read as 0.	W
b31 to b3	—	Reserved	This bit is read as 0. The write value should be 0.	R

Note 1. This bit is used in I3C Primarily Controller and I3C Secondary Controller modes.

SDAO Bit (SDA Output Control) and SCLO Bit (SCL Output Control)

These bits are used to directly control the SDA00 and SCL00 signals output from the RI3C.

When writing to these bits, also write 1 to the SOWP bit at the same time.

The result of setting these bits is input to the RI3C via the input buffer. When Target mode is selected, a START condition may be detected and the bus may be released depending on the bit settings.

Do not rewrite these bits during a START condition, STOP condition, repeated START condition, transmission, or reception. Operation after rewriting under the above conditions is not guaranteed.

35.2.18 Timeout Control Register (ICTOR)

Address(es): RI3C0.ICTOR 000E C090h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	TMOM[1:0]		TMOH	TMOL	—	—	TMOS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TMOS[1:0]	Timeout Detection Time Select	b1 b0 0 0: Timeout counter is 16 bits (up to 65536 counts) 0 1: Timeout counter is 14 bits (up to 16384 counts) 1 0: Timeout counter is 8 bits (up to 256 counts) 1 1: Timeout counter is 6 bits (up to 64 counts)	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b4	TMOL	Timeout L Count Control	0: Count is disabled while the SCL00 line is low. 1: Count is enabled while the SCL00 line is low.	R/W
b5	TMOH	Timeout H Count Control	0: Count is disabled while the SCL00 line is high. 1: Count is enabled while the SCL00 line is high.	R/W
b7, b6	TMOM[1:0]	Timeout Detection Mode Select	b7 b6 0 0: Timeout function is enabled in any of the following conditions. • When the ICBSR.BFREE flag is 0 (bus busy) in Controller mode • When the ICBSR.BFREE flag is 0 (bus busy) when the received Target address matches its own Target address in Target mode 0 1: Timeout function is enabled during ICBSR.BFREE flag is 0 (bus busy). 1 0: Timeout function is enabled during ICBSR.BFREE flag is 1 (bus free). 1 1: Setting prohibited	R/W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R

TMOS[1:0] Bits (Timeout Detection Time Select)

These bits are used to select for the timeout detection time when the timeout function is enabled (ICSER.TMOE bit = 1). While the SCL00 line is in the state that enables this counter as specified by bits TMOH and TMOL, the counter counts up in synchronization with PCLKA as a count source.

For details on the timeout function, refer to 35.5.5.3 Timeout Error Detection in section 35.5.5, Error Detection.

TMOL Bit (Timeout L Count Control)

This bit is used to enable or disable the internal counter of the timeout function to count up while the SCL00 line is held low when the timeout function is enabled (ICSER.TMOE bit = 1).

TMOH Bit (Timeout H Count Control)

This bit is used to enable or disable the internal counter of the timeout function to count up while the SCL00 line is held high when the timeout function is enabled (ICSER.TMOE bit = 1).

TMOM[1:0] Bits (Timeout Detection Mode Select)

These bits are used to set the timeout detection condition when the timeout function is enabled (ICSER.TMOE bit = 1).

When these bits are set to 00b, timeout detection is enabled during one of the following periods.

- When the bus is busy in Controller mode
- When RI3C is the target of the communication in Target mode and the bus is busy

When these bits are set to 01b, timeout detection is enabled when the bus is busy.

When these bits are set to 10b, timeout detection is enabled when the bus is free.

35.2.19 Clock Stall Control Register (ICSTCR)

Address(es): RI3C0.ICSTCR 000E C0B0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	APSE	PBSE	—	AASE	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	STT[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	STT[15:0]	Stall Time Setting	Set the clock stall time for each phase.	R/W
b27 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R
b28	AASE	Assigned Address Phase Stall Enable	0: Do not stall SCL before the Assigned Address phase. 1: Stall SCL before the Assigned Address phase.	R/W
b29	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b30	PBSE	Parity Bit Stall Enable	0: Does not stall the SCL during the parity bit period. 1: Stall the SCL during the parity bit period.	R/W
b31	APSE	ACK/NACK Phase Stall Enable	0: Does not stall the SCL during the ACK/NACK phase. 1: Stall the SCL during the ACK/NACK phase.	R/W

Note: This register is used in I3C Primarily Controller and I3C Secondary Controller modes.

When setting this register, follow section 5.1.2.5, Controller Clock Stalling of MIPI I3C Specification, and use it only when necessary because of its negative impacts on bus performance.

STT[15:0] Bits (Stall Time Setting)

These bits is used to set the SCL stall time. The SCL stall time is counted by PCLKA. This is a counter common to the enable bits of each phase.

AASE Bit (Assigned Address Phase Stall Enable)

This bit is used to select whether the Controller stalls SCL during the low period of the first bit of the Assigned Address phase of the ENTDAA CCC command. It can gain time in assigning Dynamic Address to the Device based on the BCR and DCR of the Target. However, because the Dynamic Address Assignment procedure sends the Dynamic Address set in the ICTDATRm register in sequence, it is not necessary to set this bit and it is prohibited.

PBSE Bit (Parity Bit Stall Enable)

This bit is used to select whether the Controller stalls SCL at the parity bit of the transmit data of I3C write transfer to avoid underrun of the transmit FIFO. However, when the transmit FIFO of the I3C Controller becomes empty, SCL stalling is performed regardless of the setting of this bit, it is not necessary to set this bit and it is prohibited.

APSE Bit (ACK/NACK Phase Stall Enable)

Determine the need to perform SCL stalling during the ACK/NACK phase based on the following criteria:

- It is necessary to set this bit when the I3C and I^2C Targets connected to the bus require preparation time to receive or transmit data.
- In legacy I^2C communication, if there is a possibility that the FIFO of the I3C Controller might underrun or overrun. It is not necessary to set this bit because SCL stalling is performed by FIFO Empty or Full regardless of the setting

of this bit.

- Other than legacy I2C communication, the FIFO of I3C Controller might underrun or overrun, and if SCL stalling is required in ACK phase, this bit can be set. However, it is necessary to build the software so that the FIFO does not underrun or overrun due to the interrupt generated according to the FIFO threshold setting (ICQBTCR, ICDBTCR, ICSQTCR).
- When I3C Controller responds ACK/NACK to IBI, it is not necessary to set this bit because ACK/NACK response can be set in advance by ICCR.HJC, ICTDATRm.CRRRJ, and ICTDATRm.TIRRJ.
- It is necessary to set this bit when the I3C Target connected to the bus requires preparation time to transmit data for Direct GET CCC.

35.2.20 Target Transmit/Receive Data Length Register (ICTDLR)

Address(es): RI3C0.ICTDLR 000E C0C0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DLGTH[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b31 to b16	DLGTH[15:0]	Data Length	Indicates the number of bytes to be transferred.	R/W

Note: This register is used in I3C Secondary Controller and I3C Target modes.

This register is used to set the number of bytes of data to be transmitted with the regular data transfer command.

35.2.21 Command Queue Register (ICCQR)

Address(es): RI3C0.ICCQR 000E C150h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The ICCQR register is a 32-bit write-only register used to enqueue Command Descriptors into the command queue. Command Descriptors should be written in order from lower (b31-b0) to higher (b63-b32).
Refer to section 35.3.1, Command Descriptor for details on Command Descriptors.

35.2.22 Response Queue Register (ICRQR)

Address(es): RI3C0.ICRQR 000E C154h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The ICRQR register is a 32-bit read-only register used to read the Response Descriptors generated in the response queue after a command has been processed.
Refer to section 35.3.2, Response Descriptor for details on Response Descriptors.

35.2.23 Transmit/Receive Data Register (ICDR)

Address(es): RI3C0.ICDR 000E C158h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The ICDR register is a 32-bit read/write register used to read data from the receive data buffer and write data to the transmit data buffer.

For Read Operation:

Receive data is stored in the receive data buffer every 4 bytes. If the number of data received is not an integer multiple of 4 bytes, the last 32 bits of the stored data will contain invalid bytes. Check the number of valid data bytes in the DATA_LENGTH field of the Response Descriptor.

For Write Operation:

Transmit data must be written to the ICDR register in 4-byte units. If the number of data to be transmitted is not an integer multiple of 4 bytes, add dummy data after the final data to make it 4 bytes before writing to the ICDR register. RI3C sends the number of bytes of data specified in the DATA_LENGTH field of the Command Descriptor.

35.2.24 IBI Queue Register (ICIQR)

Address(es): RI3C0.ICIQR 000E C17Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The ICIQR register is a 32-bit read-only register used to read an IBI Status Descriptor and IBI data from the IBI queue after an IBI has been received.
Refer to section 35.3.3, IBI Status Descriptor for details on IBI Status Descriptors.

35.2.25 Receive Status Queue Register (ICSQR)

Address(es): RI3C0.ICSQR 000E C180h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: This register is used in I3C Secondary Controller and I3C Target modes.

The ICSQR register is a 32-bit read-only register used to read an Receive Status Descriptor from the receive status queue.
Refer to section 35.3.4, Receive Status Descriptor for details on Receive Status Descriptors.

35.2.26 Queue Buffer Threshold Control Register (ICQBTCR)

Address(es): RI3C0.ICQBTCR 000E C190h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IQTH[7:0]								IDSS[7:0]							
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RQTH[7:0]								CQTH[7:0]							
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	CQTH[7:0]	Command Queue Empty Interrupt Threshold Setting	00h: Interrupt is generated when command queue is completely empty. 01h: Interrupt is generated when command queue has one free entry. Settings other than above are prohibited.	R/W
b15 to b8	RQTH[7:0]	Response Queue Full Interrupt Threshold Setting	00h: Interrupt is generated when response queue contains one entry (4 bytes). 01h: Interrupt is generated when response queue contains two entries (8 bytes). Settings other than above are prohibited.	R/W
b23 to b16	IDSS[7:0]	IBI Data Segment Size Setting*1	Set the maximum size of IBI data segments in 4-byte units. Setting range: 1 (4 bytes) to 5 (20 bytes)	R/W
b31 to b24	IQTH[7:0]	IBI Queue Empty/Full Interrupt Threshold Setting	• I3C Controller mode: 00h: Interrupt is generated when there is at least one outstanding IBI status. 01h: Interrupt is generated when there are two outstanding IBI status. Settings other than above are prohibited. • I3C Target mode: 00h: Interrupt is generated when IBI queue is completely empty. 01h: Interrupt is generated when IBI queue has one free entry. : 05h: Interrupt is generated when IBI queue has five free entries. Settings other than above are prohibited.	R/W

Note 1. These bits are used in I3C Primarily Controller and I3C Secondary Controller modes.

This register controls the interrupt generation thresholds for the command queue, response queue, and IBI queue.

CQTH[7:0] Bits (Command Queue Empty Interrupt Threshold Setting)

These bits control the minimum number of free command queue entries needed to generate the command queue empty interrupt (CMDI).

RQTH[7:0] Bits (Response Queue Full Interrupt Threshold Setting)

These bits control the minimum number of response queue entries needed to generate the response queue full interrupt (RESPI).

IDSS[7:0] Bits (IBI Data Segment Size Setting)

These bits specify the maximum size of IBI data segments in 4-byte units.

These bits allow the incoming IBI data to be sliced into multiple segments generating status individually, to support cut-

through readout of a long IBI payload data.

IQTH[7:0] Bits (IBI Queue Empty/Full Interrupt Threshold Setting)

(1) For I3C Controller mode (ICMMR.ACF flag = 1):

These bits control generation of the IBI queue empty/full interrupt (IBII), based on the value of the number of outstanding IBI status in the IBI queue.

Each IBI status entry can represent either the complete IBI payload (if the IBI payload size is 4 × IDSS[7:0] bytes or less), or a segment of the IBI payload (if the IBI payload size is more than 4 × IDSS[7:0] bytes).

(2) For I3C Target mode (ICMMR.ACF flag = 0):

These bits control the minimum number of free IBI data buffer entries needed to generate the IBI queue empty/full interrupt (IBII).

35.2.27 Data Buffer Threshold Control Register (ICDBTCR)

Address(es): RI3C0.ICDBTCR 000E C194h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	RSTH[2:0]			—	—	—	—	—	TSTH[2:0]		
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	RFTH[2:0]			—	—	—	—	—	TETH[2:0]		
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	TETH[2:0]	Transmit Data Empty Interrupt Threshold Setting	b2 b0 0 0 0: Interrupt is generated when the free level of the transmit data buffer reaches 2 stages (8 bytes). 0 0 1: Reserved Settings other than above are prohibited.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R
b10 to b8	RFTH[2:0]	Receive Data Full Interrupt Threshold Setting	b10 b8 0 0 0: Interrupt is generated when the fill level of the receive data buffer reaches 2 stages (5 to 8 bytes). 0 0 1: Reserved Settings other than above are prohibited.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R
b18 to b16	TSTH[2:0]	Transmit Start Threshold Setting *1	b18 b16 0 0 0: Transmission starts when the fill level of the transmit data buffer reaches 2 stages (5 to 8 bytes). 0 0 1: Reserved Settings other than above are prohibited.	R/W
b23 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R
b26 to b24	RSTH[2:0]	Receive Start Threshold Setting *1	b26 b24 0 0 0: Reception starts when the free level of the receive data buffer reaches 2 stages (8 bytes). 0 0 1: Reserved Settings other than above are prohibited.	R/W
b31 to b27	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. These bits are used in I3C Primarily Controller and I3C Secondary Controller modes.

The ICDBTCR register controls the interrupt trigger thresholds for the receive data buffer and the transmit data buffer.

TETH[2:0] Bits (Transmit Data Empty Interrupt Threshold Setting)

These bits specify the minimum number of free transmit data buffer stages needed to generate the transmit data empty interrupt (TXI).

RFTH[2:0] Bits (Receive Data Full Interrupt Threshold Setting)

These bits specify the minimum number of filled receive data buffer stages needed to generate the receive data full interrupt (RXI).

TSTH[2:0] Bits (Transmit Start Threshold Setting)

When preparing to initiate a Write Transfer on the I3C bus, RI3C waits until the indicated number of stages of data have been written to the transmit buffer.

RSTH[2:0] Bits (Receive Start Threshold Setting)

When preparing to initiate a Read Transfer on the I3C bus, RI3C waits until the receive buffer has the indicated number of free stages to receive the data bytes.

35.2.28 Receive Status Queue Threshold Control Register (ICSQTCR)

Address(es): RI3C0.ICSQTCR 000E C1C0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	SQTH[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	SQTH[7:0]	Receive Status Queue Full Interrupt Threshold Setting	00h: Interrupt is generated when receive status queue contains one entry (4 bytes). 01h: Interrupt is generated when receive status queue contains two entries (8 bytes). Settings other than above are prohibited.	R/W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This register is used in I3C Secondary Controller and I3C Target modes.

SQTH[7:0] Bits (Receive Status Queue Full Interrupt Threshold Setting)

These bits control the minimum number of receive status queue entries needed to generate the receive status queue full interrupt (RCVI).

35.2.29 Status Register 2 (ICSR2)

Address(es): RI3C0.ICSR2 000E C1D0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	TMOF	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	HDRXDF	STOP	START
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	START	START Condition Detection Flag	0: START condition is not detected. 1: START condition is detected.	R/(W) *1
b1	STOP	STOP Condition Detection Flag	0: STOP condition is not detected. 1: STOP condition is detected.	R/(W) *1
b2	HDRXDF	HDR Exit Pattern Detection Flag	0: HDR Exit Pattern is not detected. 1: HDR Exit Pattern is detected.	R/(W) *1
b19 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R
b20	TMOF	Timeout Detection Flag	0: Timeout is not detected. 1: Timeout is detected.	R/(W) *1
b31 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Only 0 can be written to this bit.

START Flag (START Condition Detection Flag)

[Setting condition]

- When a START condition (or a repeated START condition) is detected while the ICSER.STDE bit is 1.

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1.
- When a STOP condition is detected.
- When the RI3C is reset by setting the ICRCR.MRST bit to 1.
- When the internal status is reset by setting the ICRCR.ISRST bit to 1.

STOP Flag (STOP Condition Detection Flag)

[Setting condition]

- When a STOP condition is detected while the ICSER.SPDE bit is 1.

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1.
- When the RI3C is reset by setting the ICRCR.MRST bit to 1.
- When the internal status is reset by setting the ICRCR.ISRST bit to 1.

HDRXDF Flag (HDR Exit Pattern Detection Flag)

[Setting condition]

- When a HDR Exit Pattern is detected while the ICSER.HDRXDE bit is 1.

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1.

TMOF Flag (Timeout Detection Flag)

[Setting condition]

- All of the following 1 to 3 are satisfied.
  1. The ICSER.TMOE bit is 1 (timeout detection is enabled).
  2. When the Controller mode or the received Target address matches the Target address n in Target mode.
  3. When the SCL00 line state remains unchanged for the period specified by ICTOR register.

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1.

35.2.30 Status Detection Enable Register (ICSER)

Address(es): RI3C0.ICSER 000E C1D4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	TMOE	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	HDRXDE	SPDE	STDE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	STDE	START Condition Detection Enable	0: START condition detection is disabled. 1: START condition detection is enabled.	R/W
b1	SPDE	STOP Condition Detection Enable	0: STOP condition detection is disabled. 1: STOP condition detection is enabled.	R/W
b2	HDRXDE	HDR Exit Pattern Detection Enable	0: HDR Exit Pattern detection is disabled. 1: HDR Exit Pattern detection is enabled.	R/W
b19 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R
b20	TMOE	Timeout Detection Enable	0: Timeout detection is disabled. 1: Timeout detection is enabled.	R/W
b31 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R

This register is used to enable or disable detection of each status. When the status enabled by this register is detected, the corresponding flag in the ICSR2 register is set to 1.

35.2.31 Status Interrupt Enable Register (ICSIER)

Address(es): RI3C0.ICSIER 000E C1D8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	TMOIE	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	HDRXIE	SPIE	STIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	STIE	START Condition Detection Interrupt Enable	0: START condition detection interrupt is disabled. 1: START condition detection interrupt is enabled.	R/W
b1	SPIE	STOP Condition Detection Interrupt Enable	0: STOP condition detection interrupt is disabled. 1: STOP condition detection interrupt is enabled.	R/W
b2	HDRXIE	HDR Exit Pattern Detection Interrupt Enable	0: HDR Exit Pattern detection interrupt is disabled. 1: HDR Exit Pattern detection interrupt is enabled.	R/W
b19 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b20	TMOIE	Timeout Detection Interrupt Enable	0: Timeout detection interrupt is disabled. 1: Timeout detection interrupt is enabled.	R/W
b31 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R

This register is used to enable or disable interrupts for each status. When a flag in the ICSR2 register becomes 1 and the corresponding interrupt enable bit is 1, an interrupt request is output.

35.2.32 Communication Status Register (ICCSR)

Address(es): RI3C0.ICCSR 000E C1E0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	SQFF	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	DTEF	—	—	—	DTAF	RQFF	CQEF	IQEFF	RDRF	TDRE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TDRE	Transmit Data Empty Flag	0: The free level of the transmit data buffer is less than the threshold specified by the ICDBTCR.TETH[2:0] bits. 1: The free level of the transmit data buffer is equal to or more than the threshold specified by the TETH[2:0] bits.	R/(W) *1
b1	RDRF	Receive Data Full Flag	0: The fill level of the receive data buffer is less than the threshold specified by the ICDBTCR.RFTH[2:0] bits 1: The fill level of the receive data buffer is equal to or more than the threshold specified by the RFTH[2:0] bits.	R/(W) *1
b2	IQEFF	IBI Queue Empty/Full Flag	• In I3C Controller mode 0: The number of IBI statuses in IBI queue is less than the threshold specified by the ICQBTCR.IQTH[7:0] bits. 1: The number of IBI statuses in IBI queue is equal to or more than the threshold specified by the IQTH[7:0] bits. • In I3C Target mode and if the IQTH[7:0] bits are 00h: 0: IBI queue contains entry. 1: IBI queue contains no entries. • In I3C Target mode and if the IQTH[7:0] bits are not 00h: 0: The number of free IBI queue entries is less than the threshold specified by the IQTH[7:0] bits. 1: The number of free IBI queue entries is equal to or more than the threshold specified by the IQTH[7:0] bits.	R/(W) *1
b3	CQEF	Command Queue Empty Flag	• If the ICQBTCR.CQTH[7:0] bits are 00h: 0: Command queue contains data. 1: Command queue contains no data. • If the CQTH[7:0] bits are not 00h: 0: The number of free command queue entries is less than the threshold specified by the CQTH[7:0] bits. 1: The number of free command queue entries is equal to or more than the threshold specified by the CQTH[7:0] bits.	R/(W) *1
b4	RQFF	Response Queue Full Flag	0: The number of response queue entries is less than the threshold specified by the ICQBTCR.RQTH[7:0] bits. 1: The number of response queue entries is equal to or more than the threshold specified by the RQTH[7:0] bits.	R/(W) *1
b5	DTAF	Data Transfer Abort Flag	0: Data transfer abort does not occur. 1: Data transfer abort occurs.	R/(W) *1
b8 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R
b9	DTEF	Data Transfer Error Flag	0: Data transfer error does not occur. 1: Data transfer error occurs.*2	R/(W) *1
b19 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R
b20	SQFF	Receive Status Queue Full Flag *3	0: The number of receive status queue entries is less than the threshold specified by the ICSQTCR.SQTH[7:0] bits. 1: The number of receive status queue entries is equal to or more than the threshold specified by the SQTH[7:0] bits.	R/(W) *1
b31 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. To clear this flag, confirm that the flag is 1 and then set it to 0.

Note 2. The Error type for this error is available in the Response or Receive Status Descriptor structure corresponding to the transfer or command.
Note 3. This bit is used in I3C Secondary Controller and I3C Target modes.

TDRE Flag (Transmit Data Empty Flag)

[Setting condition]

- When the free level of the transmit data buffer is equal to or more than the threshold specified by the ICDBTCR.TETH[2:0] bits while the ICCSER.TDE bit is 1 (detection of transmit data empty is enabled).

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1.
- On completion of the last write access to the transmit data buffer by DMAC/DTC.
- When the free level of the transmit data buffer is less than the threshold specified by the TETH[2:0] bits.

RDRF Flag (Receive Data Full Flag)

[Setting condition]

- When the fill level of the receive data buffer is equal to or more than the threshold specified by the ICDBTCR.RFTH[2:0] bits while the ICCSER.RDE bit is 1 (detection of receive data full is enabled).

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1.
- On completion of the last read access to the receive data buffer by DMAC/DTC.
- When the fill level of the receive data buffer is less than the threshold specified by the RFTH[2:0] bits.

IQEFF Flag (IBI Queue Empty/Full Flag)

[Setting condition]

When any of the following conditions is satisfied while the ICCSER.IQEFDE bit is 1 (detection of IBI queue empty/full is enabled).

In I3C Controller mode:

- When the number of the IBI queue entries is equal to or more than the threshold specified by the ICQBTCR.IQTH[7:0] bits.

In I3C Target mode:

- When the IBI queue is completely empty (when the IQTH[7:0] bits are 00h).
- When the number of the free IBI queue entries is equal to or more than the threshold specified by the IQTH[7:0] bits (when the IQTH[7:0] bits are other than 00h).

[Clearing condition]

In I3C Controller mode:

- When 0 is written to this flag after confirming that it is 1.
- On completion of the last read access to the IBI queue by DMAC/DTC.
- When the number of the IBI queue entries is less than the threshold specified by the IQTH[7:0] bits.

In I3C Target mode:

- When 0 is written to this flag after confirming that it is 1.
- On completion of the last write access to the IBI queue by DMAC/DTC.
- When the IBI queue is not completely empty (when the IQTH[7:0] bits are 00h).
- When the number of the free IBI queue entries is less than the threshold specified by the IQTH[7:0] bits (when the IQTH[7:0] bits are other than 00h).

CQEF Flag (Command Queue Empty Flag)

[Setting condition]

When any of the following conditions is satisfied while the ICCSER.CQEDE bit is 1 (detection of command queue empty is enabled).

- When the command queue is completely empty (when the ICQBTCR.CQTH[7:0] bits are 00h).
- When the number of the free command queue entries is equal to or more than the threshold specified by the CQTH[7:0] bits (when the CQTH[7:0] bits are other than 00h).

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1.
- On completion of the last write access to the command queue by DMAC/DTC.
- When the command queue is not completely empty (when the CQTH[7:0] bits are 00h).
- When the number of the free command queue entries is less than the threshold specified by the CQTH[7:0] bits (when the CQTH[7:0] bits are other than 00h).

RQFF Flag (Response Queue Full Flag)

[Setting condition]

- When the number of the response queue entries is more than the threshold specified by the ICQBTCR.RQTH[7:0] bits while the ICCSER.RQFDE bit is 1 (detection of response queue full is enabled).

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1.
- On completion of the last read access to the response queue by DMAC/DTC.
- When the number of the response queue entries is equal to or less than the threshold specified by the RQTH[7:0] bits.

DTAF Flag (Data Transfer Abort Flag)

[Setting condition]

- When any data transfer is aborted while the ICCSER.DTADE bit is 1 (detection of data transfer abort is enabled).

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1.

DTEF Flag (Data Transfer Error Flag)

[Setting condition]

- When any data transfer error occurs on the I3C bus while the ICCSER.DTEDE bit is 1 (detection of data transfer error is enabled).

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1.

SQFF Flag (Receive Status Queue Full Flag)

[Setting condition]

- When the number of the receive status queue entries is equal to or more than the threshold specified by the ICSQTCR.SQTH[7:0] bits while the ICCSER.SQFDE bit is 1 (detection of receive status queue full is enabled).

[Clearing condition]

- When 0 is written to this flag after confirming that it is 1.
- On completion of the last read access to the receive status queue by DMAC/DTC.
- When the number of the receive status queue entries is less than the threshold specified by the SQTH[7:0] bits.

35.2.33 Communication Status Detection Enable Register (ICCSER)

Address(es): RI3C0.ICCSER 000E C1E4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	SQFDE	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	DTEDE	—	—	—	DTADE	RQFDE	CQEDE	IQEFDE	RDE	TDE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TDE	Transmit Data Empty Detection Enable	0: Detection of transmit data empty is disabled. 1: Detection of transmit data empty is enabled.	R/W
b1	RDE	Receive Data Full Detection Enable	0: Detection of receive data full is disabled. 1: Detection of receive data full is enabled.	R/W
b2	IQEFDE	IBI Queue Empty/Full Detection Enable	0: Detection of IBI queue empty/full is disabled. 1: Detection of IBI queue empty/full is enabled.	R/W
b3	CQEDE	Command Queue Empty Detection Enable	0: Detection of command queue empty is disabled. 1: Detection of command queue empty is enabled.	R/W
b4	RQFDE	Response Queue Full Detection Enable	0: Detection of response queue full is disabled. 1: Detection of response queue full is enabled.	R/W
b5	DTADE	Data Transfer Abort Detection Enable	0: Detection of data transfer abort is disabled. 1: Detection of data transfer abort is enabled.	R/W
b8 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R
b9	DTEDE	Data Transfer Error Detection Enable	0: Detection of data transfer error is disabled. 1: Detection of data transfer error is enabled.	R/W
b19 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R
b20	SQFDE	Receive Status Queue Full Detection Enable*1	0: Detection of receive status queue full is disabled. 1: Detection of receive status queue full is enabled.	R/W
b31 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. This bit is used in I3C Secondary Controller and I3C Target modes.

This register is used to enable or disable detection of each status. When the status enabled by this register is detected, the corresponding flag in the ICCSR register is set to 1.

35.2.34 Communication Status Interrupt Enable Register (ICCSIER)

Address(es): RI3C0.ICCSIER 000E C1E8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	SQFIE	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	DTEIE	—	—	—	DTAIE	RQFIE	CQEIE	IQEFIE	RIE	TIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TIE	Transmit Data Empty Interrupt Enable	0: Transmit data empty interrupt is disabled. 1: Transmit data empty interrupt is enabled.	R/W
b1	RIE	Receive Data Full Interrupt Enable	0: Receive data full interrupt is disabled. 1: Receive data full interrupt is enabled.	R/W
b2	IQEFIE	IBI Queue Empty/Full Interrupt Enable	0: IBI queue empty/full interrupt is disabled. 1: IBI queue empty/full interrupt is enabled.	R/W
b3	CQEIE	Command Queue Empty Interrupt Enable	0: Command queue empty interrupt is disabled. 1: Command queue empty interrupt is enabled.	R/W
b4	RQFIE	Response Queue Full Interrupt Enable	0: Response queue full interrupt is disabled. 1: Response queue full interrupt is enabled.	R/W
b5	DTAIE	Data Transfer Abort Interrupt Enable	0: Data transfer abort interrupt is disabled. 1: Data transfer abort interrupt is enabled.	R/W
b8 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R
b9	DTEIE	Data Transfer Error Interrupt Enable	0: Data transfer error interrupt is disabled. 1: Data transfer error interrupt is enabled.	R/W
b19 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R
b20	SQFIE	Receive Status Queue Full Interrupt Enable*1	0: Receive status queue full interrupt is disabled. 1: Receive status queue full interrupt is enabled.	R/W
b31 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. This bit is used in I3C Secondary Controller and I3C Target modes.

This register is used to enable or disable interrupts for each status. When a flag in the ICCSR register becomes 1 and the corresponding interrupt enable bit is 1, an interrupt request is generated.

35.2.35 Bus Status Register (ICBSR)

Address(es): RI3C0.ICBSR 000E C210h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	BIDL	BAVL	BFREE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BFREE	Bus Free Flag	0: Have not detected bus free. 1: Have detected bus free.	R
b1	BAVL	Bus Available Detection Flag	0: Have not detected bus available. 1: Have detected bus available.	R
b2	BIDL	Bus Idle Detection Flag	0: Have not detected bus idle. 1: Have detected bus idle.	R
b31 to b3	—	Reserved	These bits are read as 0.	R

BFREE Flag (Bus Free Flag)

This flag indicates whether the I3C bus is occupied (bus busy) or released (bus free).

[Setting condition]

- When the number of cycles set in the ICBFTR register has elapsed when the SCL00 and SDA00 lines are high after a STOP condition is detected.
- When the number of cycles set in the ICBFTR register has elapsed when the SCL00 and SDA00 lines are high after setting the ICCR.ICE bit to 1.

[Clearing condition]

- When SCL00 line and SDA00 line are other than high.
- When the ICCR.ICE bit is set to 0.

BAVL Flag (Bus Available Detection Flag)

The Bus Available condition is a period during which the Bus Free condition is sustained continuously for a duration of at least tAVAL (refer to section 49.4.5.14, RI3C). A Target can only issue a START Request (for an IBI or for a CRR) after a Bus Available condition.

[Setting condition]

- When the number of cycles set in the ICBATR register has elapsed when the SCL00 and SDA00 lines are high after a STOP condition is detected.
- When the number of cycles set in the ICBATR register has elapsed when the SCL00 and SDA00 lines are high after setting the ICCR.ICE bit to 1.

[Clearing condition]

- When SCL00 line and SDA00 line are other than high.
- When the ICCR.ICE bit is set to 0.

BIDL Flag (Bus Idle Detection Flag)

The I3C Bus Idle condition is in order to help ensure bus stability during Hot-Join events. The Bus Idle condition is a period during which the Bus Available condition is sustained continuously for a duration of at least tIDLE (refer to section 49.4.5.14, RI3C).

If a Hot-Join Device is powered up onto the I3C bus at the same time as the Primarily Controller, then the Hot-Join Device may pull SDA00 line Low after 1 ms if (1) the Primarily Controller has SCL00 line and SDA00 line pulled up, and (2) the Controller does not act on the I3C bus within the same Idle period.

[Setting condition]

- When the number of cycles set in the ICBITR register has elapsed when the SCL00 and SDA00 lines are high after a STOP condition is detected.
- When the number of cycles set in the ICBITR register has elapsed when the SCL00 and SDA00 lines are high after setting the ICCR.ICE bit to 1.

[Clearing condition]

- When SCL00 line and SDA00 line are other than High.
- When the ICCR.ICE bit is set to 0.

35.2.36 Target Device Address Table Register m (ICTDATRm) (m = 0 to 3)

Address(es): RI3C0.ICTDATR0 000E C224h, RI3C0.ICTDATR1 000E C22Ch, RI3C0.ICTDATR2 000E C234h, RI3C0.ICTDATR3 000E C23Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TYPE	NACKRC[1:0]		—	—	—	—	—	DADR[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	IBITSE	CRRRJ	TIRRJ	IBIPL	—	—	—	—	—	SADR[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	SADR[6:0]	Static Address	I3C/I^{2}C static address	R/W
b11 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R
b12	IBIPL	IBI Payload	0: IBIs from this device do not carry data payload. 1: IBIs from this device carry data payload.	R/W
b13	TIRRJ	Target Interrupt Request Reject	0: RI3C ACKs Target Interrupt Requests from this device. 1: RI3C NACKs Target Interrupt Requests from this device and automatically sends DISEC CCC.	R/W
b14	CRRRJ	Controller Role Request Reject	0: RI3C ACKs Controller Role Requests from this device. 1: RI3C NACKs Controller Role Requests from this device and automatically sends DISEC CCC.	R/W
b15	IBITSE	IBI Timestamp Enable	0: RI3C does not timestamp IBIs from this device with Controller timestamps. 1: RI3C timestamps IBIs from this device with Controller timestamps.	R/W
b23 to b16	DADR[7:0]	Dynamic Address	DADR[6:0] bits: Dynamic Address for this device DADR[7] bit: Parity bit	R/W
b28 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R
b30, b29	NACKRC[1:0]	NACK Retry Count Setting	0h: No retries on NACK 1h to 3h: Up to 1 to 3 retries on NACK	R/W
b31	TYPE	Device Type	0: I3C device 1: I^{2}C device	R/W

Note: This register is used in I3C Primarily Controller and I3C Secondary Controller modes.

The Device Address Table (DAT) stores the device addresses and control informations of Target devices attached to the I3C bus and is implemented in RI3C as the ICTDATRm registers. The Command Descriptor of a command that requires this register informations has a DEV_INDEX field, and the register is selected by the field.

IBIPL Bit (IBI Payload)

This bit indicates whether IBIs from the corresponding device have a data payload. This bit reflects the ICTDCTRm.IBIPL bit (BCR[2]: IBI Payload bit in the Bus Characteristics Register).
During IBI handling for this device, the Controller shall use this bit to determine whether or not to drive reception of the IBI data payload. Data continuation is indicated by the T-Bit.

TIRRJ Bit (Target Interrupt Request Reject)

This bit controls whether RI3C, when operating in the Active Controller role, will accept or reject Target Interrupt Requests (TIR) from the Target device indicated by this register.

CRRRJ Bit (Controller Role Request Reject)

This bit controls whether RI3C, when operating in the Active Controller role, will accept or reject Controller Role Requests (CRR) from the Secondary Controller indicated by this register.

IBITSE Bit (IBI Timestamp Enable)

This bit enables or disables IBI timestamping for a specific device.

Note: The IBI Status Descriptor for each IBI event indicates whether or not the individual IBI event was actually timestamped. Set to 0 except for Async Mode 0 and Async Mode 1 of Timing Control Mode.

NACKRC[1:0] Bits (NACK Retry Count Setting)

These bits set the number of retries when a NACK response is received from the Target for the transaction set in the Command Descriptor.

Note: When ENTDAA CCC is executed by address assignment command (refer to section 35.3.1.1, Address Assignment Command), the setting of this bit is ignored and the transaction ends when NACK is received once.

35.2.37 Extended Target Device Address Table Register (ICEDATR)

Address(es): RI3C0.ICEDATR 000E C2A0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TYPE	NACKRC[1:0]		—	—	—	—	—	DADR[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	SADR[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	SADR[6:0]	Static Address	I3C/I^{2}C static address	R/W
b15 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R
b23 to b16	DADR[7:0]	Dynamic Address	DADR[6:0] bits: Dynamic Address for this device DADR[7] bit: Parity bit	R/W
b28 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R
b30, b29	NACKRC[1:0]	NACK Retry Count Setting	0h: No retries on NACK 1h to 3h: Up to 1 to 3 retries on NACK	R/W
b31	TYPE	Device Type	0: I3C device 1: I^{2}C device	R/W

Note: This register is used in I3C Primarily Controller and I3C Secondary Controller modes.

35.2.38 Device Address Register 0 (ICDAR0)

Address(es): RI3C0.ICDAR0 000E C2B0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	DADR[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	IBIPL	—	—	SADR[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	SADR[9:0]	Static Address*2	Set a static address.	R/W
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R
b12	IBIPL	IBI Payload*1	This bit is the mirror bit of the ICDCTR.IBIPL. 0: IBIs from this device do not carry a data payload. 1: IBIs from this device carry a data payload.	R/W
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R
b22 to b16	DADR[6:0]	Dynamic Address*2	Dynamic Address is set.	R/W
b31 to b23	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: Do not write to this register when RI3C is Primary Controller.
Note 1. This bit is used in I3C Secondary Controller and I3C Target modes.
Note 2. These bits are used in I3C Secondary Controller and I3C Target modes.

The Device Characteristics Table (DCT) is a register set that stores the Device characteristics (PID, BCR, DCR) and assigned Dynamic Address of each device on the I3C bus that participates in the Dynamic Address Allocation (ENTDAA) procedure.
The ICDAR0 register is used to store the values of the assigned Dynamic Address.

SADR[9:0] Bits (Static Address)

Set the static address in Target mode to the lower 7 bits.
The upper 3 bits are reserved. Set them to 0.

IBIPL Bit (IBI Payload)

Indicates whether IBIs from this Device have a Data Payload. This bit reflects the IBI Payload bit in the Device's Bus Characteristics Register (BCR).
During IBI handling for this Device, the Controller shall use this bit to determine whether or not to drive reception of the IBI Data Payload. Data continuation is indicated by the T-Bit.

DADR[6:0] Bits (Dynamic Address)

These bits store the Dynamic Address assigned in Target mode.
These bits are updated when one any of the following events occurs.

- When writing Dynamic Address value.
- When Target Address is equal to its own Static Address in receiving SETDASA CCC (Direct), these bits are updated to the value of the subsequent Dynamic Address.
- When Dynamic Address Assignment procedure that starts by receiving ENTDAA CCC (Broadcast) is established.
- When receiving RSTDAA CCC (Broadcast), all bits are set to 0.
- When Target Address is equal to its own Dynamic Address in receiving RSTDAA CCC (Direct), all bits are set to 0.

- When Target Address is equal to its own Dynamic Address in receiving SETNEWDA CCC (Direct), these bits are updated to the value of the subsequent new Dynamic Address.
- When receiving SETAASA CCC (Broadcast), these bits are updated to the value of the SADR[6:0] bits.

35.2.39 Target Device Characteristics Table Register m (ICTDCTRm) (m = 0 to 3)

Address(es): RI3C0.ICTDCTR0 000E C2D0h, RI3C0.ICTDCTR1 000E C2D4h, RI3C0.ICTDCTR2 000E C2D8h, RI3C0.ICTDCTR3 000E C2DCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ROLE[1:0]		—	—	OFLC	IBIPL	IBIRQC	LIMIT	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	LIMIT	Max Data Speed Limitation	0: No limitation 1: Limitation	R/W
b9	IBIRQC	IBI Request Capable	0: Not capable 1: Capable	R/W
b10	IBIPL	IBI Payload	0: No data byte follows the accepted IBI. 1: One or more data bytes follow the accepted IBI. Data byte continuation is indicated by T-Bit.	R/W
b11	OFLC	Offline Capable	0: Device will always respond to I3C bus commands. 1: Device will not always respond to I3C bus commands.	R/W
b13, b12	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b15, b14	ROLE[1:0]	Device Role Setting	b15 b14 0 0: I3C Target 0 1: I3C Controller capable*1 Settings other than above are prohibited.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This register is used in I3C Primarily Controller and I3C Secondary Controller modes.
Note 1. For an I3C Device acting as I3C Primary Controller, the ROLE[1:0] bits will contain the value 01b.

The Device Characteristics Table (DCT) is a register set that stores the Device characteristics (PID, BCR, DCR) and assigned Dynamic Address of each device on the I3C bus that participates in the Dynamic Address Allocation (ENTDAA) procedure.
The ICTDCTRm register contains the value of the Bus Characteristic Register (BCR) of the Target device captured by the RI3C during Controller operation.
This register is updated when the Bus Characteristics Register (BCR) is received from the device in the Dynamic Address Allocation procedure starting by receiving the ENTDAA CCC (Broadcast).

LIMIT Bit (Max Data Speed Limitation)

This bit corresponds to bit 0 in the BCR (BCR[0]).
Controller shall use the GETMXDS CCC to interrogate the Target for specific limitation.

IBIRQC Bit (IBI Request Capable)

This bit corresponds to bit 1 in the BCR (BCR[1]).

IBIPL Bit (IBI Payload)

This bit corresponds to bit 2 in the BCR (BCR[2]).

OFLC Bit (Offline Capable)

This bit corresponds to bit 3 in the BCR (BCR[3]).
Offline capable devices retain the Dynamic Address.

ROLE[1:0] Bits (Device Role Setting)

This bit corresponds to bits 7 and 6 in the BCR (BCR[7:6]).

35.2.40 Device Characteristics Table Register (ICDCTR)

Address(es): RI3C0.ICDCTR 000E C320h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ROLE[1:0]		—	—	OFLC	IBIPL	IBIRQC	LIMIT	DCR[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	DCR[7:0]	Device Characteristics	255 available codes for describing the type of sensor, or device. Examples: Accelerometer, gyroscope, composite devices Default value is 00h: Generic device	R/W
b8	LIMIT	Max Data Speed Limitation*1	0: No limitation 1: Limitation	R/W
b9	IBIRQC	IBI Request Capable	0: Not capable 1: Capable	R/W
b10	IBIPL	IBI Payload	0: No data byte follows the accepted IBI. 1: One or more data bytes follow the accepted IBI. Data byte continuation is indicated by T-Bit.	R/W
b11	OFLC	Offline Capable*2	0: Device will always respond to I3C bus commands. 1: Device will not always respond to I3C bus commands.	R/W
b13, b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15, b14	ROLE[1:0]	Device Role Setting	b15 b14 0 0: I3C Target 0 1: I3C Controller capable*3 Settings other than above are prohibited.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This register is used in I3C Secondary Controller and I3C Target modes.
Note 1. Controller shall use the GETMXDS CCC to interrogate the Target for specific limitation.
Note 2. Offline Capable Devices retain the Dynamic Address.
Note 3. For an I3C Device acting as I3C Primarily Controller, the ROLE[1:0] bits will contain the value 01b.

The Device Characteristics Table (DCT) is a register set that stores the Device characteristics (PID, BCR, DCR) and assigned Dynamic Address of each device on the I3C bus that participates in the Dynamic Address Allocation (ENTDAA) procedure.
The ICDCTR register is used to store the values of the RI3C Bus Characteristics Register (BCR) and Device Characteristics Register (DCR). The BCR consists of the ROLE[1:0], OFLC, IBIPL, IBIRQC, and LIMIT bits.

DCR[7:0] Bits (Device Characteristics)

Each I3C device that is connected to the I3C bus has an associated Device Characteristics Register (DCR). This register describes the I3C compliant device type such as accelerometer and gyroscope, for use in Dynamic Address assignment and Common Command Codes.

LIMIT Bit (Max Data Speed Limitation)

This bit corresponds to bit 0 in the BCR (BCR[0]).

IBIRQC Bit (IBI Request Capable)

This bit corresponds to bit 1 in the BCR (BCR[1]).

IBIPL Bit (IBI Payload)

This bit corresponds to bit 2 in the BCR (BCR[2]).

OFLC Bit (Offline Capable)

This bit corresponds to bit 3 in the BCR (BCR[3]).

ROLE[1:0] Bits (Device Role Setting)

This bit corresponds to bits 7 and 6 in the BCR (BCR[7:6]).

When RI3C is in Target mode and issues IBI by Command Descriptor, set the BCR as follows:

(1) Target Interrupt Request: No IBI Payload follow the accepted IBI:

- IBIRQC = 1
- IBIPL = 0

Note: Set DATA_LENGTH field of Command Descriptor to 0000h.

(2) Target Interrupt Request: IBI Payload follow the accepted IBI:

- IBIRQC = 1
- IBIPL = 1

Note: Set DATA_LENGTH field of Command Descriptor to any value.

(3) Controller Role Request:

- IBIRQC = 1
- ROLE[1:0] = 01b

(4) Hot-Join Event:

- IBIRQC = 1

When RI3C is in Target mode and receives CCC from I3C Controller, it performs the following operations according to the setting of the BCR:

- When the IBIPL bit is 1, RI3C sends the value set in the ICMRLR.IBIPL[7:0] bits as the 3rd byte data to GETMRL CCC from I3C Controller.
- When the LIMIT bit is 0, RI3C NACKs GETMXDS CCC from I3C Controller.
- When the LIMIT bit is 1, RI3C ACKs GETMXDS CCC from I3C Controller and sends data set in the ICMWSR, ICMRSR, and ICMTTR registers.

35.2.41 Provisioned ID Low Register (ICPIDLR)

Address(es): RI3C0.ICPIDLR 000E C324h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: This register is used in I3C Secondary Controller and I3C Target modes.

The Device Characteristics Table (DCT) is a register set that stores the Device characteristics (PID, BCR, DCR) and assigned Dynamic Address of each device on the I3C bus that participates in the Dynamic Address Allocation (ENTDAA) procedure.
This register stores the lower 16 bits of the provisioned ID (b15 to b0 of the PID).
The upper 16 bits are reserved. These bits are read as 0 and the write value should be 0.

35.2.42 Provisioned ID High Register (ICPIDHR)

Address(es): RI3C0.ICPIDHR 000E C328h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: This register is used in I3C Secondary Controller and I3C Target modes.

The Device Characteristics Table (DCT) is a register set that stores the Device characteristics (PID, BCR, DCR) and assigned Dynamic Address of each device on the I3C bus that participates in the Dynamic Address Allocation (ENTDAA) procedure.
This register stores the upper 32 bits of the provisioned ID (b47 to b16 of the PID).

35.2.43 Device Address Monitor Register 0 (ICDAMR0)

Address(es): RI3C0.ICDAMR0 000E C330h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DAV	SAV	—	—	—	—	TADR[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	Reserved	These bits are read as 0.	R
b25 to b16	TADR[9:0]	Target Address*1	Target address is set.	R
b29 to b26	—	Reserved	These bits are read as 0.	R
b30	SAV	Static Address Valid Flag*2	0: Static address is disabled. 1: Static address is enabled.	R
b31	DAV	Dynamic Address Valid Flag*2	0: Dynamic Address is disabled. 1: Dynamic Address is enabled.	R

Note 1. These bits are used in I3C Secondary Controller and I3C Target modes.
Note 2. This bit is used in I3C Secondary Controller and I3C Target modes.

TADR[9:0] Bits (Target Address)

The TADR[9:0] bits indicate a valid Target address.

- When the DAV flag is 1:
  The TADR[9:7] bits are 000b.
  The TADR[6:0] bits are the ICDAR0.DADR[6:0] bits.
- When the SAV flag is 1:
  The TADR[9:7] bits are 000b.
  The TADR[6:0] bits are the ICDAR0.SADR[6:0] bits.

SAV Flag (Static Address Valid Flag)

[Setting condition]

- All of the following are satisfied.
  1. The ICTCR.TA0DE bit is 1 (detection of Target address 0 is enabled).
  2. The DAV flag is 0 (Dynamic Address is disabled).
  3. The ICDAR0.SADR[6:0] bits are not 0000000b

[Clearing condition]

- [Setting condition] is not satisfied.

DAV Flag (Dynamic Address Valid Flag)

[Setting condition]

- All of the following are satisfied.
  1. The ICTCR.TA0DE bit is 1 (detection of Target address 0 is enabled).
  2. The ICDAR0.DADR[6:0] bits are not 0000000b.

[Clearing condition]

- [Setting condition] is not satisfied.

35.2.44 Target Event Register (ICTEVR)

Address(es): RI3C0.ICTEVR 000E C350h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	ENHJ	—	ENCR	ENINT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ENINT	Target Interrupt Request Enable Flag	0: Target-initiated interrupts is disabled by the Controller. 1: Target-initiated interrupts is enabled by the Controller.	R/W
b1	ENCR	Controller Role Request Enable Flag	0: Controller Role Requests from Secondary Controllers is disabled by the Active Controller. 1: Controller Role Requests from Secondary Controllers is enabled by the Active Controller.	R/W
b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b3	ENHJ	Hot-Join Event Enable Flag	0: Target-initiated Hot-Join is disabled by the Controller. 1: Target-initiated Hot-Join is enabled by the Controller.	R/W
b31 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This register is used in I3C Secondary Controller and I3C Target modes.

ENINT Flag (Target Interrupt Request Enable Flag)

This flag allows the Controller to control when Target-initiated interrupts are allowed on the I3C bus.

[Setting conditions]

- When writing 1.
- When receiving ENEC CCC (Broadcast) with the ENINT field set to 1.
- When receiving ENEC CCC (Direct) with the ENINT field set to 1 and with its own Target address.

[Clearing conditions]

- When writing 0.
- When receiving DISEC CCC (Broadcast) with the DISINT field set to 1.
- When receiving DISEC CCC (Direct) with the DISINT field set to 1 and with its own Target address.

ENCR Flag (Controller Role Request Enable Flag)

This flag allows the Active Controller to control when Controller Role Requests from Secondary Controller are allowed on the I3C bus.

[Setting conditions]

- When writing 1.
- When receiving ENEC CCC (Broadcast) with the ENCR field set to 1.
- When receiving ENEC CCC (Direct) with the ENCR field set to 1 and with its own Target address.

[Clearing conditions]

- When writing 0.
- When receiving DISEC CCC (Broadcast) with the DISCR field set to 1.
- When receiving DISEC CCC (Direct) with the DISCR field set to 1 and with its own Target address.

ENHJ Flag (Hot-Join Event Enable Flag)

This flag allows the Controller to control when Target-initiated Hot-Join is allowed on the I3C bus.

[Setting conditions]

- When writing 1.
- When receiving ENEC CCC (Broadcast) with the ENHJ field set to 1.
- When receiving ENEC CCC (Direct) with the ENHJ field set to 1 and with its own Target address.

[Clearing conditions]

- When writing 0.
- When receiving DISEC CCC (Broadcast) with the DISHJ field set to 1.
- When receiving DISEC CCC (Direct) with the DISHJ field set to 1 and with its own Target address.

35.2.45 Activity State Register (ICASR)

Address(es): RI3C0.ICASR 000E C354h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	ENTAS3	ENTAS2	ENTAS1	ENTAS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ENTAS0	Activity State 0 Flag*1	0: Other than activity state 0 1: Activity state 0	R/W
b1	ENTAS1	Activity State 1 Flag*1	0: Other than activity state 1 1: Activity state 1	R/W
b2	ENTAS2	Activity State 2 Flag*1	0: Other than activity state 2 1: Activity state 2	R/W
b3	ENTAS3	Activity State 3 Flag*1	0: Other than activity state 3 1: Activity state 3	R/W
b31 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This register is used in I3C Secondary Controller and I3C Target modes.
Note 1. Do not set multiple bits to 1 at the same time.

ENTAS0 Flag (Activity State 0 Flag)

[Setting conditions]

- When writing 1.
- When receiving ENTAS0 CCC (Broadcast).
- When receiving ENTAS0 CCC (Direct) with its own Target address.

[Clearing conditions]

- When writing 0.
- When receiving ENTAS1 CCC (Broadcast), ENTAS2 CCC (Broadcast), or ENTAS3 CCC (Broadcast).
- When receiving ENTAS1 CCC (Direct), ENTAS2 CCC (Direct), or ENTAS3 CCC (Direct) with its own Target address.

ENTAS1 Flag (Activity State 1 Flag)

[Setting conditions]

- When writing 1.
- When receiving ENTAS1 CCC (Broadcast).
- When receiving ENTAS1 CCC (Direct) with its own Target address.

[Clearing conditions]

- When writing 0.
- When receiving ENTAS0 CCC (Broadcast), ENTAS2 CCC (Broadcast), or ENTAS3 CCC (Broadcast).
- When receiving ENTAS0 CCC (Direct), ENTAS2 CCC (Direct), or ENTAS3 CCC (Direct) with its own Target address.

ENTAS2 Flag (Activity State 2 Flag)

[Setting conditions]

- When writing 1.
- When receiving ENTAS2 CCC (Broadcast).
- When receiving ENTAS2 CCC (Direct) with its own Target address.

[Clearing conditions]

- When writing 0.
- When receiving ENTAS0 CCC (Broadcast), ENTAS1 CCC (Broadcast), or ENTAS3 CCC (Broadcast).
- When receiving ENTAS0 CCC (Direct), ENTAS1 CCC (Direct), or ENTAS3 CCC (Direct) with its own Target address.

ENTAS3 Flag (Activity State 3 Flag)

[Setting conditions]

- When writing 1.
- When receiving ENTAS3 CCC (Broadcast).
- When receiving ENTAS3 CCC (Direct) with its own Target address.

[Clearing conditions]

- When writing 0.
- When receiving ENTAS0 CCC (Broadcast), ENTAS1 CCC (Broadcast), or ENTAS2 CCC (Broadcast).
- When receiving ENTAS0 CCC (Direct), ENTAS1 CCC (Direct), or ENTAS2 CCC (Direct) with its own Target address.

35.2.46 Max Write Length Register (ICMWLR)

Address(es): RI3C0.ICMWLR 000E C358h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MWL[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	MWL[15:0]	Max Write Length	Set the maximum data write length in bytes to respond to GETMWL CCC. Set a value of 8 or more. These bits are automatically updated when SETMWL CCC is received.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This register is used in I3C Secondary Controller and I3C Target modes.

MWL[15:0] Bits (Max Write Length)

These bits are used when the I3C Controller sets the maximum data write length in RI3C or gets the maximum data write length from RI3C.

The value of these bits is automatically transmitted in response to GETMRL CCC.

When receiving SETMWL CCC (Broadcast) or SETMWL CCC (Direct) with its own Target address, these bits are automatically updated to the received MWL value.

35.2.47 Max Read Length Register (ICMRLR)

Address(es): RI3C0.ICMRLR 000E C35Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	IBIPL[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MRL[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	MRL[15:0]	Max Read Length	Set the maximum data read length in bytes to respond to GETMRL CCC. Set a value of 16 or more. These bits are automatically updated when SETMRL CCC is received.	R/W
b23 to b16	IBIPL[7:0]	IBI Payload Size	Set the maximum IBI payload size in bytes to respond to GETMRL CCC. These bits are automatically updated when SETMRL CCC is received.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This register is used in I3C Secondary Controller and I3C Target modes.

MRL[15:0] Bits (Max Read Length)

These bits are used when the I3C Controller sets the maximum data read length in RI3C or gets the maximum data read length from RI3C.

The value of these bits is automatically transmitted as the first two bytes of the response to GETMRL CCC.

When receiving SETMRL CCC (Broadcast) or SETMRL CCC (Direct) with its own Target address, these bits are automatically updated to the received MRL value.

IBIPL[7:0] Bits (IBI Payload Size)

These bits are used when the I3C Controller sets the maximum IBI payload size in RI3C or gets the maximum IBI payload size from RI3C.

When the ICDCTR.IBIPL bit is 1, the value of these bits is transmitted as the third byte of the response to GETMRL CCC. When the IBIPL bit is 0, these bits are not transmitted.

When receiving SETMRL CCC (Broadcast) or SETMRL CCC (Direct) with its own Target address, these bits are automatically updated to the received IBI Payload Size value.

35.2.48 Test Mode Register (ICTMR)

Address(es): RI3C0.ICTMR 000E C360h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	TM[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	TM[7:0]	Test Mode	00h: Exit test mode This value removes all I3C devices from test mode. 01h: Vendor test mode This value indicates that I3C devices shall return a random 32-bit value in the provisioned ID during the Dynamic Address Assignment procedure. Settings other than above are prohibited.	R
b31 to b8	—	Reserved	These bits are read as 0.	R

Note: This register is used in I3C Secondary Controller and I3C Target modes.

TM[7:0] Bits (Test Mode)

These bits are used when the I3C Controller puts RI3C into test mode.

When receiving ENTTM CCC (Broadcast), these bits are automatically updated to the received Test Mode Byte value.

35.2.49 Device Status Register (ICDSR)

Address(es): RI3C0.ICDSR 000E C364h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	VRSV[7:0]								CAS[1:0]		PERR	—	PNDINT[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	PNDINT[3:0]	Pending Interrupt Number	Contains the interrupt number of any pending interrupt, or 0000b if no interrupts are pending.	R/W
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R
b5	PERR	Protocol Error Flag	0: The Target has not detected a protocol error since the last status read. 1: The Target has detected a protocol error since the last status read.	R
b7, b6	CAS[1:0]	Current Activity State	b7 b6 0 0: Activity state 0 0 1: Activity state 1 1 0: Activity state 2 1 1: Activity state 3	R/W
b15 to b8	VRSV[7:0]	Vendor Reserved Area	Reserved for vendor-specific meaning	R/W
b31 to b16	—	Reserved	This bit is read as 0. The write value should be 0.	R

Note: This register is used in I3C Secondary Controller and I3C Target modes.

This register is used to store the status to be returned when current status is requested from I3C Controller.
The value of bits b15 to b0 in this register is automatically transmitted when a GETSTATUS CCC (Direct) is received.

PNDINT[3:0] Bits (Pending Interrupt Number)

These bits are used to indicate whether any interrupts are pending. If no interrupts are pending, set these bits to 0000b.
If multiple interrupts are pending, set the interrupt number of the highest priority interrupt.

PERR Flag (Protocol Error Flag)

This bit indicates whether the Target has detected a protocol error since the last status read.
This flag is automatically cleared each time the Controller successfully reads the Target status.
[Setting condition]

- When the Target detected a protocol error.

[Clearing condition]

- When a transmission of status (value set in this register) is successfully completed in response to GETSTATUS CCC (Direct) with its own Target address.

CAS[1:0] Bits (Current Activity State)

These bits are used to indicate the current Activity state of RI3C.

35.2.50 Max Write Speed Register (ICMWSR)

Address(es): RI3C0.ICMWSR 000E C368h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	MSWDR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	MSWDR[2:0]	Maximum Sustained Write Data Rate	b2 b0 0 0 0: Maximum value of f_{SCL} 0 0 1: 8 MHz 0 1 0: 6 MHz 0 1 1: 4 MHz 1 0 0: 2 MHz Settings other than above are prohibited.	R/W
b31 to b3	—	Reserved	This bit is read as 0. The write value should be 0.	R

Note: This register is used in I3C Secondary Controller and I3C Target modes.

This register is used to store the maximum write speed to be returned when the maximum data speed is requested from I3C Controller.

The value of bits b7 to b0 in this register is automatically transmitted as MaxWr when a GETMXDS CCC (Direct) is received.

35.2.51 Max Read Speed Register (ICMRSR)

Address(es): RI3C0.ICMRSR 000E C36Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	TSCO[2:0]			MSRDR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	MSRDR[2:0]	Maximum Sustained Read Data Rate	b2 b0 0 0 0: Maximum value of f_{SCL} 0 0 1: 8 MHz 0 1 0: 6 MHz 0 1 1: 4 MHz 1 0 0: 2 MHz Settings other than above are prohibited.	R/W
b5 to b3	TSCO[2:0]	Clock-to-Data Turnaround Time	b5 b3 0 0 0: 8 ns or less 0 0 1: 9 ns or less 0 1 0: 10 ns or less 0 1 1: 11 ns or less 1 0 0: 12 ns or less 1 1 1: t_{SCO} is more than 12 ns, and is reported by private agreement. Settings other than above are prohibited.	R/W
b31 to b6	—	Reserved	This bit is read as 0. The write value should be 0.	R

Note: This register is used in I3C Secondary Controller and I3C Target modes.

This register is used to store the maximum read speed to be returned when the maximum data speed is requested from I3C Controller.

The value of bits b7 to b0 in this register is automatically transmitted as MaxRd when a GETMXDS CCC (Direct) is received.

35.2.52 Maximum Read Turnaround Time Register (ICMTTR)

Address(es): RI3C0.ICMTTR 000E C370h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MRTTE	—	—	—	—	—	—	—	MRTT[23:16]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MRTT[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b23 to b0	MRTT[23:0]	Maximum Read Turnaround Time	Set the read turnaround time in 1 µs units in the range of 0 to 16 seconds. 000000h: 0 µs (minimum value) 000001h: 1 µs : : F42400h: 16 seconds (maximum value) Settings other than above are prohibited.	R/W
b30 to b24	—	Reserved	This bit is read as 0. The write value should be 0.	R
b31	MRTTE	Maximum Read Turnaround Time Transmit Enable	0: Disables transmission of the maximum read turnaround time. (GETMXDS format 1: without turnaround) 1: Enables transmission of the maximum read turnaround time. (GETMXDS format 2: with turnaround)	R/W

Note: This register is used in I3C Secondary Controller and I3C Target modes.

This register is used to store the three-byte maximum read turnaround time to be returned when the maximum data speed is requested from I3C Controller.

The value of bits b23 to b0 in this register is automatically transmitted as MaxRdTurn when a GETMXDS CCC (Direct) is received.

35.2.53 Timing Support Information Register (ICTSIR)

Address(es): RI3C0.ICTSIR 000E C374h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	INAC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FREQ[7:0]								—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	This bit is read as 0. The write value should be 0.	R
b15 to b8	FREQ[7:0]	Frequency Setting	These bits specify the Target's internal oscillation frequency in increments of 0.5 MHz. b15 b8 00000000: 32.0 kHz 00000001: 0.5 MHz 00000010: 1.0 MHz : : 01111111: 63.5 MHz Settings other than above are prohibited.	R/W
b23 to b16	INAC[7:0]	Frequency Inaccuracy Setting	These bits specify the inaccuracy of the oscillation frequency from 0.0% to 25.5% in increments of 0.1%. b23 b16 00000000: 0.0% 00000001: 0.1% : : 11111110: 25.4% 11111111: 25.5%	R/W
b31 to b24	—	Reserved	This bit is read as 0. The write value should be 0.	R

Note: This register is used in I3C Secondary Controller and I3C Target modes.

This register is used to store the value to be returned when the timing information is requested from I3C Controller. When a GETMXDS CCC (Direct) is received, the value of bits b15 to b8 and b23 to b16 in this register is automatically transmitted as Frequency Byte and Inaccuracy Byte, respectively.

35.2.54 Bit Count Register (ICBCR)

Address(es): RI3C0.ICBCR 000E C380h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	BC[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	BC[4:0]	Bit Counter	These bits indicate the number of bits remaining to be transferred. Refer to Table 35.3 and Table 35.4 for details.	R
b31 to b5	—	Reserved	These bits are read as 0.	R

BC[4:0] Bits (Bit Counter)

These bits function as a counter that indicates the number of bits remaining to be transferred at the detection of a rising edge on the SCL00 line.

Table 35.3 Counter Value and Number of Remaining Bits (Legacy I²C Transfer)

	Controller		Target	
BC[4:0]	Address Phase	Data Phase	Address Phase	Data Phase
00h	2 or 1 bits	2 or 1 bits	3 to 1 bits	2 or 1 bits
01h	3 bits	3 bits	4 bits	3 bits
02h	4 bits	4 bits	5 bits	4 bits
03h	5 bits	5 bits	6 bits	5 bits
04h	6 bits	6 bits	7 bits	6 bits
05h	7 bits	7 bits	8 bits	7 bits
06h	8 bits	8 bits	9 bits	8 bits
07h	9 bits	9 bits	—	9 bits

Table 35.4 Counter Value and Number of Remaining Bits (I3C Transfer)

	SDR*1	
BC[4:0]	Transmission	Reception
00h	1 bit	2 or 1 bits
01h	2 bits	3 bits
02h	3 bits	4 bits
03h	4 bits	5 bits
04h	5 bits	6 bits
05h	6 bits	7 bits
06h	7 bits	8 bits
07h	8 bits	9 bits
08h	9 bits	—

Note 1. The address phase is the same as in Table 35.3.

35.2.55 Queue Buffer Status Register (ICQBSR)

Address(es): RI3C0.ICQBSR 000E C394h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	ISC[4:0]					IQFL[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RQFL[7:0]								CQFL[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	CQFL[7:0]	Command Queue Free Level	Number of free entries currently in the command queue.	R
b15 to b8	RQFL[7:0]	Response Queue Fill Level	Number of entries currently in the response queue.	R
b23 to b16	IQFL[7:0]	IBI Queue Fill Level	Number of entries currently in the IBI queue.	R
b28 to b24	ISC[4:0]	IBI Status Count*1	Number of IBI status entries currently in the IBI queue.	R
b31 to b29	—	Reserved	These bits are read as 0.	R

Note 1. These bits are used in I3C Primarily Controller and I3C Secondary Controller modes.

35.2.56 Data Buffer Status Register (ICDBSR)

Address(es): RI3C0.ICDBSR 000E C398h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RBFL[7:0]								TBFL[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	TBFL[7:0]	Transmit Data Buffer Free Level	Indicates the free level of the transmit data buffer.	R
b15 to b8	RBFL[7:0]	Receive Data Buffer Fill Level	Indicates the fill level of the receive data buffer.	R
b31 to b16	—	Reserved	These bits are read as 0.	R

35.2.57 Receive Status Queue Status Register (ICSQSR)

Address(es): RI3C0.ICSQSR 000E C3C0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	SQFL[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	SQFL[7:0]	Receive Status Queue Fill Level	Number of entries currently in the receive status queue.	R
b31 to b8	—	Reserved	These bits are read as 0.	R

Note: This register is used in I3C Secondary Controller and I3C Target modes.

35.2.58 Internal Status Monitor Register (ICIMR)

Address(es): RI3C0.ICIMR 000E C3CCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	SDAO	SCLO	SDAI	SCLI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	SCLI	SCL Line Monitor Flag	0: SCL00 line is low. 1: SCL00 line is high.	R
b1	SDAI	SDA Line Monitor Flag	0: SDA00 line is low. 1: SDA00 line is high.	R
b2	SCLO	SCL Output Monitor Flag	0: The RI3C drives the SCL00 pin low. 1: The RI3C releases the SCL00 pin.	R
b3	SDAO	SDA Output Monitor Flag	0: The RI3C drives the SDA00 pin low. 1: The RI3C releases the SDA00 pin.	R
b31 to b4	—	Reserved	These bits are read as 0.	R

SCLO Flag (SCL Output Monitor Flag) and SDAO Flag (SDA Output Monitor Flag)

When reading these bits, the state of signals output from the RI3C can be read.

35.2.59 Controller Error Count Register (ICCECR)

Address(es): RI3C0.ICCECR 000E C3D0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	CE2C[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	CE2C[7:0]	CE2 Error Counter	These bits counts I3C Type CE2 errors on the I3C bus. These bits are cleared when read.	R
b31 to b8	—	Reserved	These bits are read as 0.	R

Note: This register is used in I3C Primarily Controller and I3C Secondary Controller modes.

35.3 Data Structures

35.3.1 Command Descriptor

The Command Descriptor is 64 bits in length. The Command Descriptor is added to the command queue via writes to the command queue register (ICCQR).
Write to the ICCQR register in the following order:

1. First write: The lower 32 bits of Command Descriptor
2. Second write: The upper 32 bits of Command Descriptor.

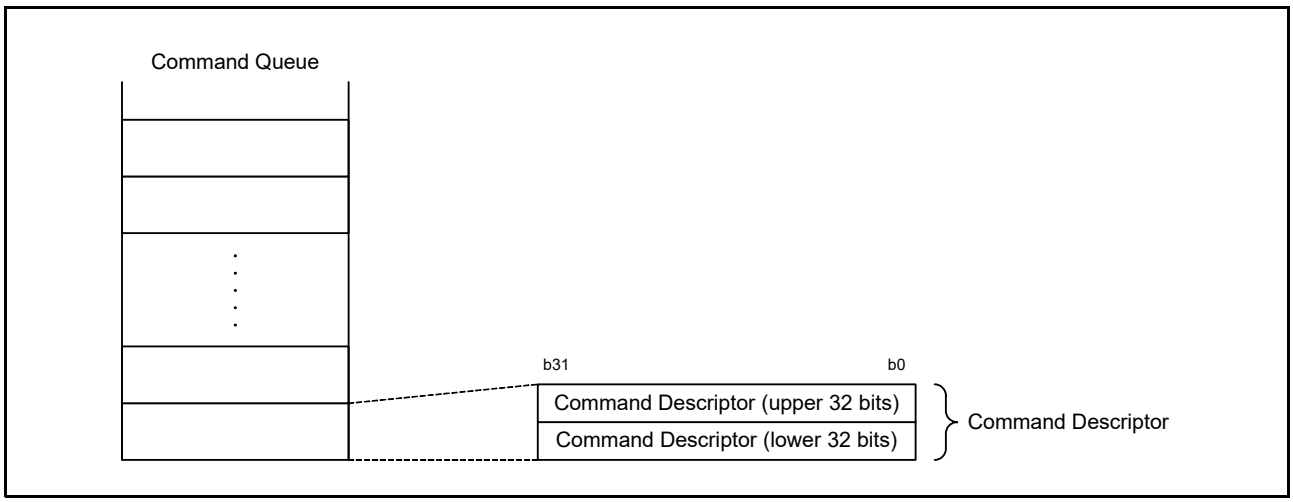


Figure 35.2 Command Descriptor Data Structure

RI3C provides a Command Descriptor structure for each command type as follows:

- Address Assignment Command
- Immediate Data Transfer Command
- Regular Data Transfer Command
- Combo Transfer Command
- Internal Control Command

35.3.1.1 Address Assignment Command

This command is used for Address Assignment (ENTDAA, SETDASA)*1.
Details of the address assignment command structure are as follows.

Note 1. When issuing SETAASA CCC, use the immediate data transfer command.

	b63	b62	b61	b60	b59	b58	b57	b56	b55	b54	b53	b52	b51	b50	b49	b48
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b47	b46	b45	b44	b43	b42	b41	b40	b39	b38	b37	b36	b35	b34	b33	b32
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TOC	ROC	DEV_COUNT				—	—	—	—	EXT_DEVICE	DEV_INDEX				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	CMD								TID				CMD_ATTR		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CMD_ATTR	Command Attributes	Set this field to 010b (address assignment command).	W
b6 to b3	TID	Transaction ID	Used as a tag for this command. Specify in the range of 0h to 7h.	W
b14 to b7	CMD	Transfer Command CCC Value	Specify the command code of the command to be used. ENTDAA (Broadcast): 07h SETDASA (Direct): 87h	W
b15	—	Reserved	Set this field to 0.	W
b20 to b16	DEV_INDEX	Device Index	Specify the DAT table index (m of the ICTDATRm register) for the Target device being assigned an address via Dynamic Address Assignment.	W
b21	EXT_DEVICE	Extended Device Index	0: Use the ICTDATRm register indicated by the DEV_INDEX field. 1: Use the ICEDATR register.	W
b25 to b22	—	Reserved	The write value should be 0.	W
b29 to b26	DEV_COUNT	Device Count	Specify the number of devices that a Dynamic Address shall be assigned to.	W
b30	ROC	Response on Completion	0: NOT_REQUIRED: Response status is not required. 1: REQUIRED: Response status is required.	W
b31	TOC	Terminate on Completion	0: RESTART: Issue repeated START condition (Sr) at end of transfer 1: STOP: Issue STOP condition (P) at end of transfer	W
b63 to b32	—	Reserved	The write value should be 0.	W

TID Field (Transaction ID)

The value specified in this field is reflected in the TID field of the Response Descriptor.

DEV_INDEX Field (Device Index)

Indicates the DAT table index for the Target device being assigned an address via Dynamic Address Assignment.
Write a Dynamic Address into each DAT entry (the ICTDATRm registers) that will be used with this command type, as well as a Static Address (when used with the SETDASA CCC).

DEV_COUNT Field (Device Count)

Indicates the number of devices that a Dynamic Address is assigned to. The DAT entries are processed starting at index DEV_INDEX for up to DEV_COUNT times.

ROC Field (Response on Completion)

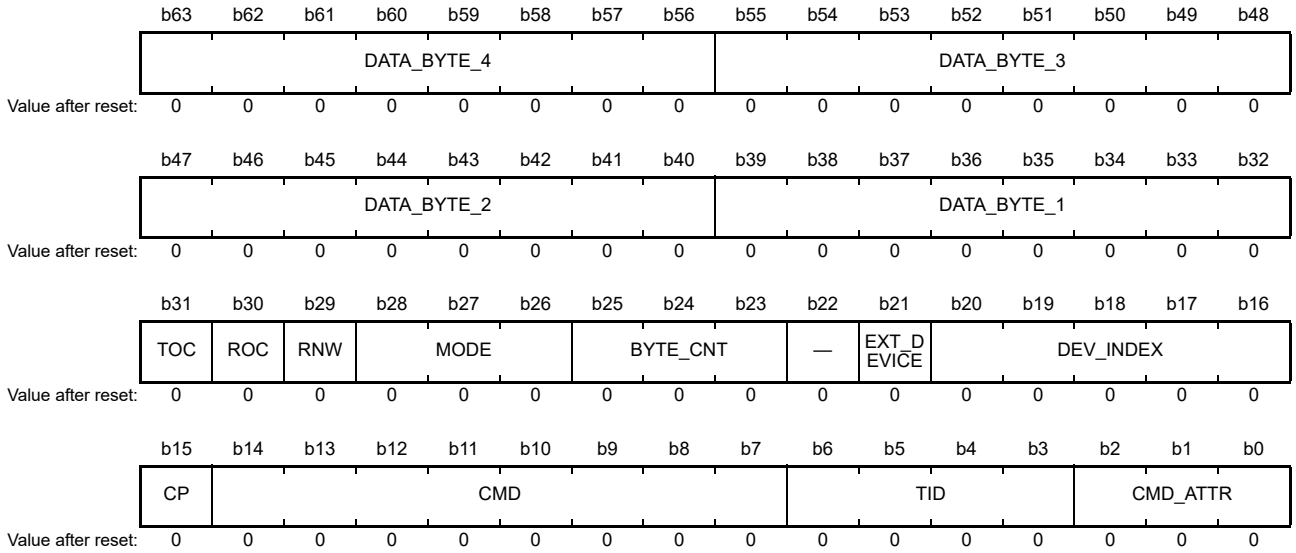
Controls whether Response status is sent after successful completion of the transfer command. The successful completion shall be read from the ICRQR register. Upon unsuccessful transfer the Response status shall always be sent.

TOC Field (Terminate on Completion)

Controls what bus condition to issue after the transfer command completes.
For ENTDAA, a STOP condition is issued regardless of the value set in this field.
When sending SETDASA CCC with the TOC field set to 0 (RESTART), the next command must also be set to an address assignment command specifying SETDASA CCC. When the next command is not SETDASA CCC, the TOC field must be set to 1 (STOP).

35.3.1.2 Immediate Data Transfer Command

This Command Descriptor directly contains data bytes (up to 4 bytes) to be transferred, and as a result is only useful for shorter write-type transfers or CCCs that write data. This command shall not be used for read-type transfers.
Details of the immediate data transfer command structure are as follows.



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CMD_ATTR	Command Attribute	Set this field to 001b (immediate data transfer command).	W
b6 to b3	TID	Transaction ID	Used as a tag for this command. Specify in the range of 0h to 7h.	W
b14 to b7	CMD	Transfer Command CCC Value	Specify the command code of the command to be used.	W
b15	CP	Command Present	0: TRANSFER: This structure describes an SDR transfer, so the CMD field is not valid. 1: CCC: This structure describes a CCC transfer, so the CMD field is valid.	W
b20 to b16	DEV_INDEX	Device Index	Specify the DAT table index (m of the ICTDATRm register) for the Target device being addressed with the transfer.	W
b21	EXT_DEVICE	Extended Device Index	0: Use the ICTDATRm table indicated by DEV_INDEX. 1: Use the ICEDATR table.	W
b22	—	Reserved	The write value should be 0.	W
b25 to b23	BYTE_CNT	Byte Count	Number of valid data bytes to use in this descriptor. Specify in the range of 0h (no payload) to 4h (4 bytes).	W
b28 to b26	MODE	Mode and Speed	• I3C transfer b28 b26 0 0 0: I3C SDR0, ICSBR register setting (up to 12.5 MHz) 0 0 1: I3C SDR1, ICEBR register setting (up to 8 MHz) 0 1 1: I3C SDR2, ICSBR register setting × 2 (up to 6 MHz) 1 0 0: I3C SDR3, ICEBR register setting × 2 (up to 4 MHz) 1 0 1: I3C SDR4, ICEBR register setting × 4 (up to 2 MHz) Settings other than above are prohibited. • I^2C transfer b28 b26 0 0 0: I^2C FM, ICSBR register setting (up to 400 kHz) 0 0 1: I^2C FM+, ICEBR register setting (up to 1 MHz) Settings other than above are prohibited.	W
b29	RNW	Transfer Direction (RnW)	Set this field to 0 (write transfer)	W
b30	ROC	Response on Completion	0: NOT_REQUIRED: Response status is not required. 1: REQUIRED: Response status is required.	W

Bit	Symbol	Bit Name	Description	R/W
b31	TOC	Terminate on Completion	0: RESTART: Issue repeated START condition (Sr) at end of transfer 1: STOP: Issue STOP condition (P) at end of transfer	W
b39 to b32	DATA_BYTE_1	Data Byte 1	First byte of data to be transmitted	W
b47 to b40	DATA_BYTE_2	Data Byte 2	Second byte of data to be transmitted	W
b55 to b48	DATA_BYTE_3	Data Byte 3	Third byte of data to be transmitted	W
b63 to b56	DATA_BYTE_4	Data Byte 4	Fourth byte of data to be transmitted	W

DEV_INDEX Field (Device Index)

Indicates the DAT table (the ICTDATRm register) index for the Target device being addressed with the transfer. The ICTDATRm register indicated by this field must contain a valid Dynamic Address. This field is ignored for Broadcast CCCs.

BYTE_CNT Field (Byte Count)

Number of valid data bytes to use in this descriptor. This field should be set to non-zero value, except for CCCs that do not have any payload defined.

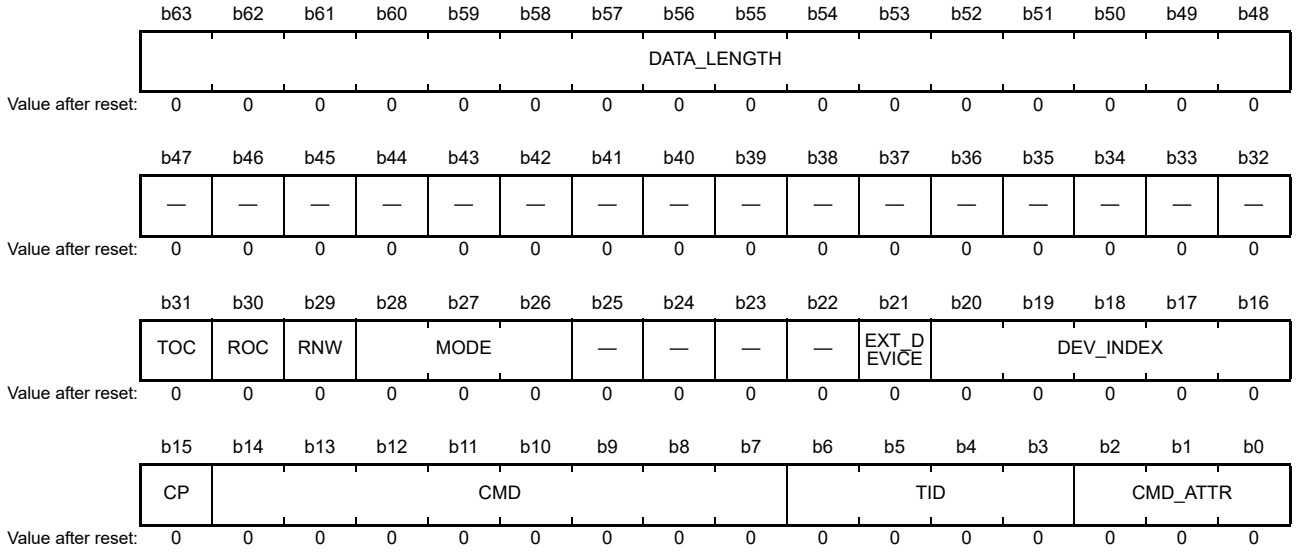
MODE Field (Mode and Speed)

Sets the mode and speed for the I3C or I^2C transfer.
Interpretation of this field depends on the TYPE bit in the ICTDATRm register indexed by the DEV_INDEX field.

35.3.1.3 Regular Data Transfer Command

This Command Descriptor does not contain data to be transferred. For I3C Controller mode, data is transferred via the transmit/receive data register (ICDR), and for I3C Target mode, data is transferred via the IBI queue register (ICIQR). Details of the regular data transfer command structure of each mode are as follows.

(1) I3C Controller Mode



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CMD_ATTR	Command Attribute	Set this field to 000b (regular data transfer command).	W
b6 to b3	TID	Transaction ID	Used as a tag for this command. Specify in the range of 0h to 7h.	W
b14 to b7	CMD	Transfer Command CCC Value	Specify the command code of the command to be used.	W
b15	CP	Command Present	0: TRANSFER: This structure describes an SDR transfer, so the CMD field is not valid. 1: CCC: This structure describes a CCC transfer, so the CMD field is valid.	W
b20 to b16	DEV_INDEX	Device Index	Specify the DAT table index (m of the ICTDATRm register) for the Target device being addressed with the transfer.	W
b21	EXT_DEVICE	Extended Device Index	0: Use the ICTDATRm table indicated by DEV_INDEX. 1: Use the ICEDATR table.	W
b25 to b22	—	Reserved	The write value should be 0.	W
b28 to b26	MODE	Mode and Speed	• I3C transfer b28 b26 0 0 0: I3C SDR0, ICSBR register setting (up to 12.5 MHz) 0 0 1: I3C SDR1, ICEBR register setting (up to 8 MHz) 0 1 1: I3C SDR2, ICSBR register setting × 2 (up to 6 MHz) 1 0 0: I3C SDR3, ICEBR register setting × 2 (up to 4 MHz) 1 0 1: I3C SDR4, ICEBR register setting × 4 (up to 2 MHz) Settings other than above are prohibited. • I^2C transfer b28 b26 0 0 0: I^2C FM, ICSBR register setting (up to 400 kHz) 0 0 1: I^2C FM+, ICEBR register setting (up to 1 MHz) Settings other than above are prohibited.	W
b29	RNW	Transfer Direction (RnW)	0: WRITE: Write transfer 1: READ: Read transfer	W

Bit	Symbol	Bit Name	Description	R/W
b30	ROC	Response on Completion	0: NOT_REQUIRED: Response status is not required. 1: REQUIRED: Response status is required.	W
b31	TOC	Terminate on Completion	0: RESTART: Issue repeated START condition (Sr) at end of transfer 1: STOP: Issue STOP condition (P) at end of transfer	W
b47 to b32	—	Reserved	The write value should be 0.	W
b63 to b48	DATA_LENGTH	Data Length	Indicates the number of bytes to be transferred.	W

RNW Field (Transfer Direction (RnW))

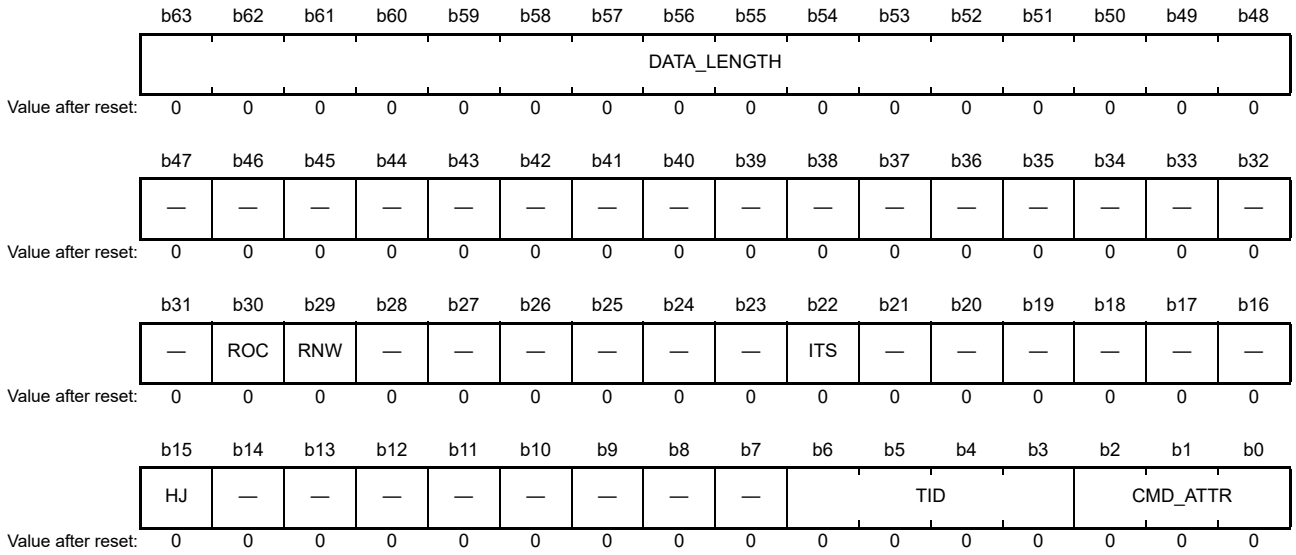
Identifies the direction of this transfer.

DATA_LENGTH Field (Data Length)

Number of bytes to be transferred.

This field should be set to a non-zero value, except for CCCs that do not have payload defined. Length setting of GETMXDS command should be fixed to “0005h”.

(2) I3C Target Mode



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CMD_ATTR	Command Attribute	Set this field to 000b (regular data transfer command).	W
b6 to b3	TID	Transaction ID	Used as a tag for this command. Specify in the range of 0h to 7h.	W
b14 to b7	—	Reserved	The write value should be 0.	W
b15	HJ	Hot-Join Event	0: Target Interrupt Request or Controller Role Request 1: Hot-Join event	W
b21 to b16	—	Reserved	The write value should be 0.	W
b22	ITS	Include Timestamp	0: Do not include timestamp. 1: Include timestamp.	W
b28 to b23	—	Reserved	The write value should be 0.	W
b29	RNW	Transfer Direction (RnW)	0: WRITE: Write transfer (Controller Role Request) 1: READ: Read transfer (Target Interrupt Request)	W
b30	ROC	Response on Completion	0: NOT_REQUIRED: Response status is not required. 1: REQUIRED: Response status is required.	W
b47 to b31	—	Reserved	The write value should be 0.	W
b63 to b48	DATA_LENGTH	Data Length	Indicates the number of bytes to be transferred.	W

HJ Field (Hot-Join Event)

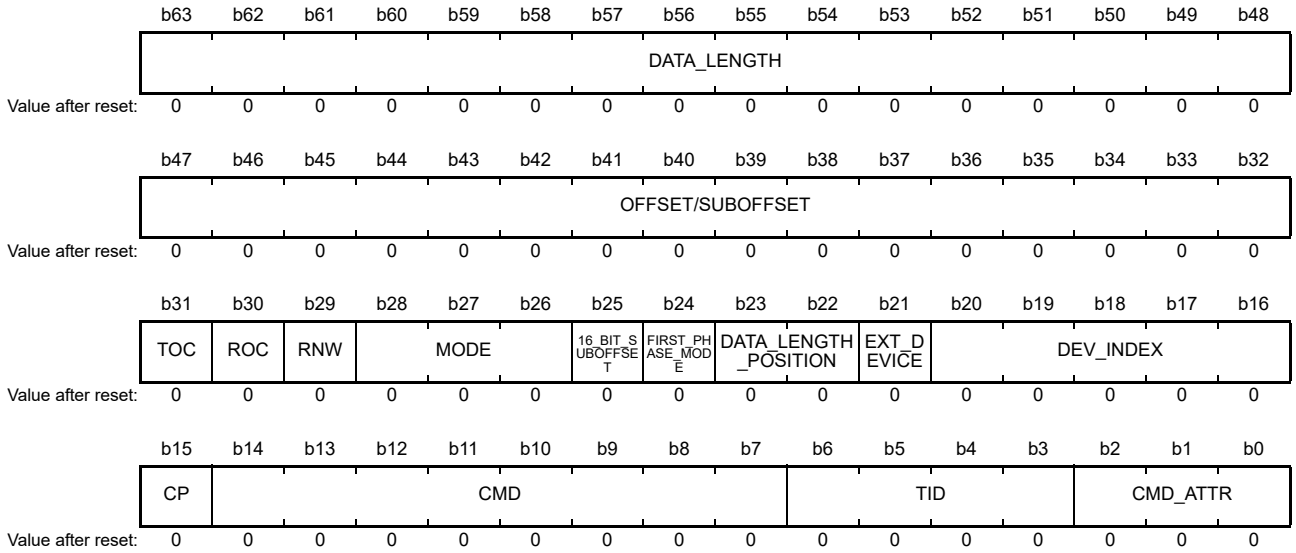
Indicates whether Hot-Join Event is valid in this IBI Data transfer.

RNW Field (Transfer Direction (RnW))

Identifies the direction of this transfer. This field is invalid for Hot-Join event.

35.3.1.4 Combo Transfer Command

This Command Descriptor describes a combined Write + Read or Write + Write operation.
Data is transferred via the transmit/receive data register (ICDR).
Details of the combo transfer command structure are as follows.



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CMD_ATTR	Command Attribute	Set this field to 011b (combo transfer command).	W
b6 to b3	TID	Transaction ID	Used as a tag for this command. Specify in the range of 0h to 7h.	W
b14 to b7	CMD	Transfer Command CCC Value	This field is reserved for this command. Set it to 00h.	W
b15	CP	Command Present	Set this field to 0 for this command.	W
b20 to b16	DEV_INDEX	Device Index	Specify the DAT table index (m of the ICTDATRm register) for the Target device being addressed with the transfer.	W
b21	EXT_DEVICE	Extended Device Index	0: Use the ICTDATRm table indicated by DEV_INDEX. 1: Use the ICEDATR table.	W
b23, b22	DATA_LENGTH_POSITION	Data Length Field Position	Set this field to 00b.	W
b24	FIRST_PHASE_MODE	First Phase Mode	0: SDR: First phase is executed in SDR mode. 1: MODE: First phase is executed in the mode indicated by the MODE field.	W
b25	16_BIT_SUBOFFSET	Sub Offset Size	0: 8_BIT_SUBOFFSET: Sub-offset is 8 bits long. Value is encoded in lower byte of OFFSET/SUBOFFSET field. 1: 16_BIT_SUBOFFSET: Sub-offset is 16 bits long.	W
b28 to b26	MODE	Mode and Speed	• I3C transfer b28 b26 0 0 0: I3C SDR0, ICSBR register setting (up to 12.5 MHz) 0 0 1: I3C SDR1, ICEBR register setting (up to 8 MHz) 0 1 1: I3C SDR2, ICSBR register setting × 2 (up to 6 MHz) 1 0 0: I3C SDR3, ICEBR register setting × 2 (up to 4 MHz) 1 0 1: I3C SDR4, ICEBR register setting × 4 (up to 2 MHz) Settings other than above are prohibited. • I²C transfer b28 b26 0 0 0: I²C FM, ICSBR register setting (up to 400 kHz) 0 0 1: I²C FM+, ICEBR register setting (up to 1 MHz) Settings other than above are prohibited.	W

Bit	Symbol	Bit Name	Description	R/W
b29	RNW	Transfer Direction (RnW)	0: WRITE: Write transfer 1: READ: Read transfer	W
b30	ROC	Response on Completion	0: NOT_REQUIRED: Response status is not required. 1: REQUIRED: Response status is required.	W
b31	TOC	Terminate on Completion	0: RESTART: Issue repeated START condition (Sr) at end of transfer 1: STOP: Issue STOP condition (P) at end of transfer	W
b47 to b32	OFFSET/ SUBOFFSET	Offset/Sub-Offset	Set the value of the Offset field to be transferred in the first phase.	W
b63 to b48	DATA_LENGTH	Data Length	Specify the number of bytes to be transferred in the second phase. This field must be set to non-zero value.	W

DATA_LENGTH_POSITION Field (Data Length Field Position)

Indicates whether and where to put Data Length (DATA_LENGTH) in the first phase of the transfer.

FIRST_PHASE_MODE Field (First Phase Mode)

Indicates whether the first phase of the combo transfer is executed in SDR mode, vs. the mode indicated by the MODE field.

16_BIT_SUBOFFSET Field (Sub Offset Size)

Indicates whether the length of the Offset and Length fields in the first phase of the transfer is 8 bits or 16 bits. For 8-bit, data should be written in lower bytes of the OFFSET/SUBOFFSET and DATA_LENGTH fields.

RNW Field (Transfer Direction (RnW))

Identifies the direction of the second phase of the transfer.

35.3.1.5 Internal Control Command

This Command Descriptor is used for controlling RI3C itself (not for I3C transfer commands).
Details of the internal control command structure are as follows.

	b63	b62	b61	b60	b59	b58	b57	b56	b55	b54	b53	b52	b51	b50	b49	b48
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b47	b46	b45	b44	b43	b42	b41	b40	b39	b38	b37	b36	b35	b34	b33	b32
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	ON_OFF	MIPI_CMD				—	TID				CMD_ATTR		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CMD_ATTR	Command Attribute	Set this field to 111b (internal control command).	W
b6 to b3	TID	Transaction ID*1	Used as a tag for this command. Specify in the range of 0h to 7h.	W
b7	—	Reserved	The write value should be 0.	W
b11 to b8	MIPI_CMD	MIPI Alliance Command	b11 b8 0 0 0 0: NoOp (performs no transfer) 0 0 1 0: Broadcast Address automatic transmission Settings other than above are prohibited.	W
b12	ON_OFF	On/Off	0: disables automatic transmission of Broadcast Address 1: enables automatic transmission of Broadcast Address*2	W
b63 to b13	—	Reserved	The write value should be 0.	W

Note 1. The Response Descriptor is not stored when the internal control command is executed.
Note 2. Automatic transmission of Broadcast Address enabled by this command is disabled when the ICRCR.ISRST is set to 1.

35.3.2 Response Descriptor

The Response Descriptor is a 32-bit read-only structure describing the success or failure of a command, and the amount of data transferred.
The Response Descriptor is read from the response queue via the response queue register (ICRQR).

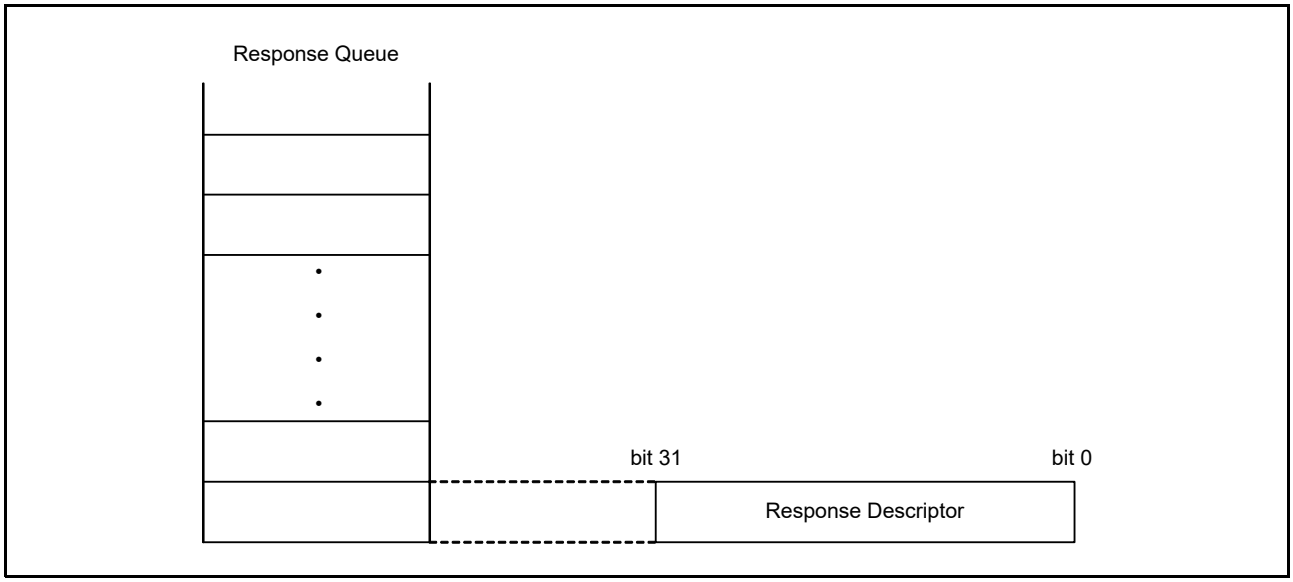
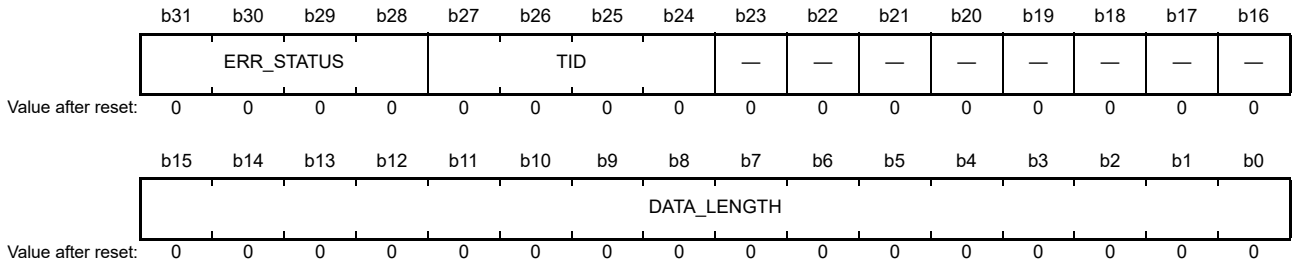


Figure 35.3 Response Descriptor Data Structure

The Response Descriptor structure differs between I3C Controller mode and I3C Target mode.
Details of the Response Descriptor structure of each mode are as follows.

(1) I3C Controller Mode



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	DATA_LENGTH	Data Length/Device Count	For write transfer: Remaining data length (in bytes) For read transfer: Received data length (in bytes) For address assignment: Remaining device count	R
b23 to b16	—	Reserved	These bits are read as 0.	R
b27 to b24	TID	Transaction ID	This value shall match the value of the TID field, for a previously enqueued Command Descriptor that was sent on the bus.	R
b31 to b28	ERR_STATUS	Error Status	b31 b28 0 0 0 0: SUCCESS: Transfer successful, no error 0 1 0 0: ADDR_HEADER: Address header error 0 1 0 1: NACK: Address NACKed or Dynamic Address Assignment NACKed 0 1 1 0: OVL: Receive overflow or transmit underflow error 1 0 0 0: ABORTED: Transfer aborted 1 0 0 1: I2C_WR_DATA_NACK: NACK is received during an I^2C write data transfer 1 0 1 0: NOT_SUPPORTED: Command is not supported	R

TID Field (Transaction ID)

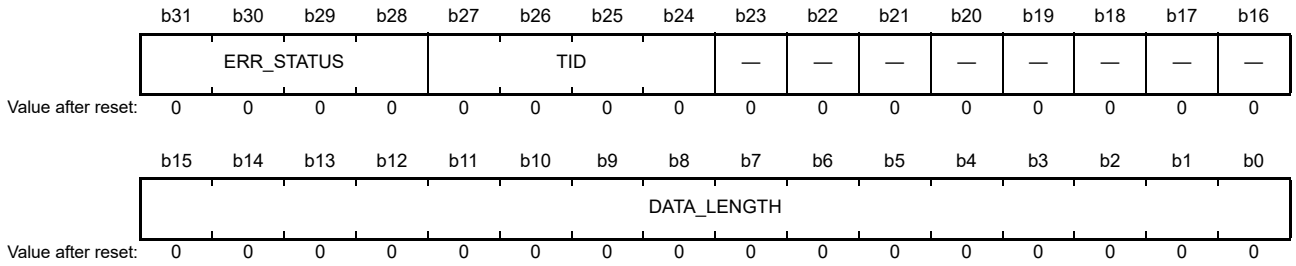
This field reflects the value specified in the TID field of the Command Descriptor. Software can determine which command the response is for by checking this field.

ERR_STATUS Field (Error Status)

This field indicates the transfer status of the command indicated by the TID field.
The details of each error are as follows.

- ADDR_HEADER: No Target devices acknowledge the Broadcast Address.
- NACK: Any Target device that responds to the ENTDAA procedure does not accept the Dynamic Address or an addressed Target device does not acknowledge its Dynamic Address.
- OVL: Transmit data buffer underflowed or receive data buffer overflowed.
- ABORTED: A transaction was aborted.
- I2C_WR_DATA_NACK: The I^2C Target device does not acknowledge the I^2C write data transfer.
- NOT_SUPPORTED: A Command Descriptor structure with an illegal or invalid combination of field values was stored.

(2) I3C Target Mode



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	DATA_LENGTH	Data Length	For Target Interrupt Request: Remaining data length (in bytes)	R
b23 to b16	—	Reserved	These bits are read as 0.	R
b27 to b24	TID	Transaction ID	This value shall match the value of the TID field, for a previously enqueued Command Descriptor that was sent on the bus.	R
b31 to b28	ERR_STATUS	Error Status	b31 b28 0 0 0 0: SUCCESS: Transfer successful, no error 0 1 0 0: ADDR_HEADER: Address header error 0 1 0 1: NACK: Address NACKed or Dynamic Address Assignment NACKed 0 1 1 0: OVL: Receive overflow or transmit underflow error 1 0 0 0: ABORTED: Transfer aborted 1 0 1 0: NOT_SUPPORTED: Command is not supported	R

ERR_STATUS Field (Error Status)

This field indicates the transfer status of the command indicated by the TID field.

The details of each error are as follows.

- ADDR_HEADER: No Target devices acknowledge the Broadcast Address.
- NACK: Any Target device that responds to the ENTDAA procedure does not accept the Dynamic Address or an addressed Target device does not acknowledge its Dynamic Address.
- OVL: Transmit data buffer underflowed or receive data buffer overflowed.
- ABORTED: A transaction was aborted.
- NOT_SUPPORTED: A Command Descriptor structure with an illegal or invalid combination of field values was stored, an IBI disabled by the ICTEVR register is requested, or an IBI is disabled by DISEC CCC after a Command Descriptor structure for the IBI was enqueued in the command queue.

35.3.3 IBI Status Descriptor

The IBI Status Descriptor is a 32-bit read-only structure describing an IBI event received from a Target device on the I3C bus. This Descriptor is used in I3C Controller mode.
The IBI Status Descriptor is read from the IBI queue via the IBI queue register (ICIQR).

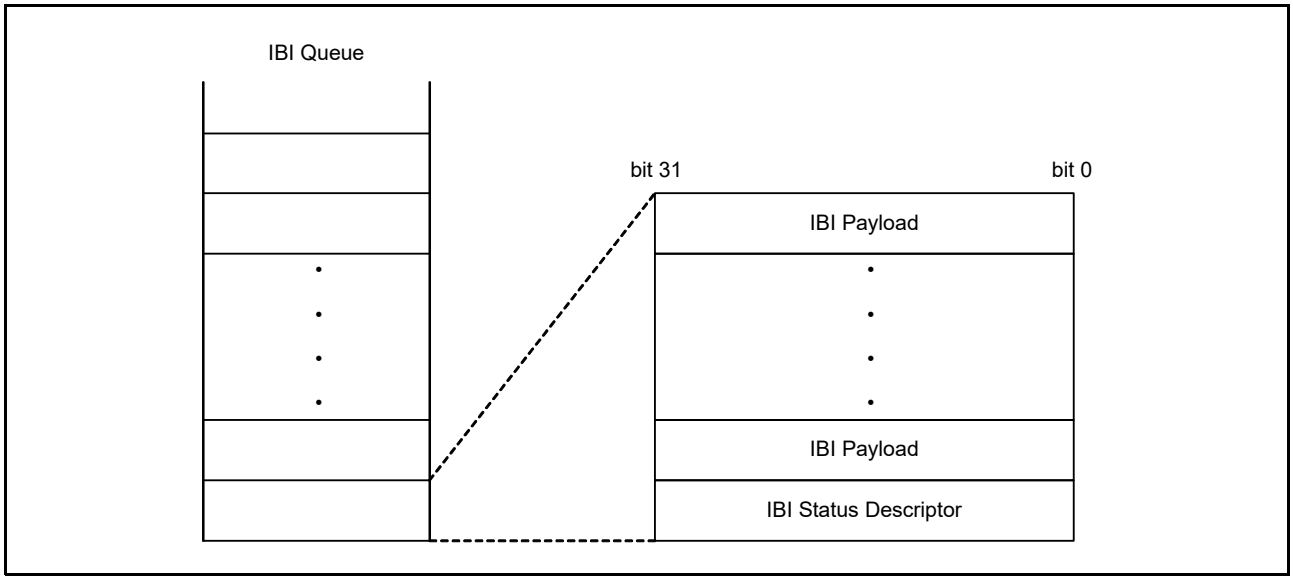


Figure 35.4 IBI Status Descriptor Data Structure

Details of the IBI Status Descriptor Structure are as follows.

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IBI_STS	—	—	ERR_STATUS			TS	LAST_STATUS	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	IBI_ID								DATA_LENGTH							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	DATA_LENGTH	IBI Data Length	Number of data bytes in IBI data.	R
b15 to b8	IBI_ID	IBI Received ID	• Target Interrupt Request or Controller Role Request: b15 to b9: Target's Device Address b8: Value of RnW bit • Hot-Join Request: b15 to b9: Hot-Join ID (= 02h) b8: Value of RnW bit (= 0)	R
b23 to b16	—	Reserved	These bits are read as 0.	R
b24	LAST_STATUS	Last IBI Status	Last IBI status for the IBI transaction.*1	R
b25	TS	IBI Timestamp Present	Indicates whether a timestamp is available for the IBI. 0: OFF: IBI is not timestamped. 1: ON: IBI is timestamped.	R

Bit	Symbol	Bit Name	Description	R/W
b28 to b26	ERR_STATUS	IBI Error Status	b28 b26 0 0 0: SUCCESS 0 1 1: FRAME: Frame error 1 0 0: ADDR_HEADER: Address header error 1 0 1: NACK: Address NACKed 1 1 1: ABORT: Aborted to Controller	R
b30, b29	—	Reserved	These bits are read as 0.	R
b31	IBI_STS	IBI Received Status	0: The IBI was handled with ACK. 1: The IBI was handled with NACK, and then Auto-Disabled.	R

Note 1. Even if the LAST_STATUS field is set to 0, examine the DATA_LENGTH field to evaluate the data payload length.

LAST_STATUS Field (Last IBI Status)

For some IBIs, multiple IBI Status Descriptors might be needed to describe a single IBI event. A 0 in this field indicates that there is a subsequent IBI Status Descriptor.
Even if this field is set to 0, examine the DATA_LENGTH field to evaluate the data payload length.

IBI_STS Field (IBI Received Status)

Indicates how the received IBI was handled by Active Controller.

35.3.4 Receive Status Descriptor

The Receive Status Descriptor is a 32-bit read-only structure describing the success or failure of read/write transfer issued by the Controller, and the amount of data transferred. This Descriptor is used in I3C Target mode.
The Receive Status Descriptor is read from the receive status queue via the receive status queue register (ICSQR).

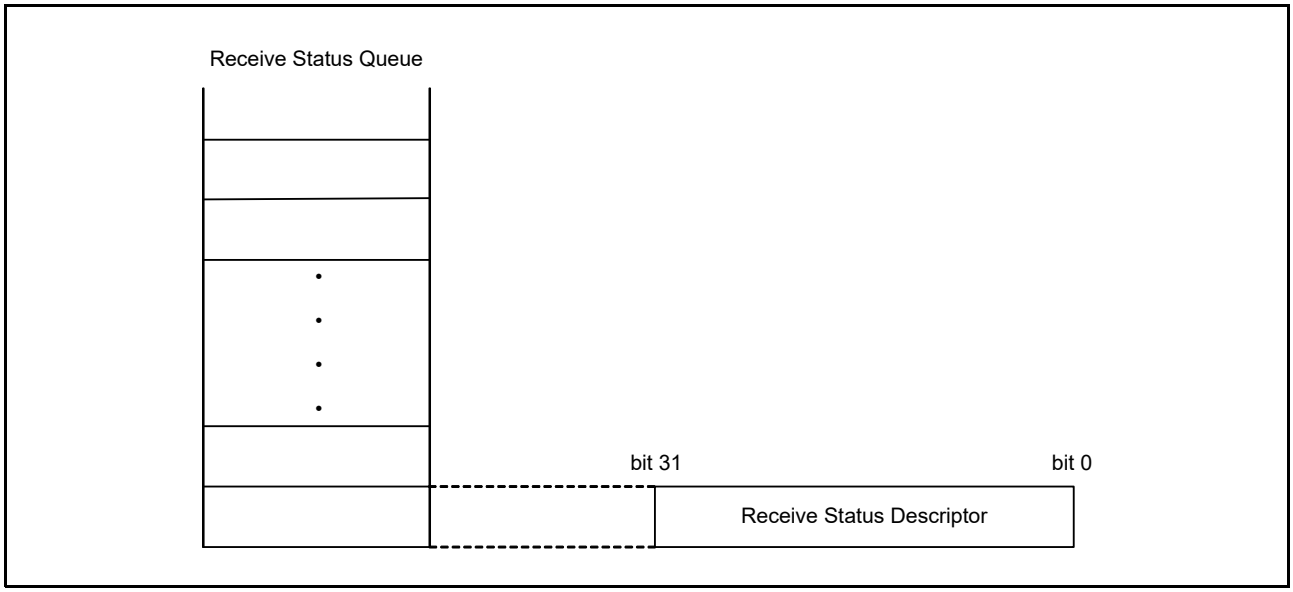
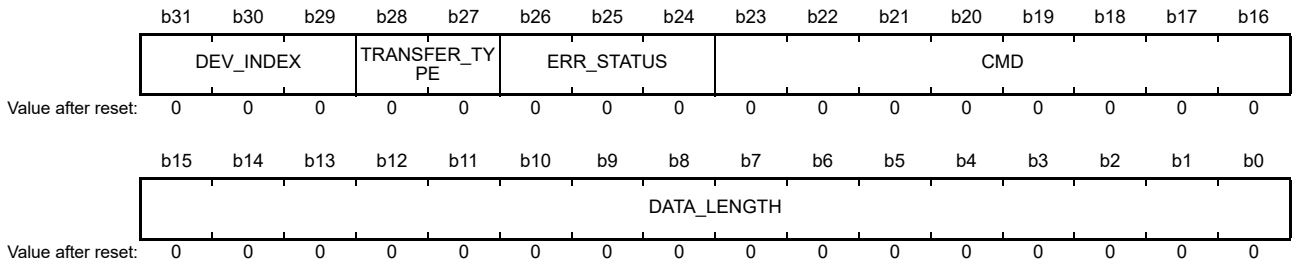


Figure 35.5 Receive Status Descriptor Data Structure

Details of the Receive Status Descriptor structure are as follows.



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	DATA_LENGTH	Data Length	For Write Transfer: Received data length (in bytes) For Read Transfer: Transmitted data length (in bytes)	R
b23 to b16	CMD	Command Code	• I3C SDR private transfer or Legacy I2C transfer: b23: Transfer type (0: Write/1: Read) b22 to b20: Reserved b19: Transfer mode (0: I3C SDR/1: Legacy I2C) b18 to b16: Reserved • SDR CCC transfer: Indicates the CCC command code specified in the Command Descriptor of the transfer.	R
b26 to b24	ERR_STATUS	Error Status	b26 b24 0 0 0: SUCCESS: Transfer successful, no error 1 0 0: ADDR_HEADER: Address header error 1 0 1: NACK: Target NACKed 1 1 0: OVL: FIFO overflow/underflow 1 1 1: ABORTED: Aborted to Controller	R

Bit	Symbol	Bit Name	Description	R/W
b28, b27	TRANSFER_TYPE	Transfer Type	b28 b27 0 0: I3C SDR private transfer or Legacy I^2C transfer 0 1: I3C CCC	R
b31 to b29	DEV_INDEX	Device Index	Indicates the DAT table index (m of the ICTDATRm register) for the responded device with the transfer.	R

35.4 Operation

35.4.1 Data Handler

The relationship between the transfer method and the queue is shown in Table 35.5.
RI3C supports only FIFO buffer transfer. RI3C autonomously starts transfer when data and command are written.

Table 35.5 Transfer Method and Queue

Protocol	Transfer Method	Queue/Buffer	Size	Controller	Target	Secondary Controller
I3C Mode	FIFO buffer transfer	Command Queue	64 bits × 2 stages	✓	✓	✓
		Response Queue	32 bits × 2 stages	✓	✓	✓
		Transmit Data Buffer	32 bits × 2 stages	✓	✓	✓
		Receive Data Buffer	32 bits × 2 stages	✓	✓	✓
		Receive Status Queue	32 bits × 2 stages	—	✓	✓
		IBI Status Queue	32 bits × 2 stages	✓	—	✓
		IBI Data Buffer	32 bits × 6 stages	✓	✓	✓

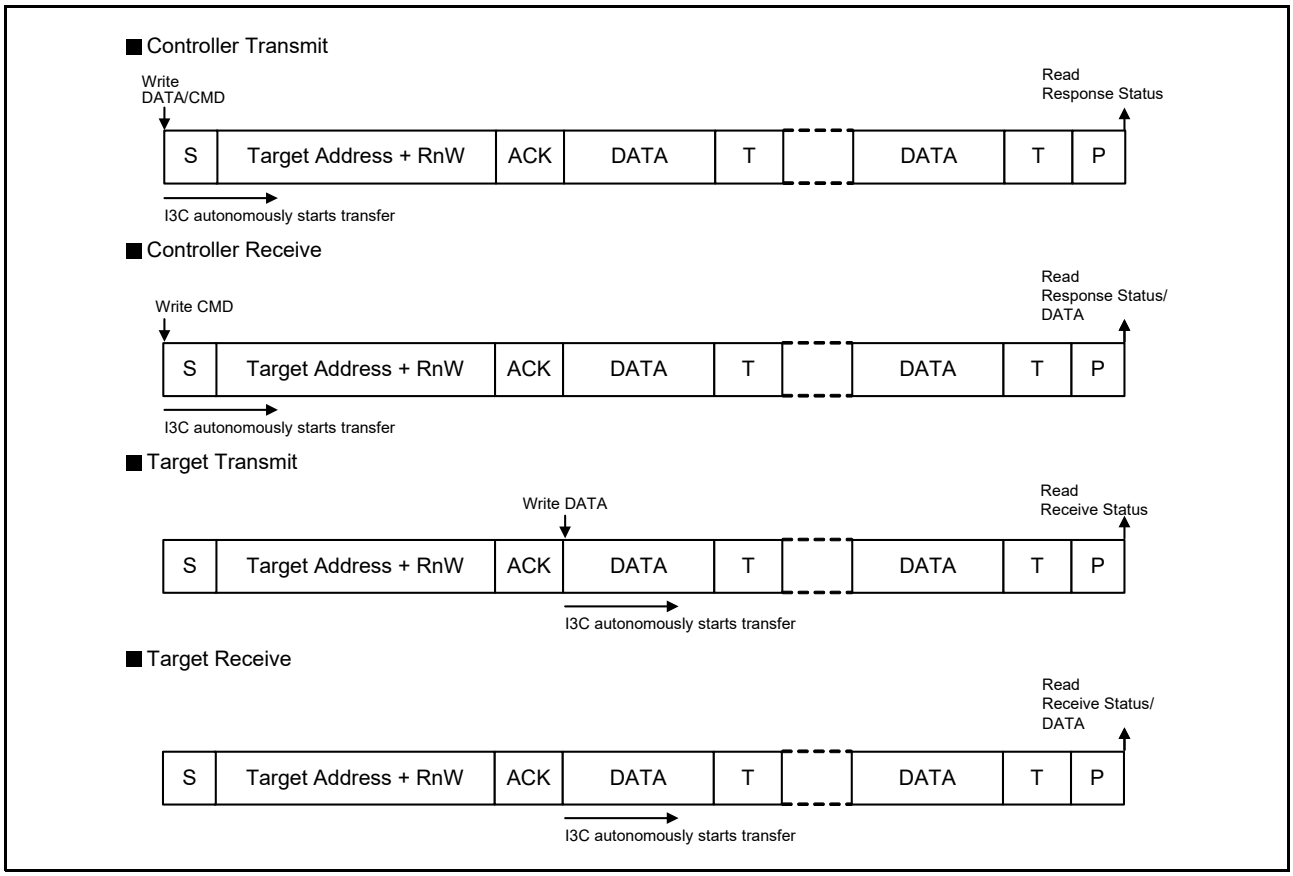


Figure 35.6 Data Handler with FIFO Buffer Transfer

35.4.2 I3C Protocol

35.4.2.1 Communication Protocol

(1) I3C Communication Data Format

Figure 35.7 through Figure 35.10 illustrate a typical communication for each of the supported I3C Protocols. While these diagrams do not exhaustively illustrate all possible RI3C communications, they do serve as useful introductions to the signaling and transmission formatting used in each supported I3C Protocol.

Figure 35.7 illustrates example communication using I3C Single Data Rate (SDR) mode with Broadcast Address (7Eh). It shows the Controller reading a byte of data from the Target at Address 2Bh in SDR Mode.
From the Bus Free condition, the Controller issues a START condition by driving the SDA line Low while keeping the SCL line High. It then issues the Broadcast Address (7Eh) followed by RnW (0 for Write). Then the Controller turns on a pull-up resistor and goes to Open Drain. All Targets ACK by pulling the SDA line Low (in the Figure, pink fill means the Target is in control of the SDA line at this time). The Controller then issues a repeated START condition, then the Address of the Target (2Bh) it wants to read followed by RnW (1 for Read). The Controller then turns on a pull-up resistor and goes to Open Drain, allowing the Target to acknowledge by pulling the SDA line Low. At this point, the Controller continues to toggle the SCL line and release the SDA line, allowing the Target to drive SDA to send one byte of data (4Ah) followed by T. T = 1 informs the Controller that there is additional data, whereas T = 0 signals the end. Here there is additional data, so the Target drives SDA High until SCL goes High, at which time it releases SDA. The Controller has the option of holding SDA High with a weak pull-up, which signals to the Target that the Controller allows another byte to be transmitted, or to pull SDA Low (while SCL is High – hence a repeated START condition), which would signal to the Target that the Controller has terminated the Read and is taking over.

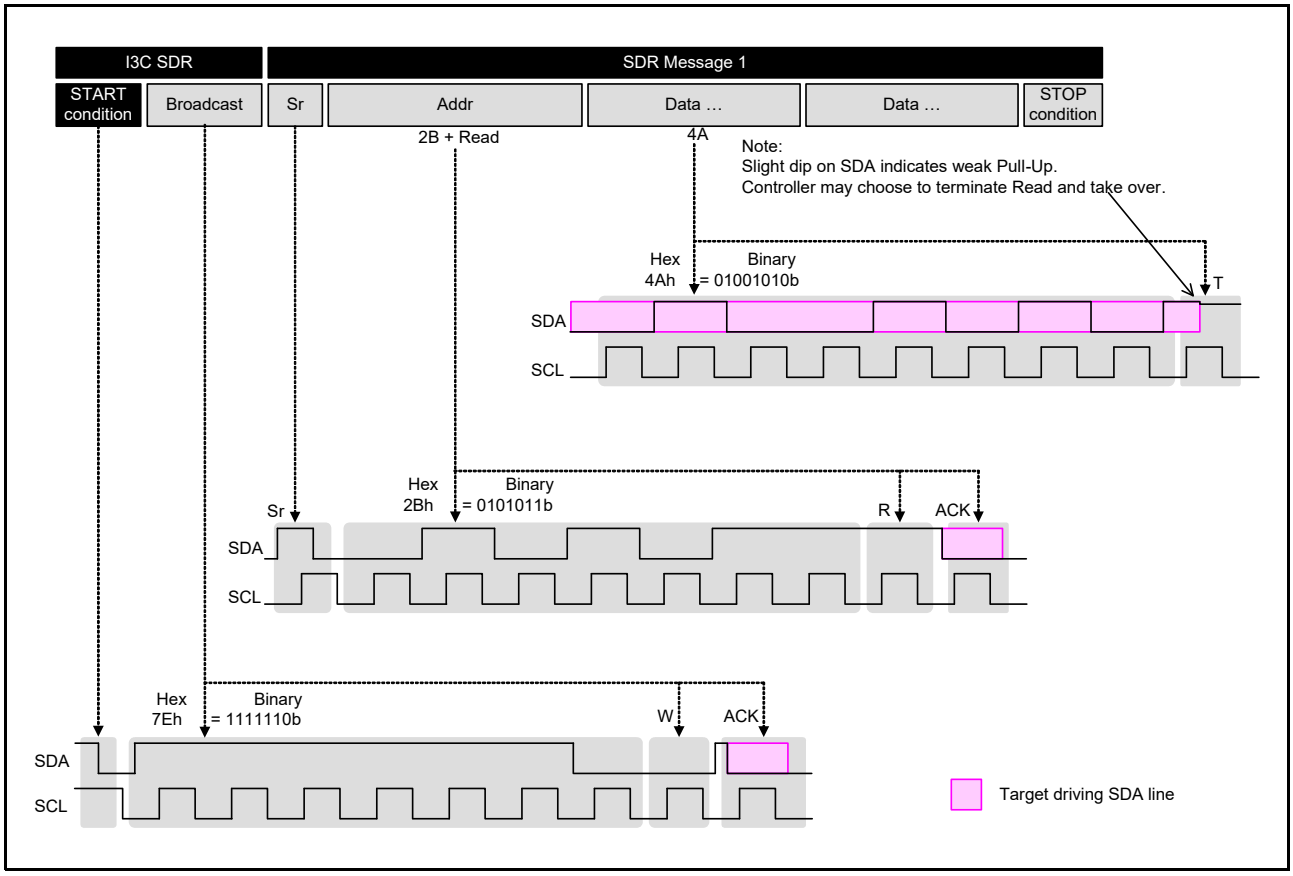


Figure 35.7 Example Communication Using I3C Mode SDR with Broadcast Address (7Eh)

Figure 35.8 illustrates example communication using I3C Single Data Rate (SDR) mode without Broadcast Address (7Eh). It shows the Controller reading a byte of data from the Target at Address 2Bh in SDR Mode.
From the Bus Free condition, the Controller issues a START condition, then the Address of the Target (2Bh) it wants to read followed by RnW (1 for Read). The Controller then turns on a pull-up resistor and goes to Open Drain, allowing the Target to acknowledge by pulling the SDA line Low. At this point, the Controller continues to toggle the SCL line and release the SDA line, allowing the Target to drive SDA to send one byte of data (4Ah) followed by T. T = 1 informs the Controller that there is additional data, whereas T = 0 signals the end. Here there is additional data, so the Target drives SDA High until SCL goes High, at which time it releases SDA. The Controller has the option of holding SDA High with a weak pull-up, which signals to the Target that the Controller allows another byte to be transmitted, or to pull SDA Low (while SCL is High – hence a repeated START condition), which would signal to the Target that the Controller has terminated the Read and is taking over.

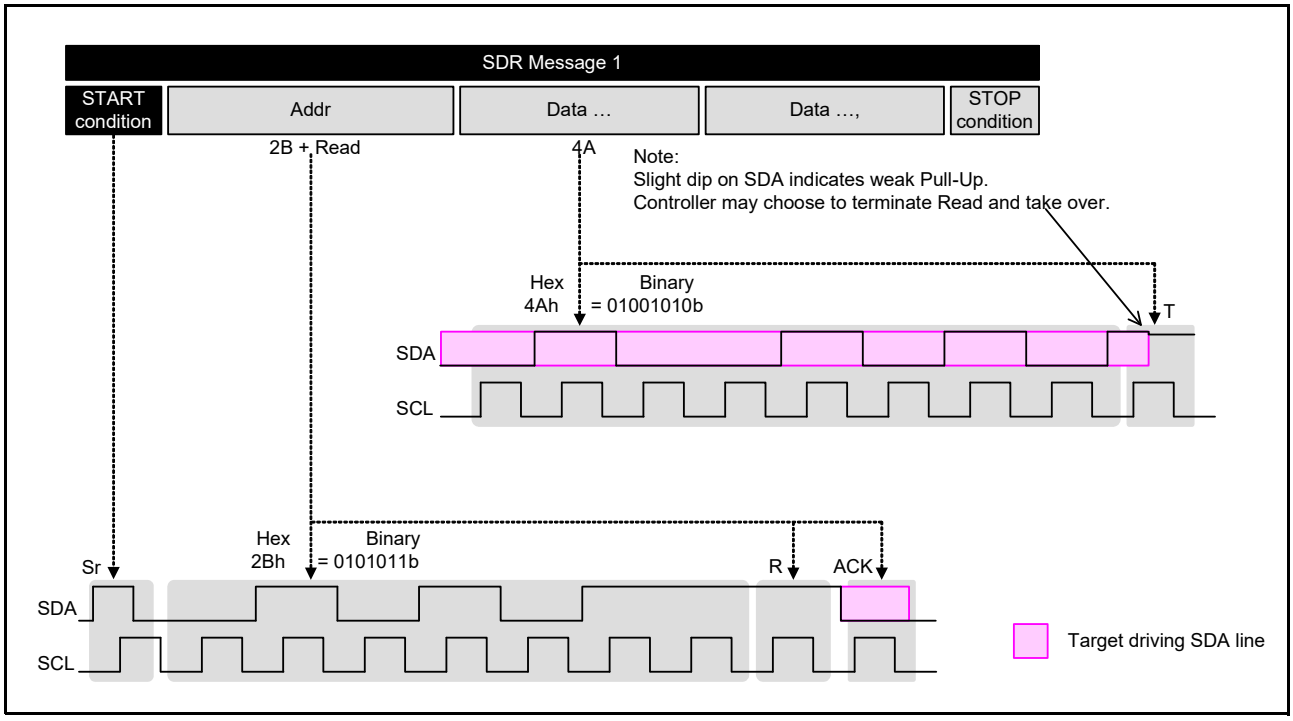


Figure 35.8 Example Communication Using I3C Mode SDR without Broadcast Address (7Eh)

Figure 35.9 shows the Controller issuing a Direct CCC to a single Target. This particular command (GETPID) reads the Provisioned ID of a Target.

From the Bus Free condition, the Controller issues a START condition by driving the SDA line Low while keeping the SCL line High. It then issues the Broadcast Address (7Eh) followed by RnW (0 for Write). Then the Controller turns on a pull-up resistor and goes to Open Drain. All Targets ACK by pulling SDA Low (in the Figure, pink fill means the Targets are in control of SDA at this time). The Controller then issues the Direct Common Command Code for GETPID (8Dh) followed by parity bit T (odd parity = 1 for 8Dh) then the 7-bit Dynamic Address of the Target (chosen arbitrarily here to be 2Bh) followed by a RnW bit (1 for Read). Then the Controller turns on a pull-up resistor and goes to Open Drain, allowing the Target at Address 2Bh to ACK by pulling SDA Low, which tells the Controller that the Target acknowledges the command and will comply. (Alternatively, the Target may NACK by not pulling SDA Low, which would inform the Controller that the Target will not comply – in this case, that an error occurred.) Following the ACK the Target outputs its 48-bit PID one byte at a time, and then the Controller issues a repeated START condition (this part of the waveform sequence is not shown in the Figure).

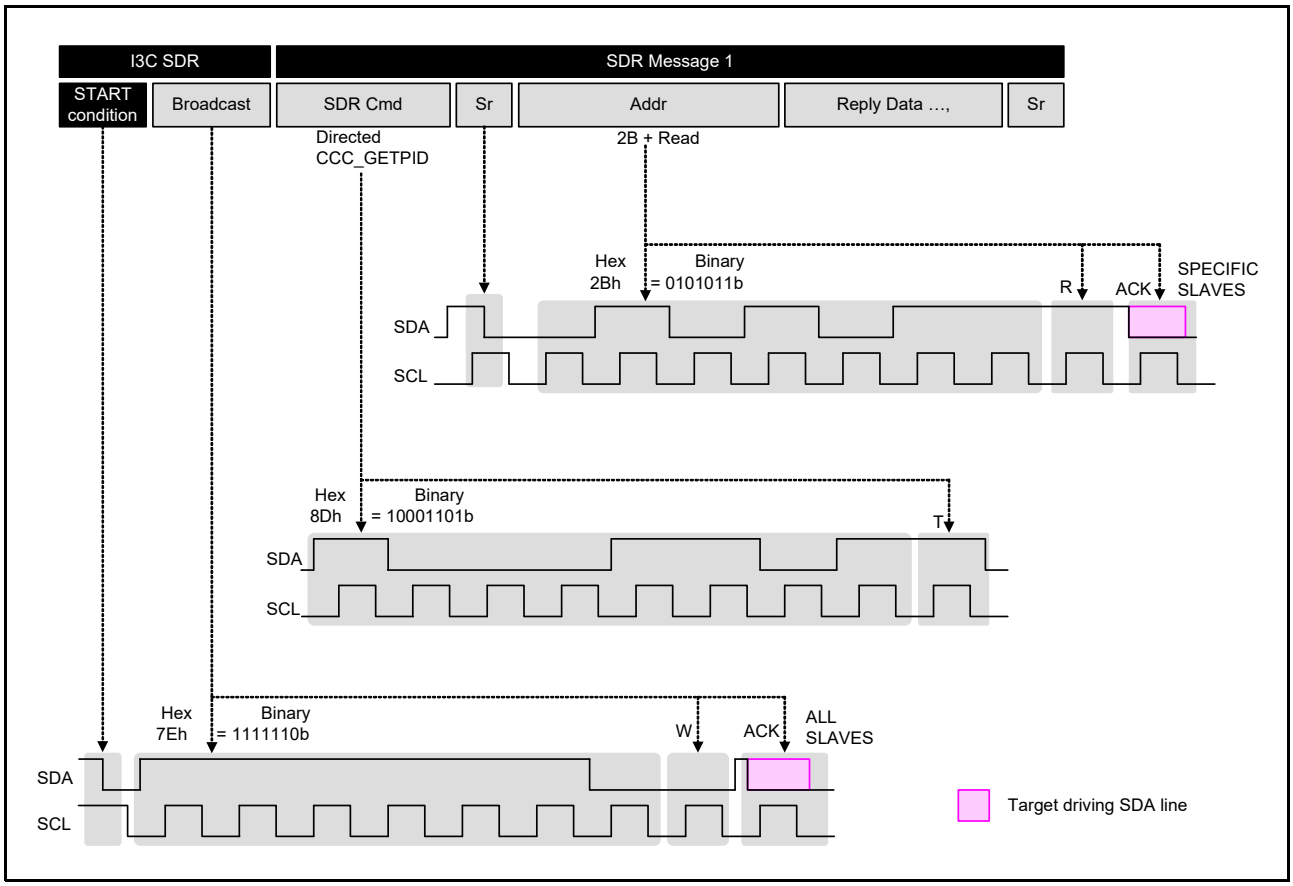


Figure 35.9 Example Communication Using I3C Mode SDR with Direct CCC

Figure 35.10 illustrates example SDR communication with a Broadcast CCC. The command used in this example sets the Maximum Read Length of all Targets to 43 bytes (002Bh).
From the Bus Free condition, the Controller issues a START condition by driving the SDA line Low while keeping the SCL line High. It then issues the Broadcast Address (7Eh) followed by RnW (0 for Write). Then the Controller turns on a pull-up resistor and goes to Open Drain. All Targets ACK by pulling SDA Low (in the Figure, pink fill means the Targets are in control of SDA at this time). The Controller then issues the Broadcast Common Command Code for SETMRL (0Ah) followed by parity bit T (odd parity = 1 for 0Ah), and then 2 data bytes (MSB first) to define the maximum number of bytes that can be read from a Target in a single read operation. Each data byte is followed by a T bit (parity bit – odd parity). After this the Controller issues a repeated START condition.

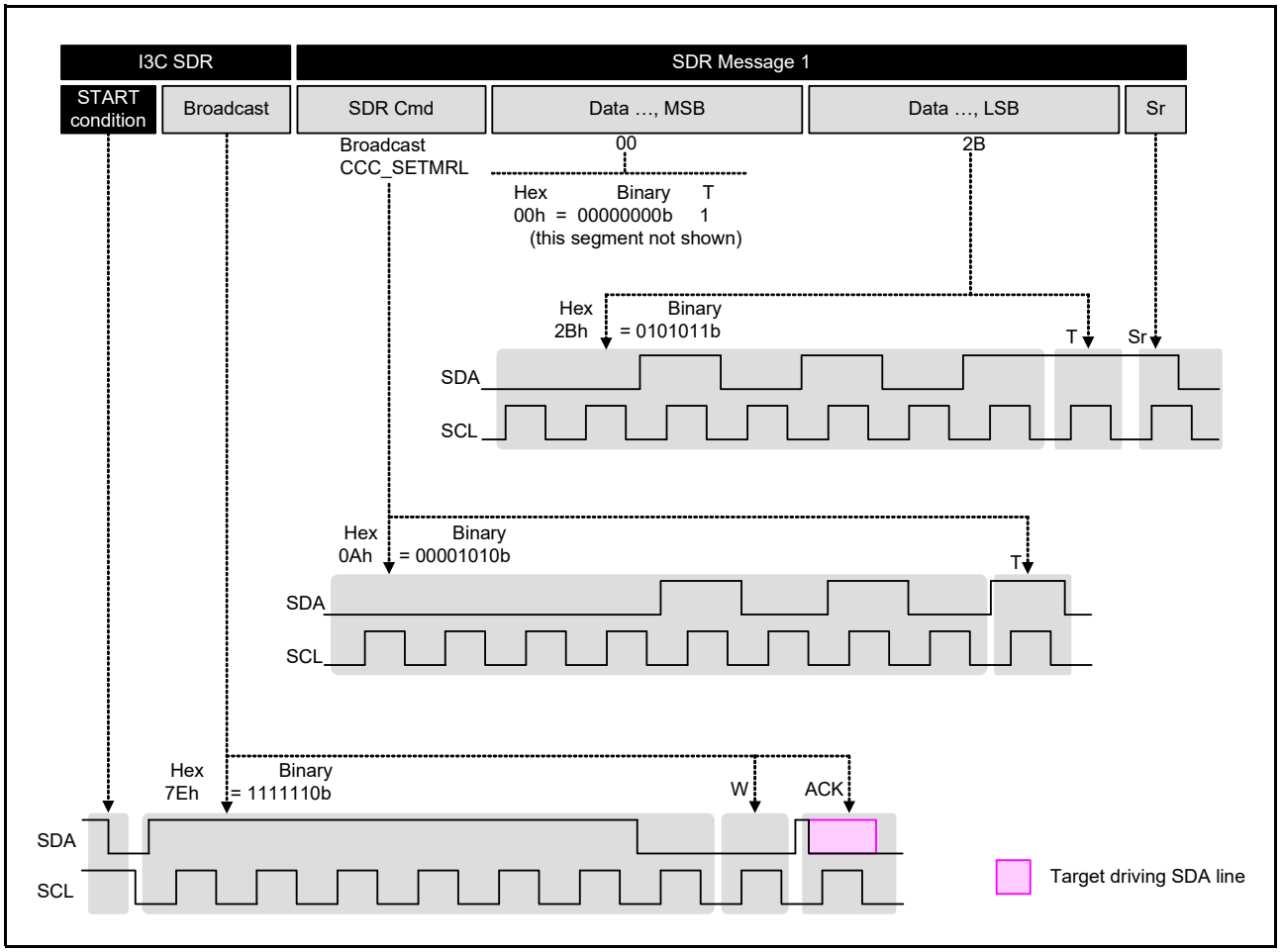


Figure 35.10 Example Communication Using I3C Mode SDR with Broadcast CCC

35.4.2.2 Bus Conditions

RI3C defines three distinct conditions in which the I3C bus shall be considered inactive: Bus Free, Bus Available, and Bus Idle (see Figure 35.11).

(1) Bus Free Condition

State on the I3C bus where both the SCL line and the SDA line are High for at least the period set by ICBFTR register.

(2) Bus Available Condition

State on the I3C bus where both the SCL line and the SDA line are High for at least the period set by ICBATR register.

A Target may only issue a START Request (For example, for an IBI, or for a CRR) after a Bus Available condition.

(3) Bus Idle Condition

State on the I3C bus where both the SCL line and the SDA line are High for at least the period set by ICBITR register.

A Target may only issue a START Request (For example, for a Hot-Join) after a Bus Idle condition.

Specifications are as follows. Bus Idle period needs to be the largest.

ICBFTR < ICBATR < ICBITR

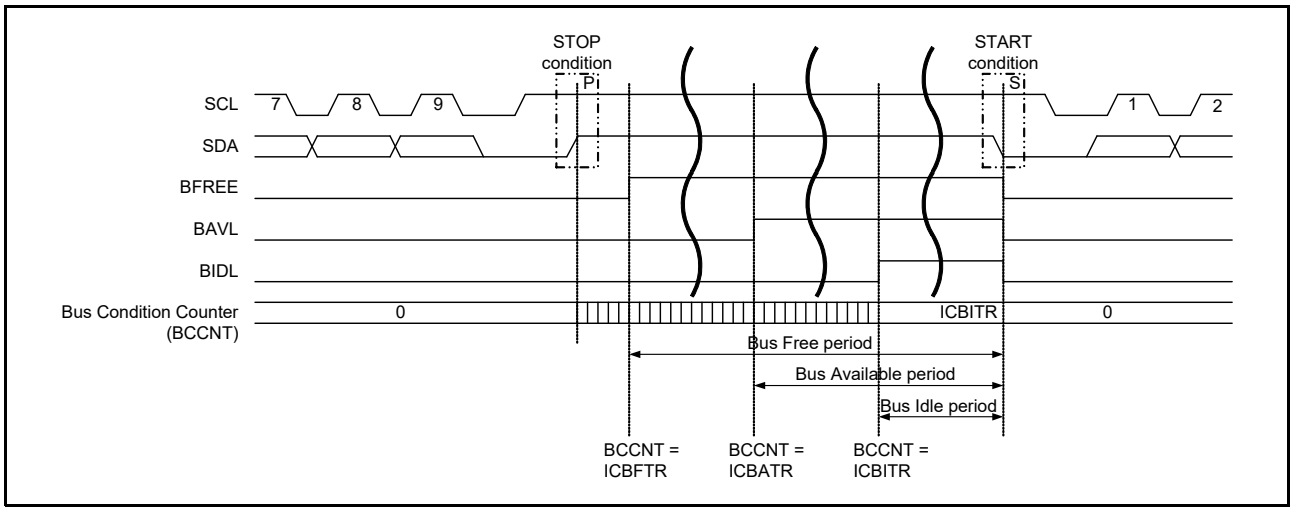


Figure 35.11 Bus Conditions

35.4.3 Initial Settings

Before starting data transmission and reception, initialize the RI3C according to the procedure in Figure 35.12.

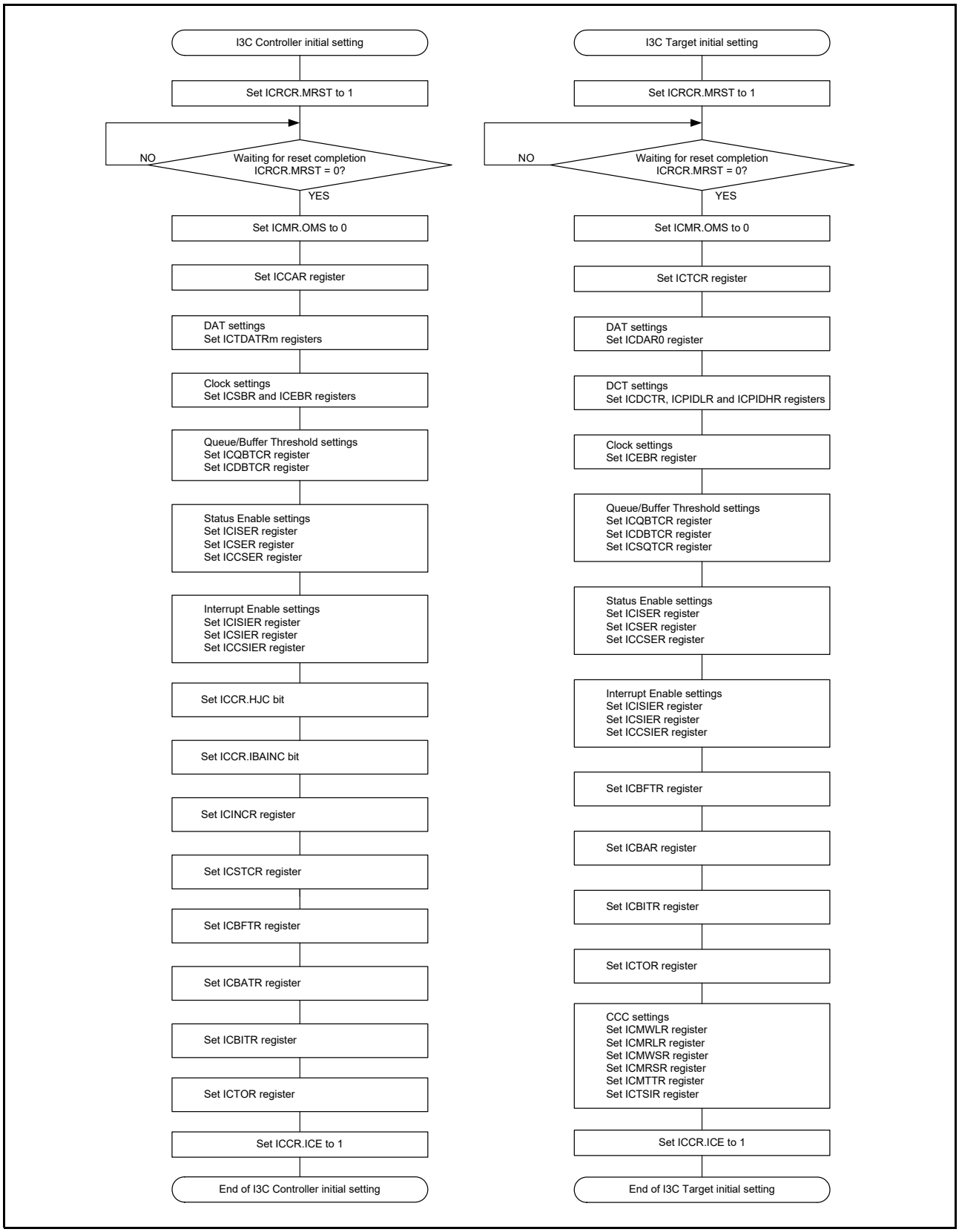


Figure 35.12 Example of RI3C Initialization Flowchart

35.4.4 I3C Communication Flow

Figure 35.13 illustrates how I3C communication is initiated:

- All I3C communication occurs within a frame. The frame begins with a START condition, followed by one or more transfers, and a STOP condition.

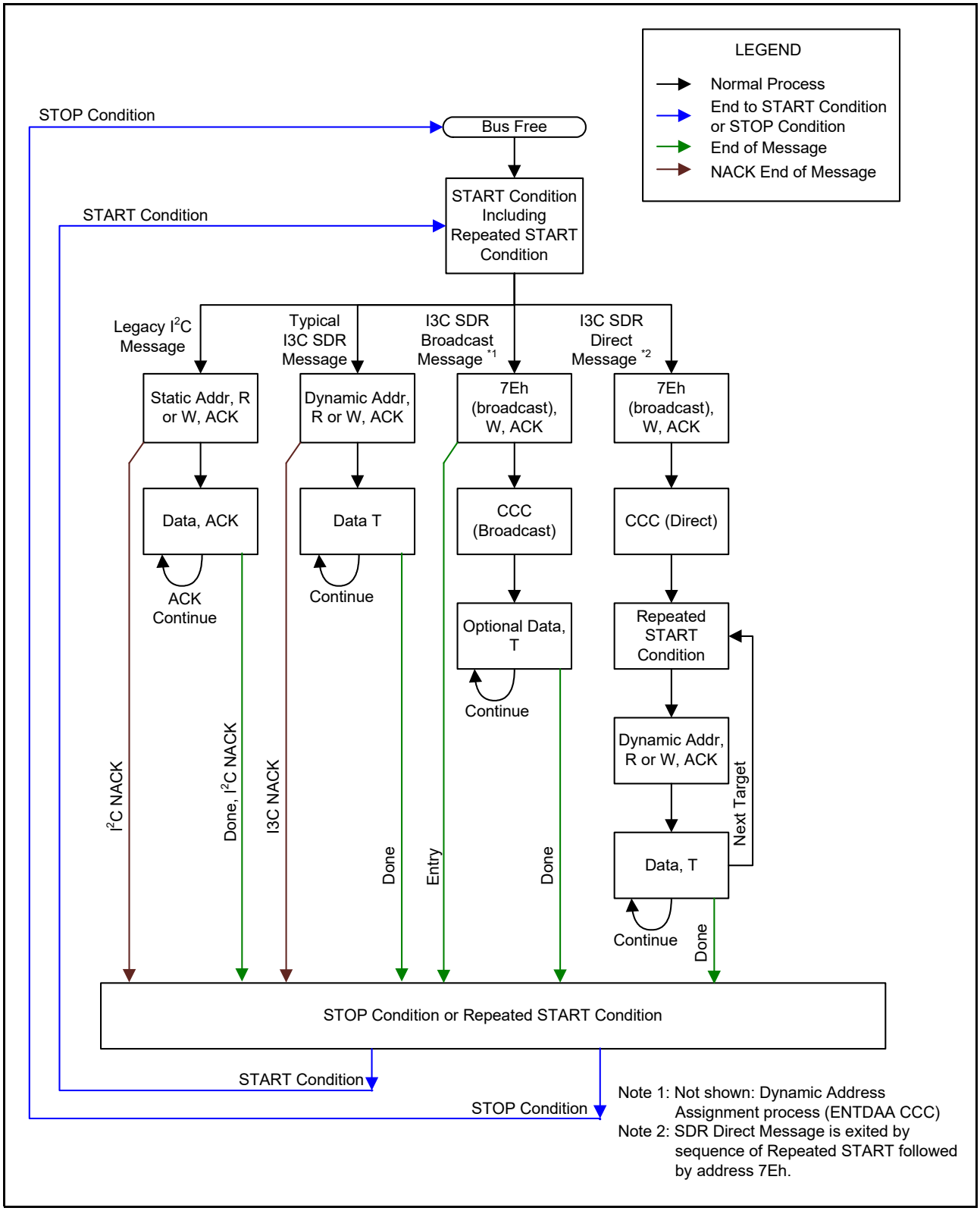


Figure 35.13 I3C Communication Flow

I3C is based on a frame encapsulation approach. A frame includes a data payload. The transfer protocol for the data payload is SDR. Frames are bordered by I^2C-like bus management.
The I3C frame always includes at least the START condition, the Header, the Data, and the STOP condition. The Header following a START condition allows for bus arbitration. The Controller uses the Header to address Target device(s). Target device(s) may use the Header Arbitration for multiple purposes: for IBI, for Hot-Join, and for Secondary Controller functionality.
I3C allows only one Controller to have control of the I3C bus at a time. Mechanisms for handoff of the Controller role from one device to another device are provided.

35.4.5 I3C Controller Operation

35.4.5.1 Dynamic Address Assignment Procedure

After initializing RI3C, first execute Dynamic Address Assignment Procedure for I3C Target connected on the I3C bus. The following describes the procedure.

(1) Initial setting (refer to section 35.4.3, Initial Settings for details)

(2) Execute Dynamic Address Assignment with ENTDAA or SETDASA Common Command Code (CCC) for I3C Target set in DAT (ICTDATRm register).
Write Command Descriptor (address assignment command) to command queue via the ICCQR register.

(3) When Command Descriptor is written to command queue, Transaction is issued on I3C bus.

(4) When ENTDAA is specified for the CMD field of address assignment command:
Executes Dynamic Address Assignment for I3C Targets corresponding to the number of the ICTDATRm registers specified by the DEV_COUNT filed starting with the ICTDATRm register indexed by the DEV_INDEX filed of address assignment command.
When SETDASA is specified for the CMD filed of address assignment command:
Execute Dynamic Address Assignment for I3C Target indicated by the ICTDATRm register indexed by the DEV_INDEX filed of address assignment command.

(5) In case of ENTDAA, the Provisioned ID, BCR, DCR transmitted from I3C Target is stored in Receive Data Buffer (BCR is also automatically stored in the ICTDCTRm register).
Read the Provisioned ID, BCR, and DCR from the receive data buffer via the ICDR register with a receive data full interrupt.

(6) When execution of Dynamic Address Assignment is completed, issue STOP condition and store the Response Descriptor into the response queue.

(7) Read the Response Descriptor via the ICRQR register and check the status.

(8) Check whether the value of the DATA_LENGTH filed of the Response Descriptor matches the value of the DEV_COUNT filed of the address assignment command.

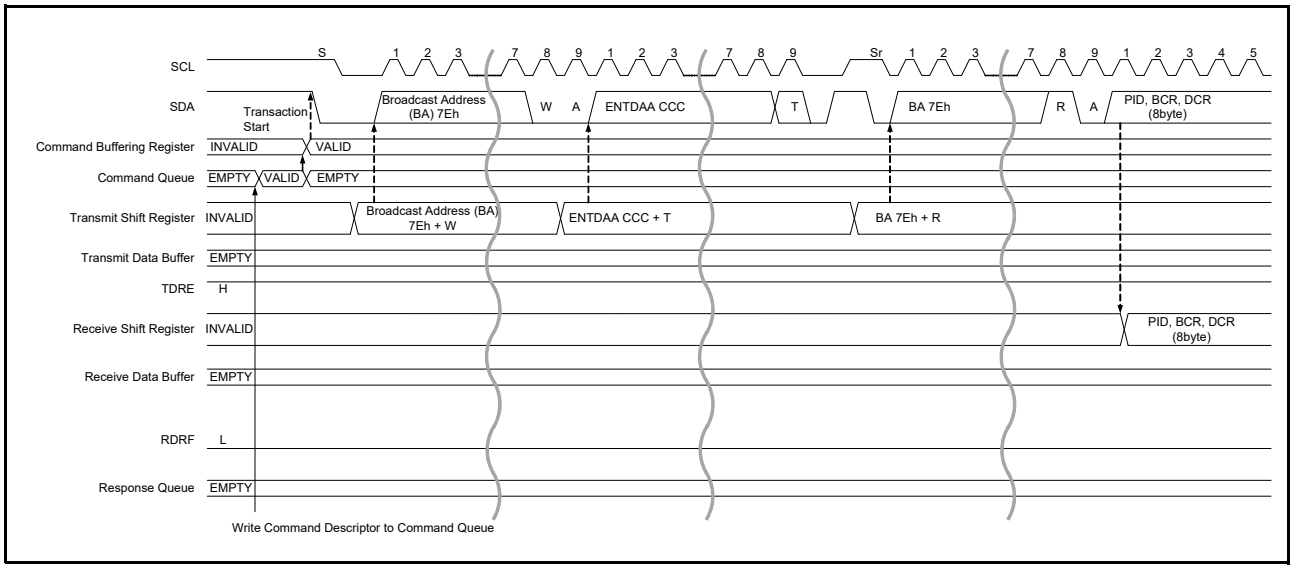


Figure 35.14 Dynamic Address Assignment Procedure (ENTDAA CCC) Timing (1/3)

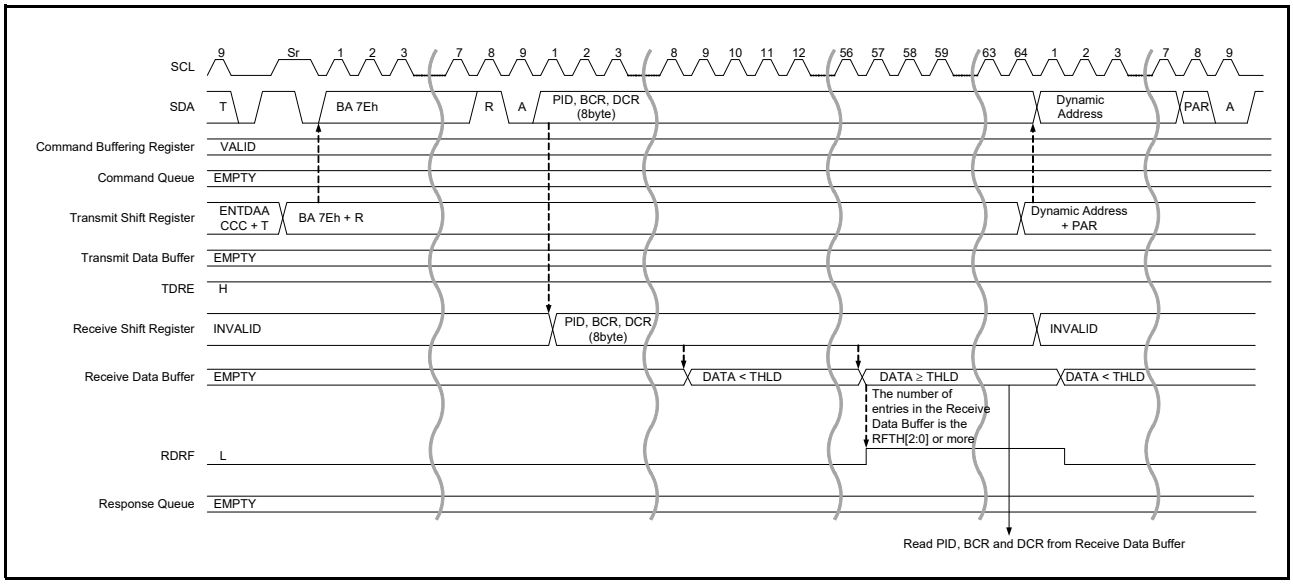


Figure 35.15 Dynamic Address Assignment Procedure (ENTDAA CCC) Timing (2/3)

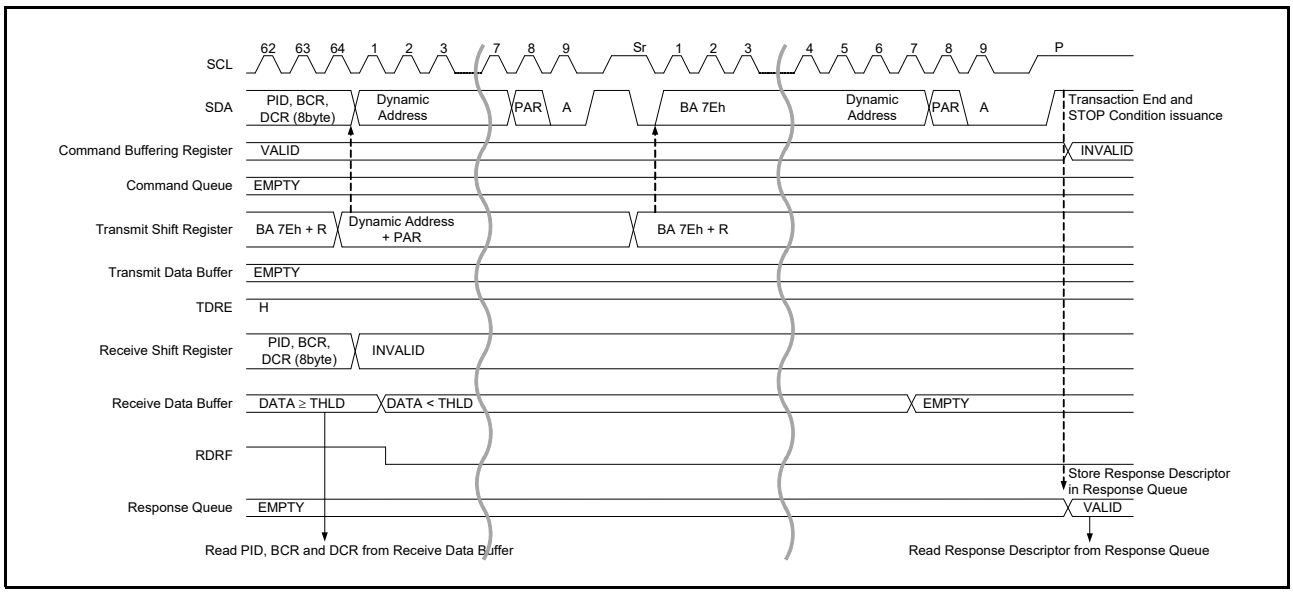


Figure 35.16 Dynamic Address Assignment Procedure (ENTDAA CCC) Timing (3/3)

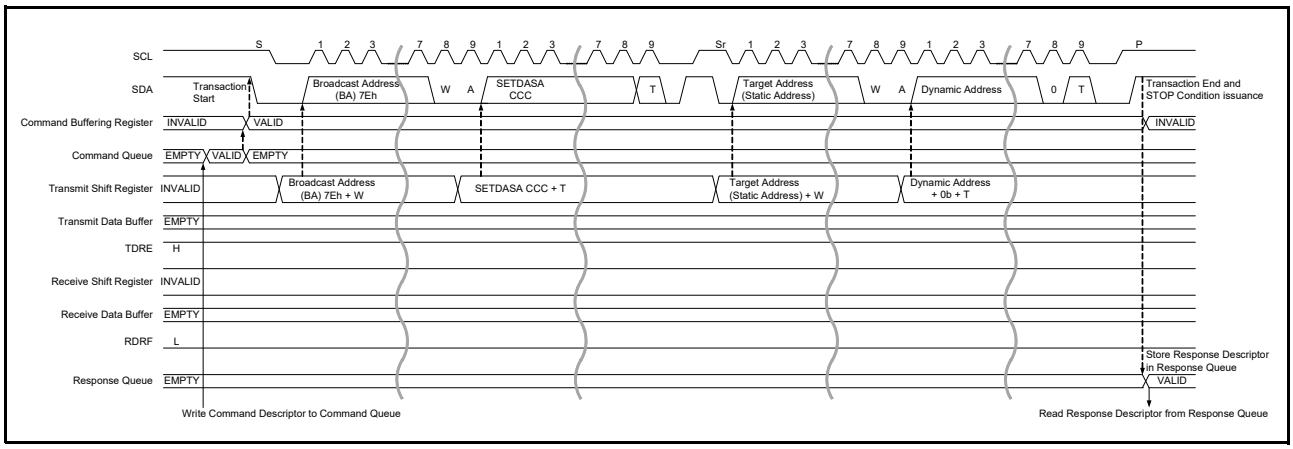


Figure 35.17 Dynamic Address Assignment Procedure (SETDASA CCC) Timing

35.4.5.2 SDR Write Transfer

(1) Write data for transmission to the Transmit Data Buffer via the ICDR register.
(2) Write a Command Descriptor (immediate data transfer command or regular data transfer command or combo transfer command) for Data Transfer to the command queue via the ICCQR register.
(3) When Command Descriptor is written to command queue, transaction is issued on I3C bus.
When NACK is received with the Address header, transaction of the same command is automatically issued according to the NACK Retry Count value (ICTDATRm.NACKRC[1:0]) of DAT.
(4) If data for transmission still remain, write data for transmission by a transmit data empty interrupt to the Transmit Data Buffer via the ICDR register.
(5) When data transmission for the number of bytes specified in the DATA_LENGTH field of the Command Descriptor is completed, the repeated START condition or STOP condition is issued and the Response Descriptor is stored in the response queue.
(6) Read the Response Descriptor via the ICRQR register and check the status.
(7) Check that the value of the DATA_LENGTH field of the Response Descriptor is 0.

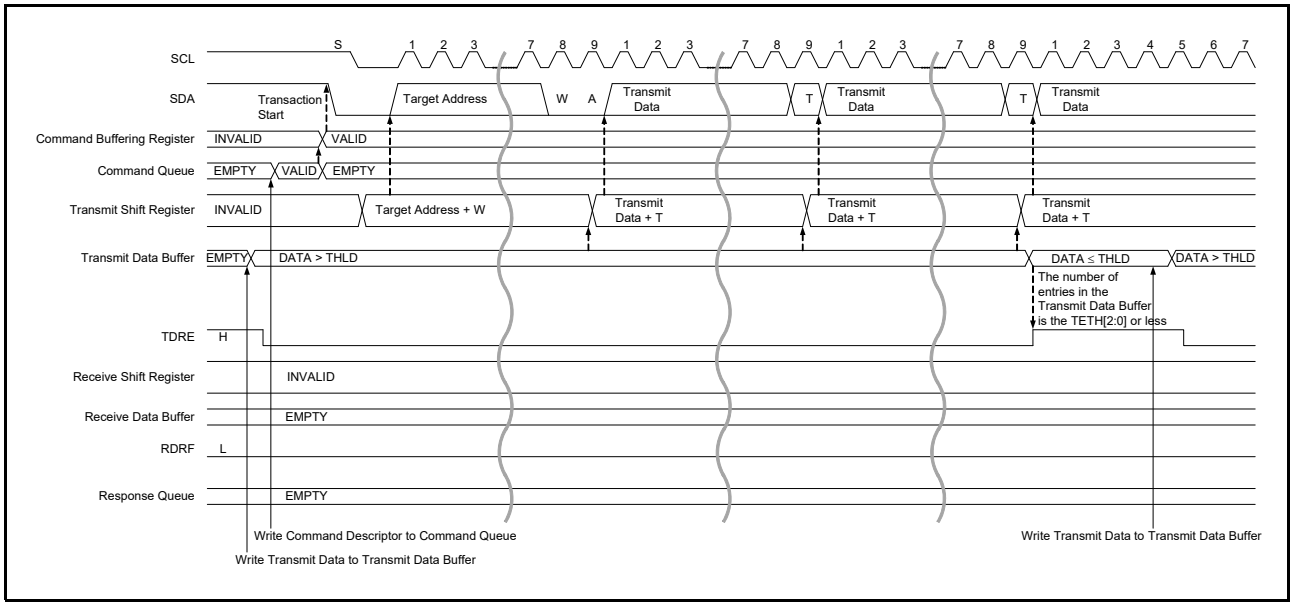


Figure 35.18 SDR Write Transfer Timing (1/2)

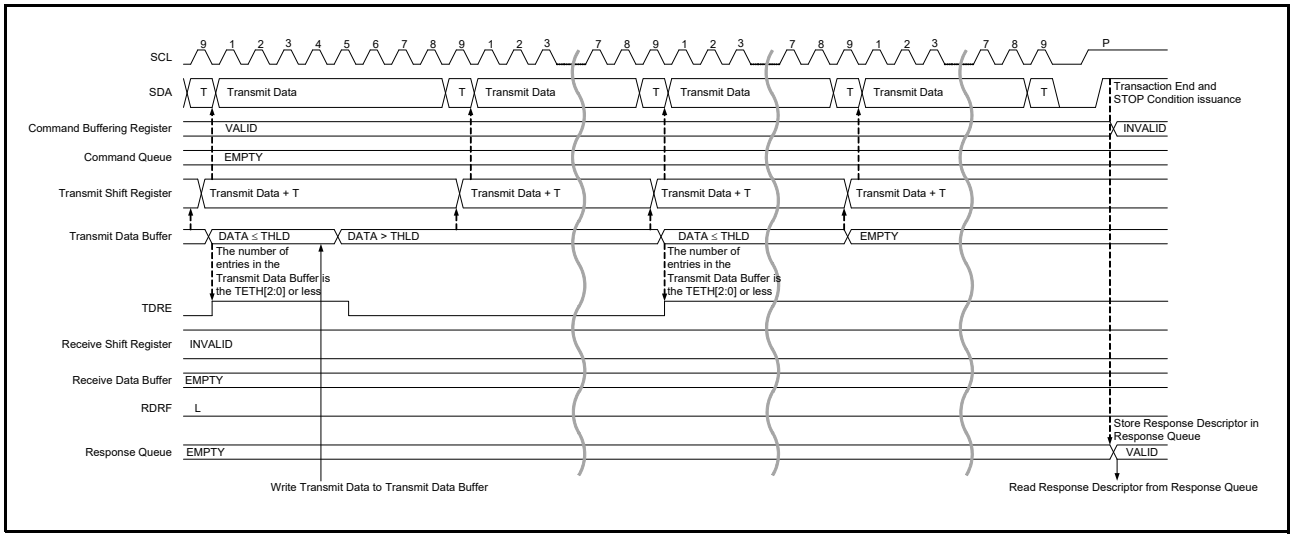


Figure 35.19 SDR Write Transfer Timing (2/2)

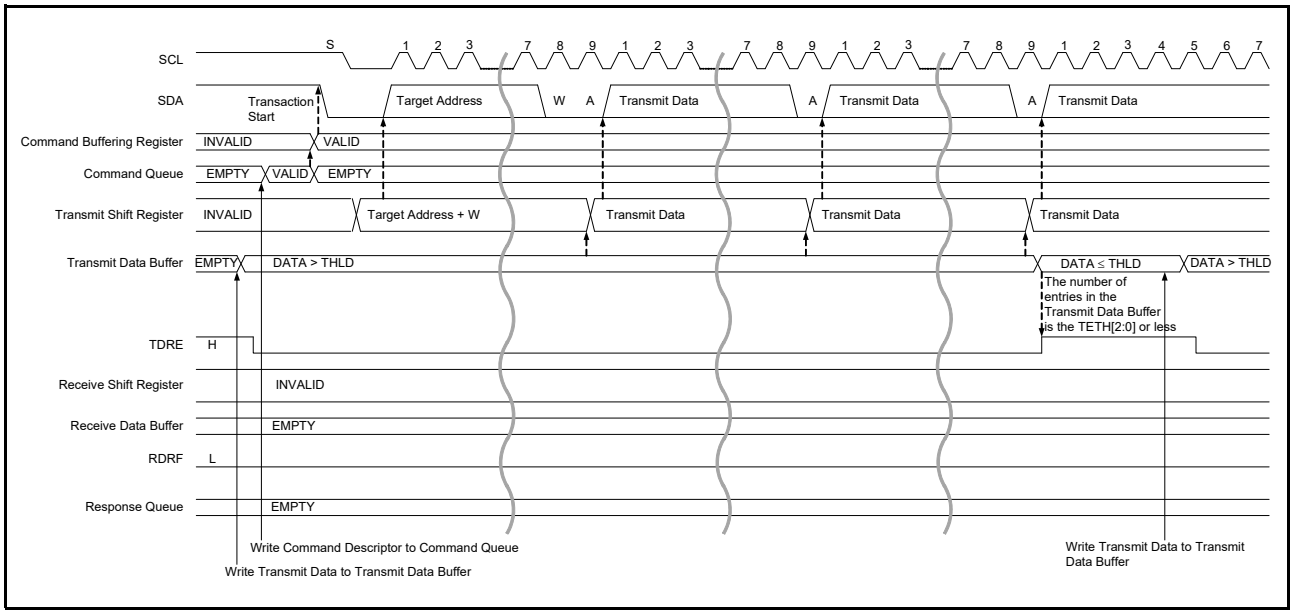


Figure 35.20 Legacy I²C Write Transfer Timing (1/2)

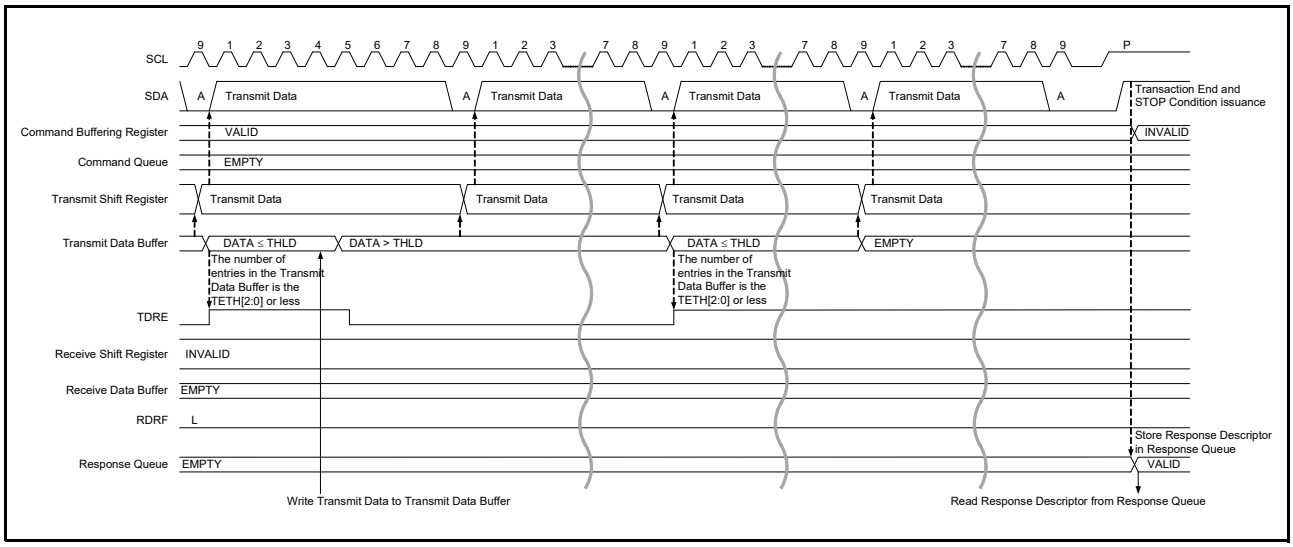


Figure 35.21 Legacy I²C Write Transfer Timing (2/2)

35.4.5.3 SDR Read Transfer

(1) Write a Command Descriptor (immediate data transfer command or regular data transfer command or combo transfer command) for Data Transfer to the command queue via the ICCQR register.

(2) When Command Descriptor is written to command queue, transaction is issued on I3C bus.
When NACK is received with the Address header, transaction of the same command is automatically issued according to the NACK Retry Count value (ICTDATRm.NACKRC[1:0]) of DAT.

(3) Data received from the I3C Target is stored in the Receive Data Buffer.

(4) In the receive data full interrupt, read the received data from the Receive Data Buffer via the ICDR register.

(5) SDR:
Detecting Low in T-bit or receiving Data for the number of bytes specified in the DATA_LENGTH filed of Command Descriptor is completed, issue repeated START condition or STOP condition and store the Response Descriptor into the response queue.
Legacy I^2C Message:
When data reception for the number of bytes specified in the DATA_LENGTH filed of Command Descriptor is completed, NACK is issued. After that, issue a repeated START condition or STOP condition and store the Response Descriptor into the response queue.

(6) Read the Response Descriptor via the ICRQR register and check the status.

(7) Check whether the value of the DATA_LENGTH filed of the Response Descriptor matches the data length setting value of the Command Descriptor.

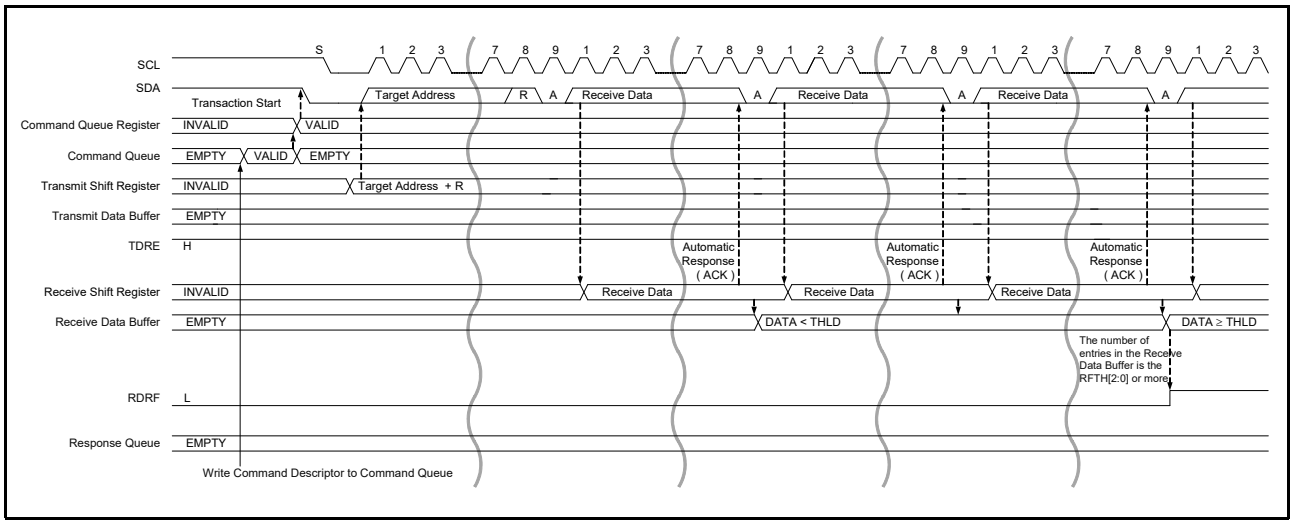


Figure 35.22 SDR Read Transfer Timing (1/2)

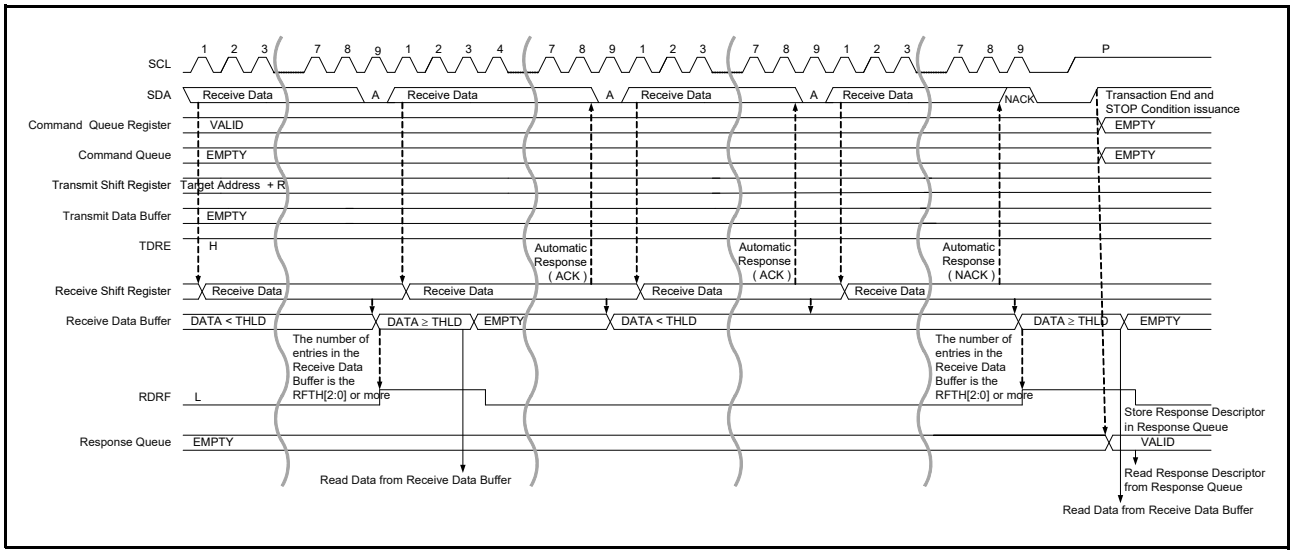


Figure 35.23 SDR Read Transfer Timing (2/2)

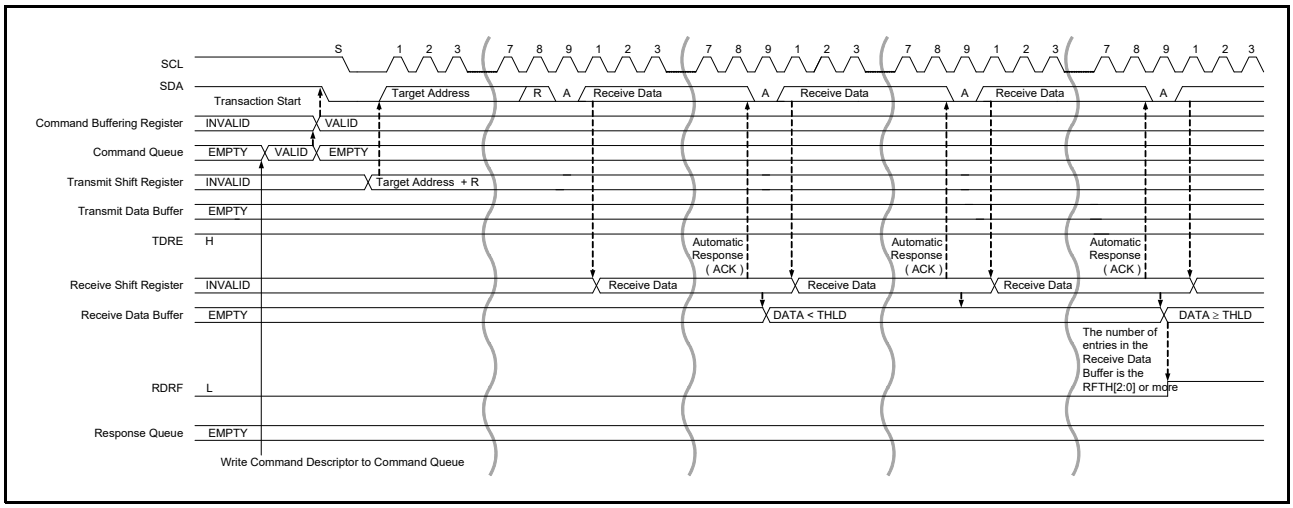


Figure 35.24 Legacy I²C Read Transfer Timing (1/2)

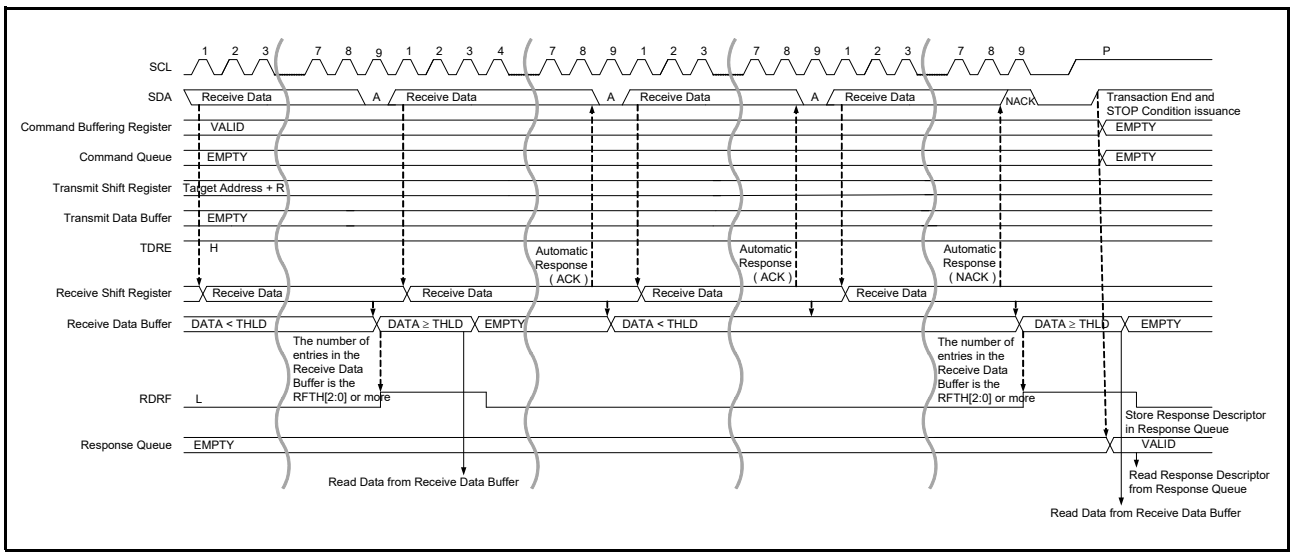


Figure 35.25 Legacy I²C Read Transfer Timing (2/2)

35.4.5.4 IBI Transfer

(1) Write Command Descriptor to the command queue and issue Transaction on I3C bus.
If START Request (SDA Low Drive) is issued from the Target device, RI3C drives SCL to Low and completes START condition.
Thereafter, the SCL is supplied and IBI Request is received.
(2) In Target Address and RnW field in the Address header, if losing Arbitration by issuing IBI from I3C Target, stop issuing Transaction.
(3) According to section 35.5.3, IBI, detect IBI and process.
(4) In the IBI queue empty/full interrupt routine, read the IBI Status Descriptor from the IBI queue via the ICIQR register and check the status.
When detected a Target Interrupt Request and responded with ACK, Read the IBI Data for the Data Length indicated by the DATA_LENGTH filed of the IBI Status Descriptor from the IBI queue via the ICIQR register.
(5) Restart issuing Transaction of Command of Step1.

An example of the processing procedure after detection of IBI is shown below.
Processing procedure for detecting Controller Role Request and transferring Controller role to Secondary Controller
(1) If the I3C Secondary Controller wins the Arbitration, issue a DEFTGTS CCC and notify Target information to Secondary Controller.
(2) Issue a GETACCCR CCC and complete CCC by a STOP condition.

Note: • After transferring Controller role to Secondary Controller, to get Controller role again, issue a Controller Role Request according to section 35.4.5.4, IBI Transfer.
• After Controller Role Request is accepted by the Active Controller, to get Controller role again at receiving the GETACCCR CCC and complete CCC by a STOP condition.

Processing procedure when Hot-Join Event is detected
(1) Issue a Broadcast Command Code Enter Dynamic Address Assignment (ENTDAA) to start the Dynamic Address Assignment process.
(2) Issue a DEFTGTS CCC and notify Target information to Secondary Controller.

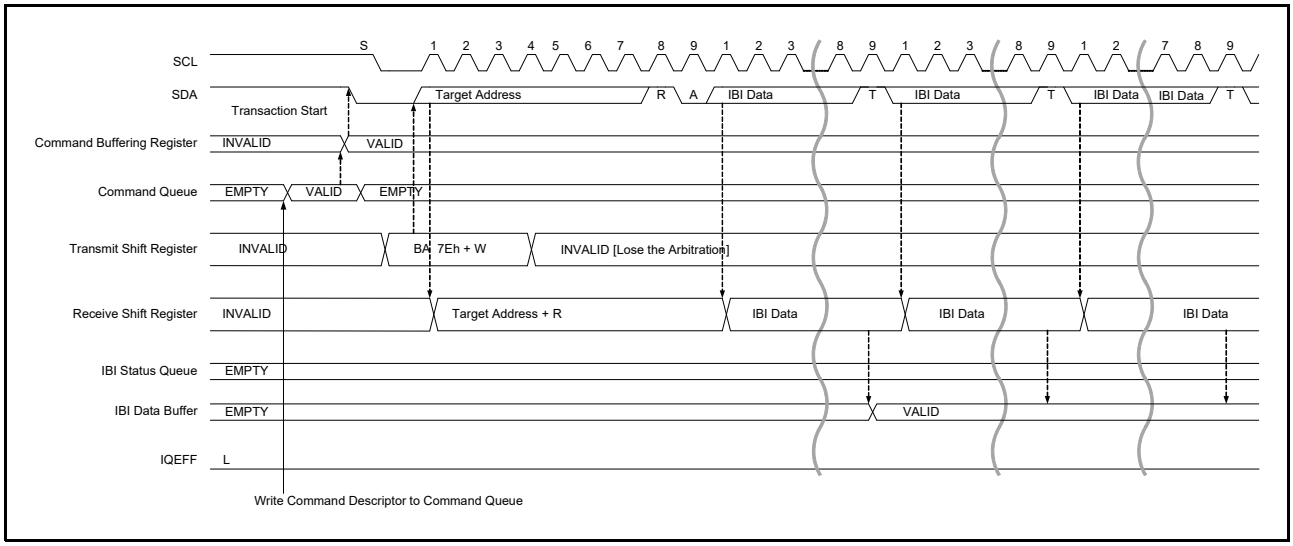


Figure 35.26 I3C Controller IBI Transfer Timing (1/2)

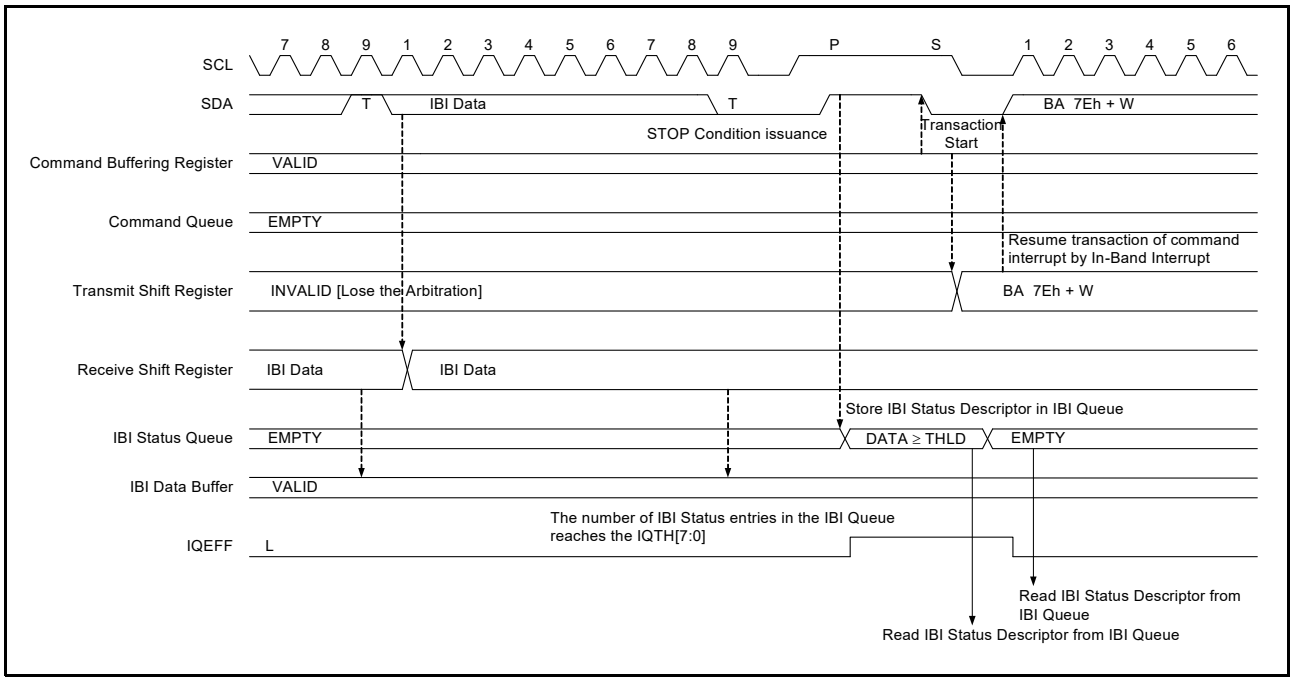
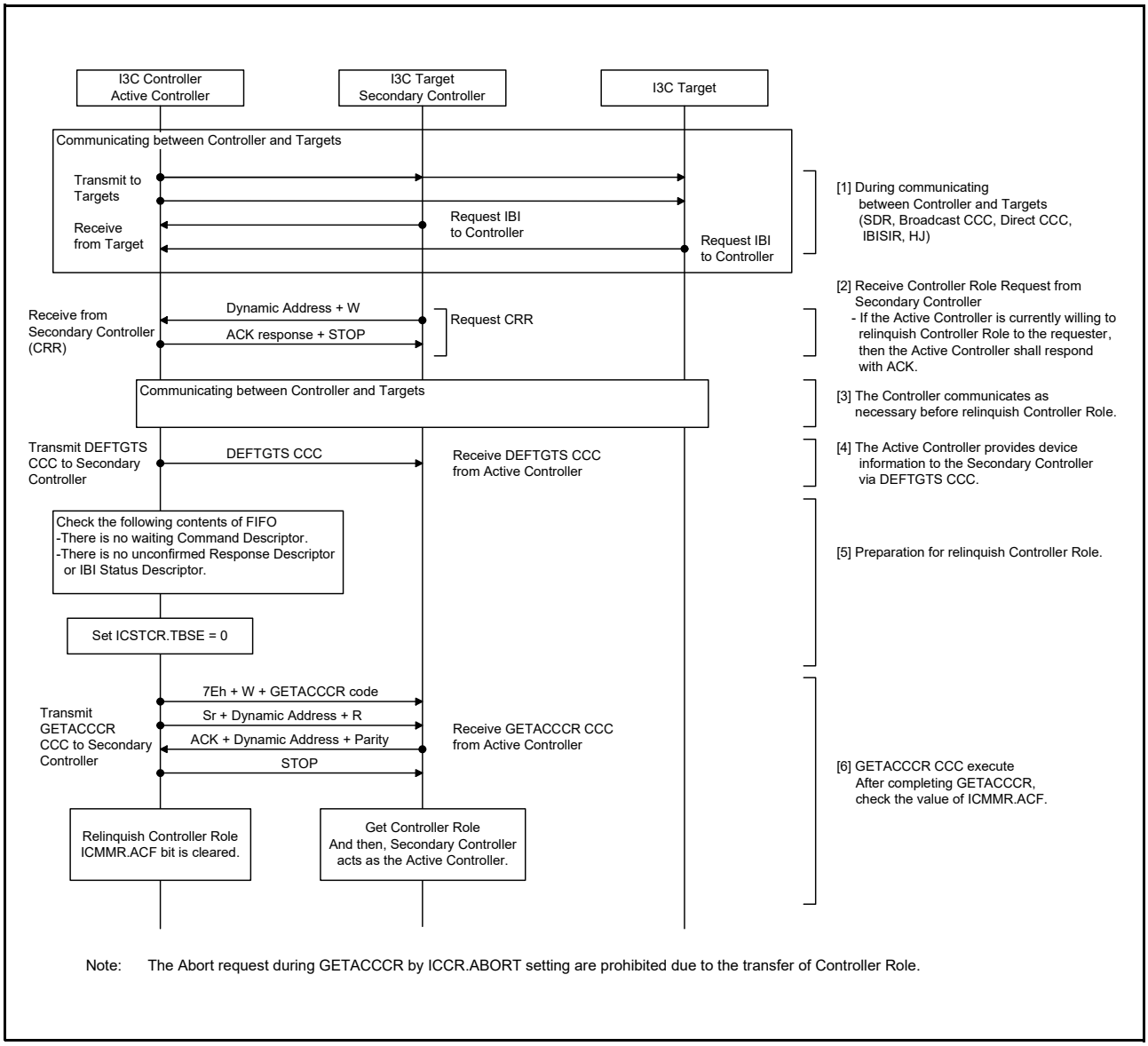


Figure 35.27 I3C Controller IBI Transfer Timing (2/2)

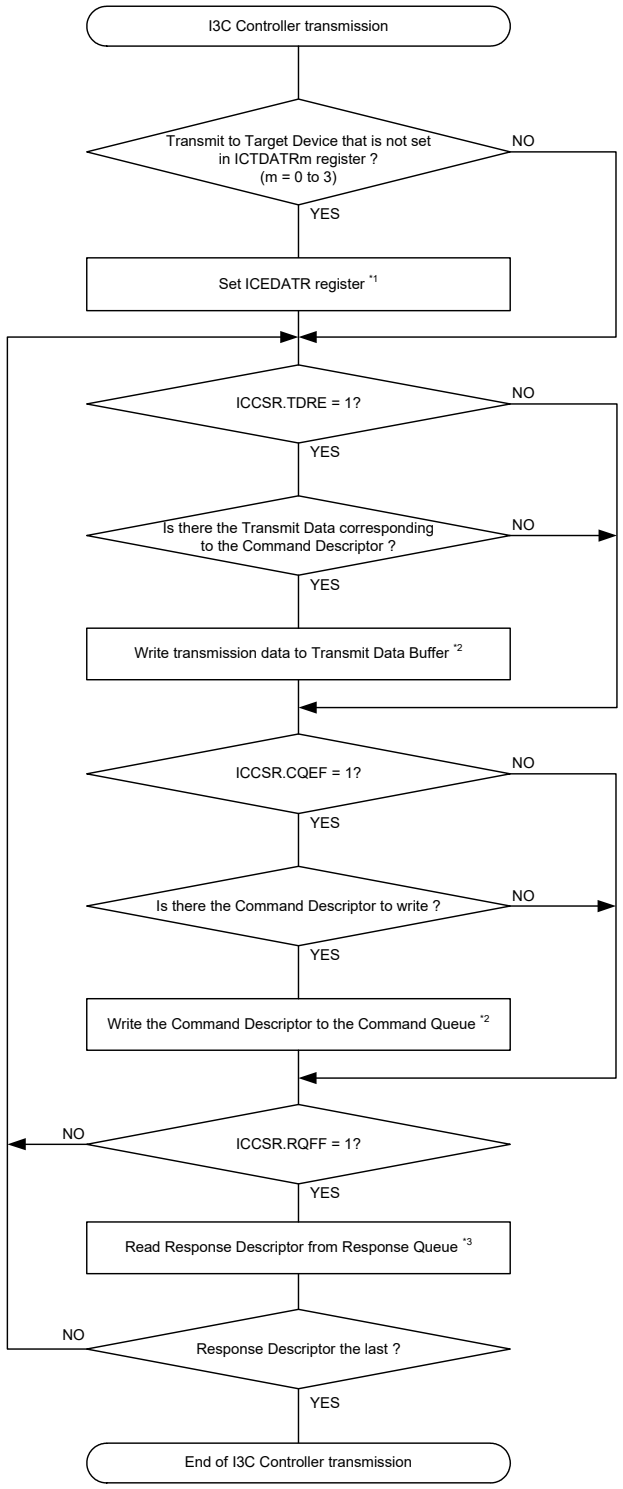


Note: The Abort request during GETACCCR by ICCR.ABORT setting are prohibited due to the transfer of Controller Role.

Figure 35.28 I3C Controller CRR Processing Flow

35.4.5.5 I3C Controller Transmission Flow (FIFO Buffer Transfer)

Controller transmission flow in FIFO buffer transfer is common to Legacy I^2C, SDR (Private Transfer, Broadcast CCC, Direct CCC).

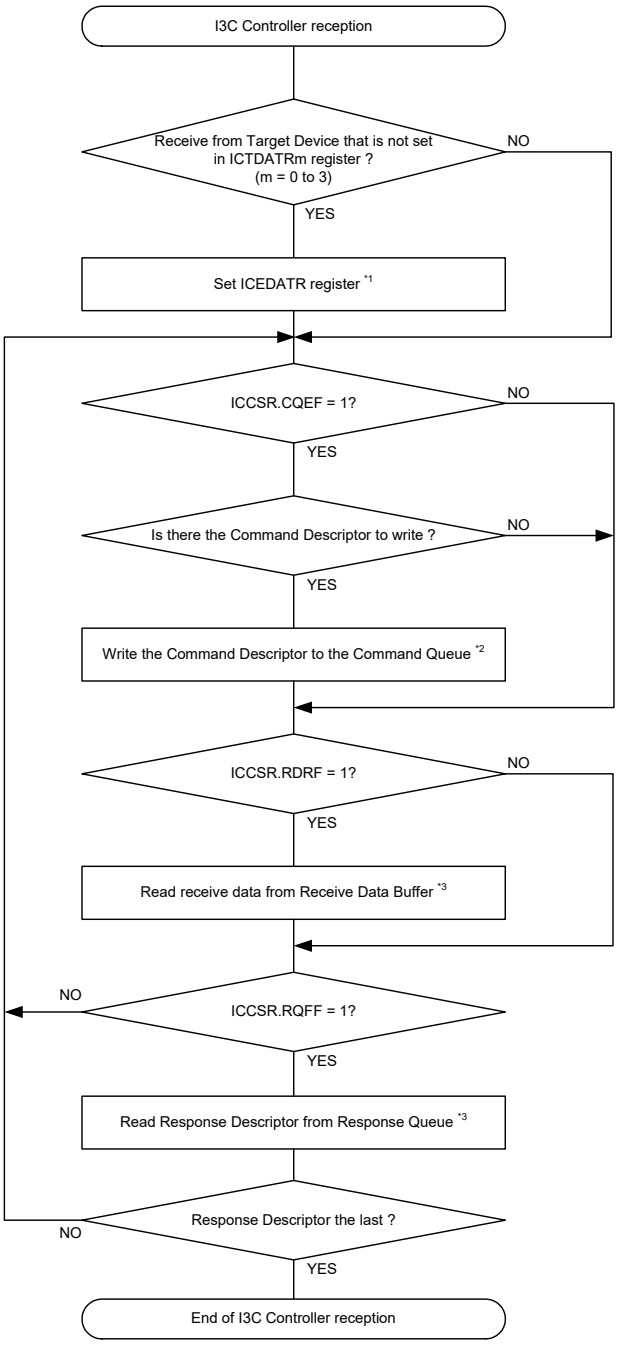


Note 1. Changing the ICEDATR register is prohibited until all Response Descriptors of command for the Target device set in the ICEDATR register are stored.
Note 2. In the case the register is write by a way other than DMAC/DTC, clear corresponding flags after write access.
Note 3. In the case the register is read by a way other than DMAC/DTC, clear corresponding flags after read access.

Figure 35.29 Example of I3C Controller Transmission Flowchart (FIFO Buffer Transfer)

35.4.5.6 I3C Controller Reception Flow (FIFO Buffer Transfer)

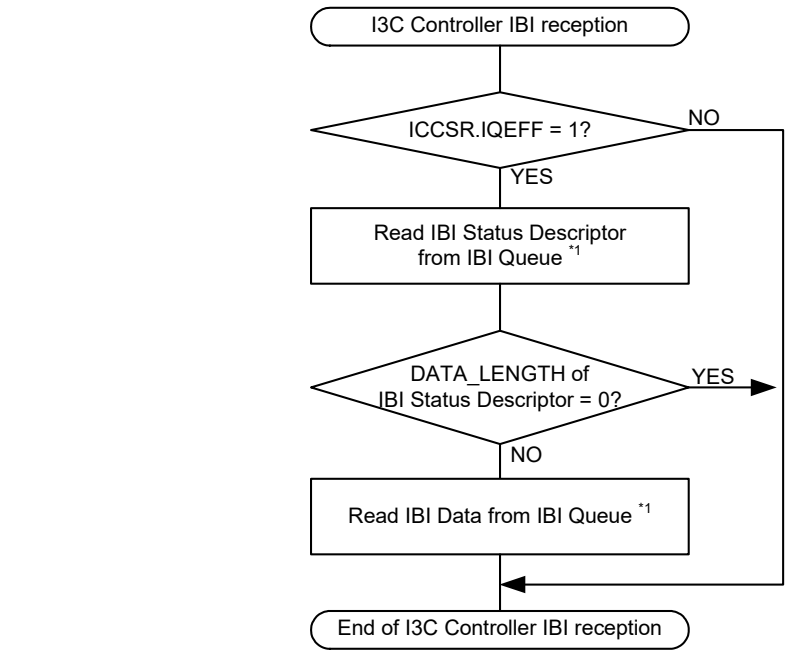
Controller reception flow in FIFO buffer transfer is common to Legacy I^2C, SDR (Private Transfer, Broadcast CCC, Direct CCC).



Note 1. Changing the ICEDATR register is prohibited until all Response Descriptors of command for the Target device set in the ICEDATR register are stored.
Note 2. In the case the register is write by a way other than DMAC/DTC, clear corresponding flags after write access.
Note 3. In the case the register is read by a way other than DMAC/DTC, clear corresponding flags after read access.

Figure 35.30 Example of I3C Controller Reception Flowchart (FIFO Buffer Transfer)

35.4.5.7 I3C Controller IBI Reception Flow



Note 1. In the case the register is read by a way other than DMAC/DTC, clear corresponding flags after read access.

Figure 35.31 Example of I3C Controller IBI Reception Flowchart

35.4.6 I3C Target Operation

35.4.6.1 Dynamic Address Assignment Procedure

After initializing RI3C, the I3C Controller first performs Dynamic Address Assignment Procedure.
The operation of RI3C during the Dynamic Address Assignment Procedure by ENTDAA CCC is described below.

(1) Initial setting (For details, refer to section 35.4.3, Initial Settings)
(2) When ENTDAA CCC is received, RI3C transmits Provisioned ID (the ICPIDHR register and bits b15 to b0 of the ICPIDLR register), BCR (bits b15 to b8 of the ICDCTR register), DCR (bits b7 to b0 of the ICDCTR register) until a Dynamic Address is assigned. (For details, refer to In case of Broadcast CCC (ENTDAA) of 35.5.1 CCC detection function.)
(3) When ENTDAA CCC is completed and a STOP condition is detected, Receive Status Descriptor is stored in the receive status queue.
(4) Read Receive Status Descriptor via the ICSQR register and check the status.
(5) Read the data for the data length indicated by the DATA_LENGTH filed of the Receive Status Descriptor from the Receive Data Buffer via the ICDR register.

Note: When multiple RI3C (I3C Target) are connected on the I3C bus assign Dynamic Addresses in the following order.
1. Set the ICPIDHR and ICPIDLR registers (6 bytes in total) of the RI3C (I3C Target) to a value (all 1 etc.) that has a lower priority than other Target devices by Dynamic Address Arbitration.
2. After setting the Static Address in the RI3C (I3C Target), assign the Dynamic Address using the SETDASA/ SETAASA command.
3. Assign a Dynamic Address to an I3C Target Device other than the RI3C (I3C Target) using the ENTDAA command.

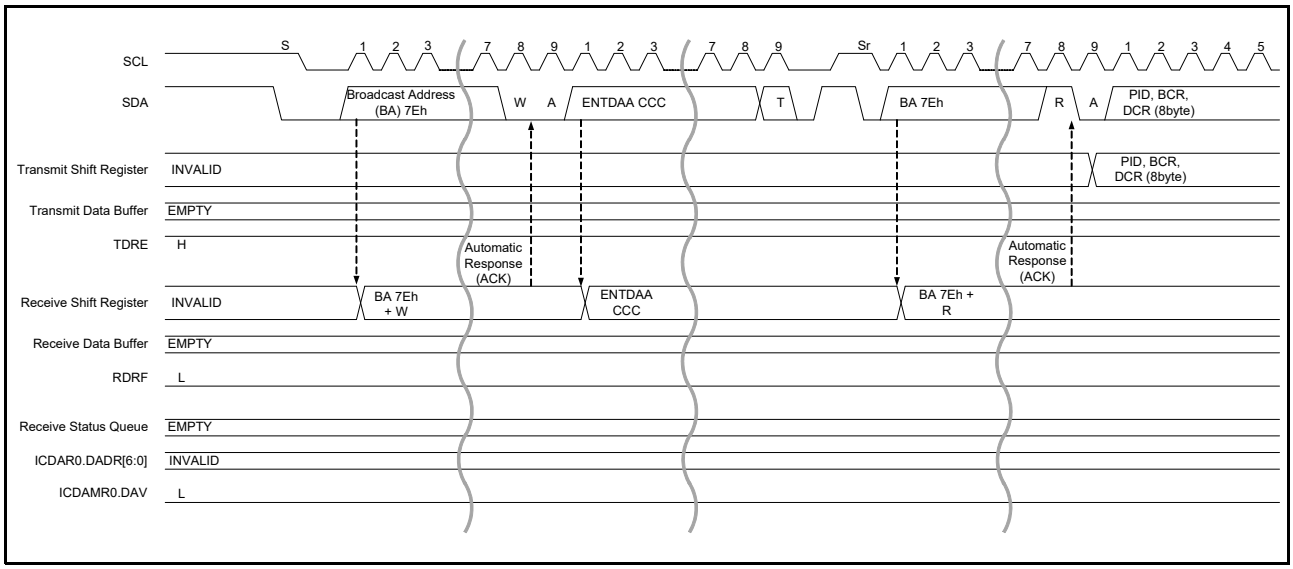


Figure 35.32 Dynamic Address Assignment Procedure (ENTDAA CCC) Timing (1/3)

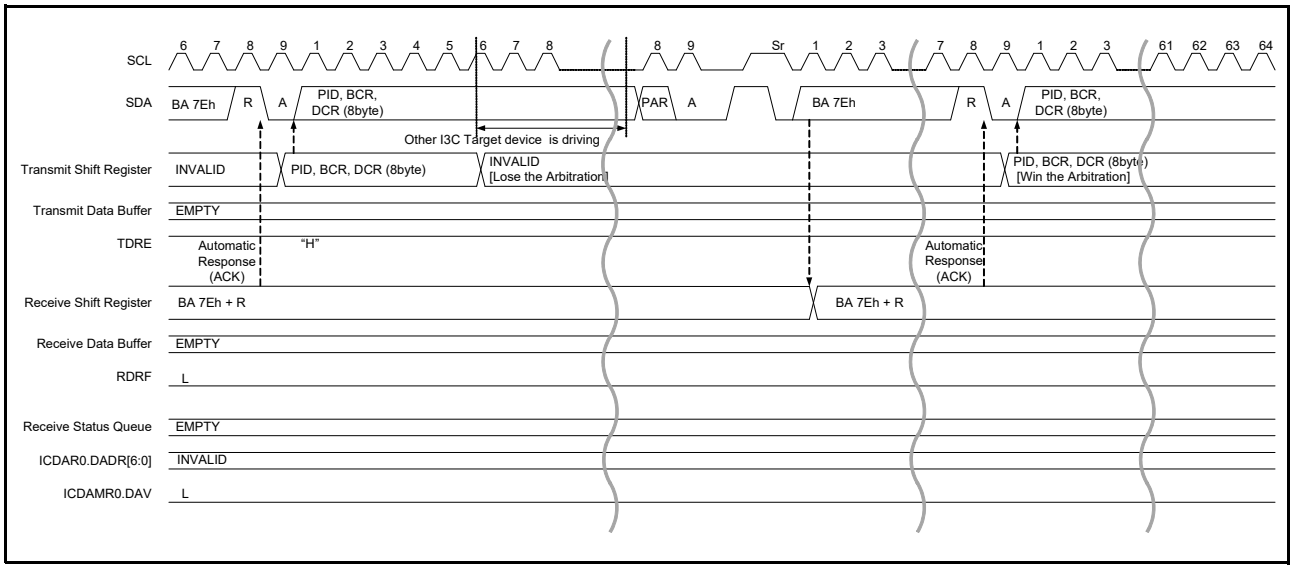


Figure 35.33 Dynamic Address Assignment Procedure (ENTDAA CCC) Timing (2/3)

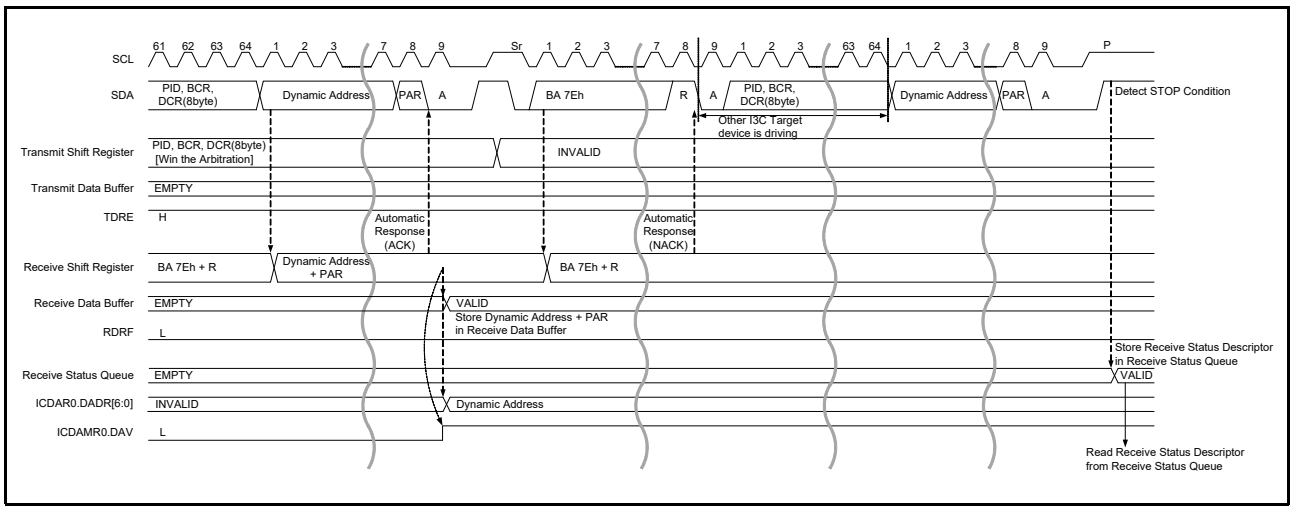


Figure 35.34 Dynamic Address Assignment Procedure (ENTDAA CCC) Timing (3/3)

When communicating with a Static Address until the Dynamic Address is assigned from the I3C Controller, by setting to the ICDAR0.SADR[6:0] bits, the ICDAMR0.SAV flag is set to 1 and the Static Address Will be effective.

If the I3C Target has a Static Address and the I3C Controller executes the Dynamic Address Assignment Procedure, it is possible to assign a Dynamic Address with SETDASA CCC.

The operation of RI3C during SETDASA CCC Dynamic Address Assignment Procedure is described below.

(1) Initial setting (For details, refer to section 35.4.3, Initial Settings)

(2) When SETDASA CCC which agrees with its own Static Address is received, the ICDAR0.DADR[6:0] bits are renewed and the ICDAMR0.DAV flag is set to 1. (For details, refer to In case of Direct Write CCC of 35.5.1 CCC detection function.)

(3) When SETDASA CCC is completed and a STOP condition is detected, Receive Status Descriptor is stored in receive status queue.

(4) Read Receive Status Descriptor via ICSQR register and check the status.

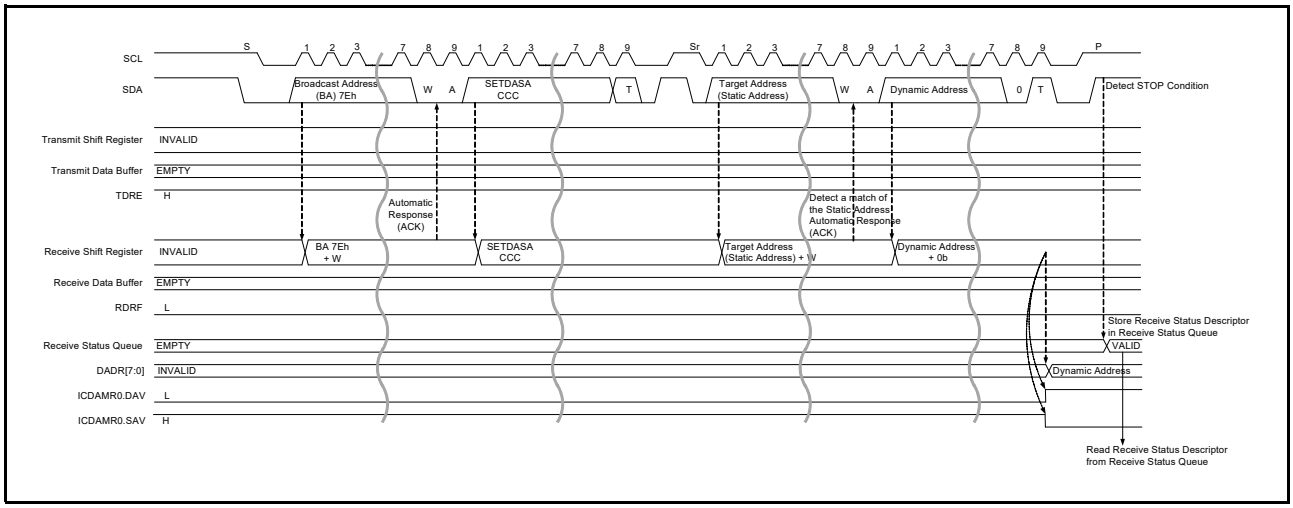


Figure 35.35 Dynamic Address Assignment Procedure (SETDASA CCC) Timing

35.4.6.2 SDR Write Transfer

(1) When Transaction is issued from the I3C Controller, it compares the Target Address in Address header with its own Target Address, and if it matches, RI3C responds with ACK.
When a Transaction is received, if the Receive Data Buffer is full, the I3C Target will respond with NACK in the Address header.
In preparation for retrying the I3C Controller, read the data from the Receive Data Buffer via the ICDR register, and empty the Receive Data Buffer.

(2) Data received from I3C Controller is stored in the Receive Data Buffer.

(3) In the receive data full interrupt, read the received data from the Receive Data Buffer via the ICDR register.

(4) When repeated START condition or STOP condition is detected, the Receive Status Descriptor is stored in the receive status queue.

(5) Read Receive Status Descriptor via ICSQR register and check the status.

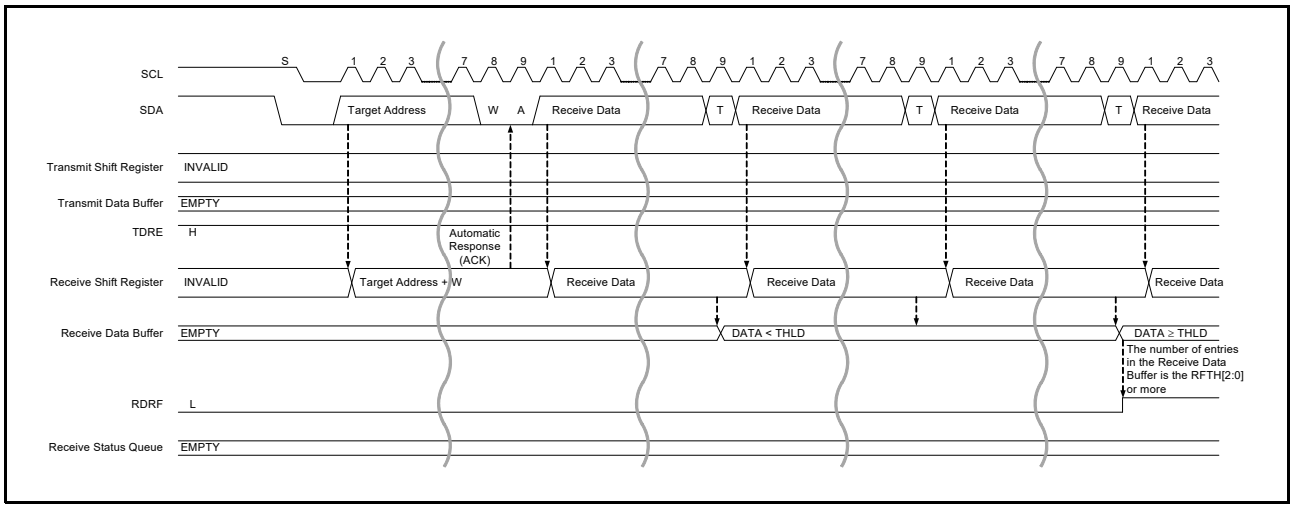


Figure 35.36 SDR Write Transfer Timing (1/2)

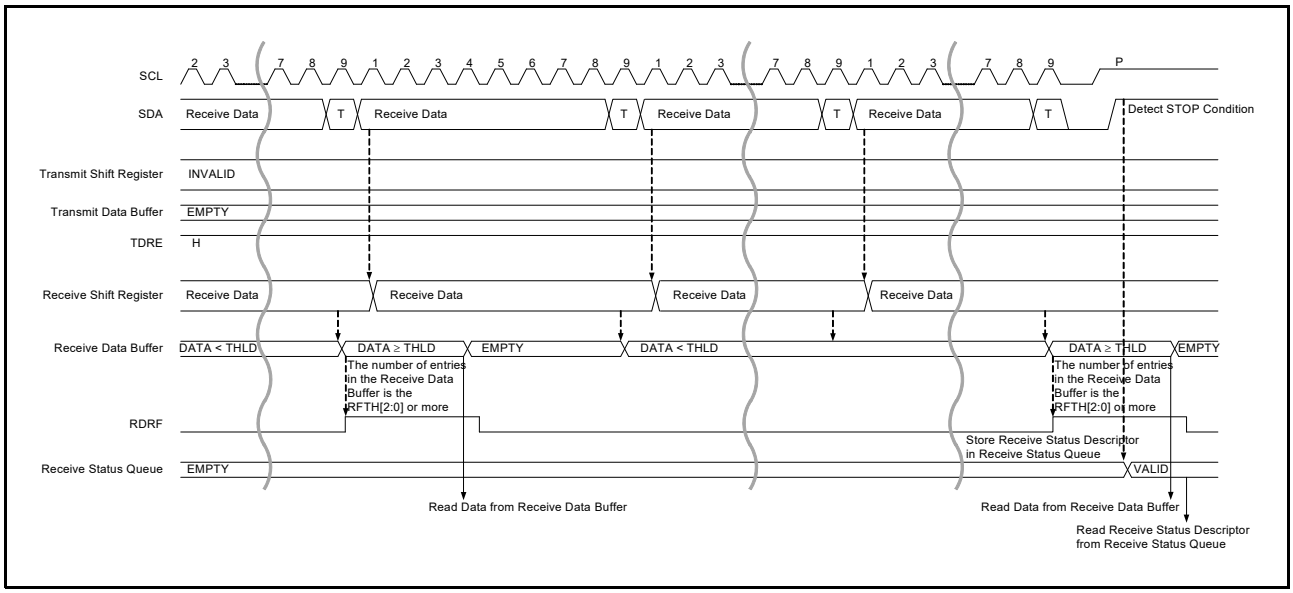


Figure 35.37 SDR Write Transfer Timing (2/2)

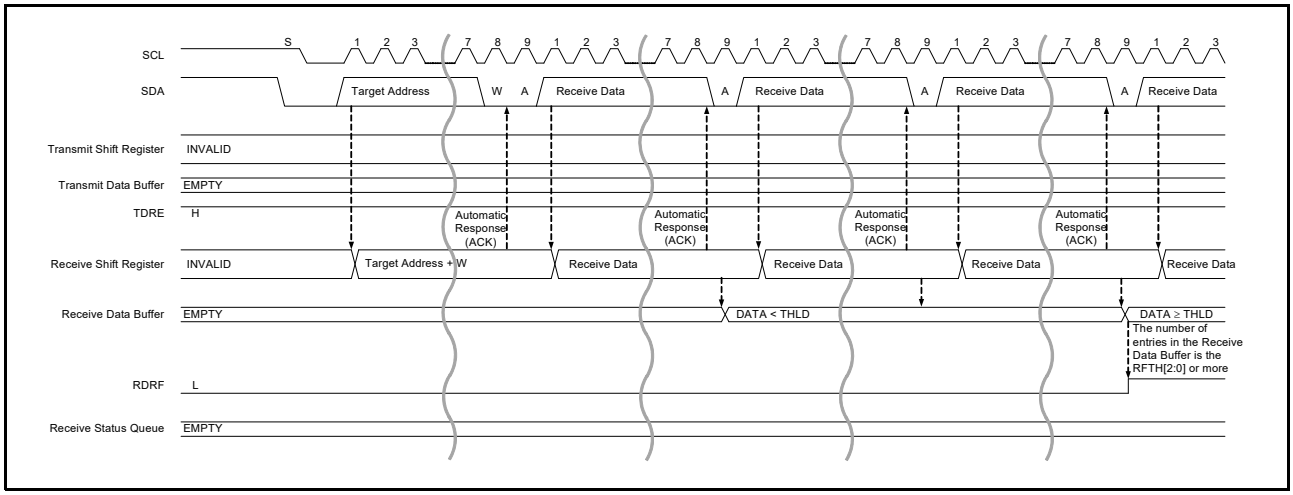


Figure 35.38 Legacy I²C Write Transfer Timing (1/2)

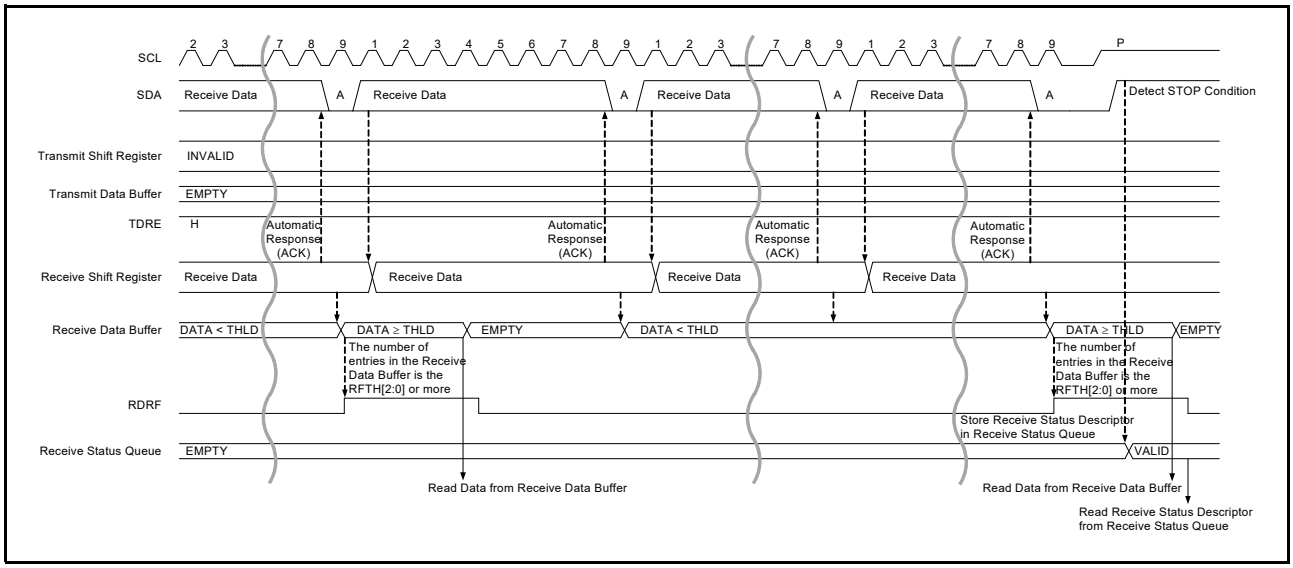


Figure 35.39 Legacy I²C Write Transfer Timing (2/2)

35.4.6.3 SDR Read Transfer

(1) Write the data requested from the I3C Controller to the Transmit Data Buffer via the ICDR register.

(2) When Transaction is issued from the I3C Controller, it compares the Target Address in Address header with its own Target Address, and if it matches, RI3C responds with ACK.
When a Transaction is received, if the Transmit Data Buffer is EMPTY, I3C Target responds with NACK with the Address header.
In preparation for retrying the I3C Controller, write data to the Transmit Data Buffer via the ICDR register.

(3) Transmit the data stored in the Transmit Data Buffer.

(4) If data to be transmitted still remains, write the data to be transmitted with a transmit data empty interrupt to the Transmit Data Buffer via the ICDR register.

(5) SDR:
When the transmission of the data stored in the Transmit Data Buffer is completed, Low is output to the T-bit following Data, and it is notified to the I3C Controller that it is the final data.
Legacy I^2C Message:
When NACK is detected, data transmission is terminated.

(6) When a repeated START condition or STOP condition is detected, the Receive Status Descriptor is stored in the receive status queue.

(7) Read the Receive Status Descriptor via ICSQR and check the status.
MCU Ver.1 has the following restriction and workaround. This restriction and workaround are not required for MCU Ver.2.
If the data length does not match, set the ICRCR.ISRST bit to 1 and then reset the internal states of this module.
For details, refer to 35.5.5.6 Error Recovery Operation.

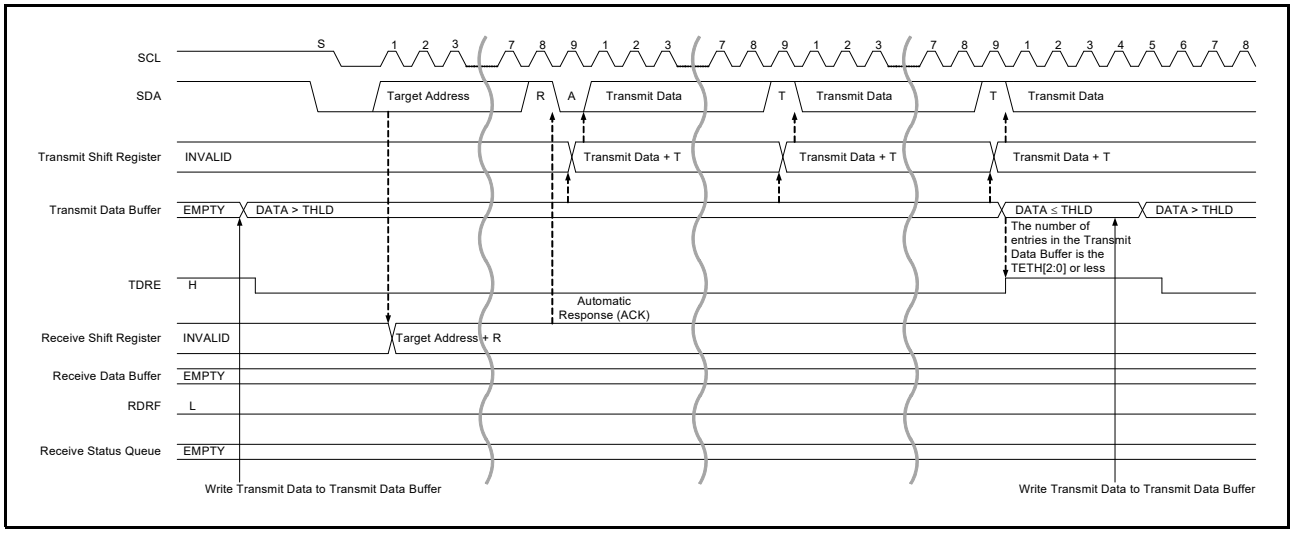


Figure 35.40 SDR Read Transfer Timing (1/2)

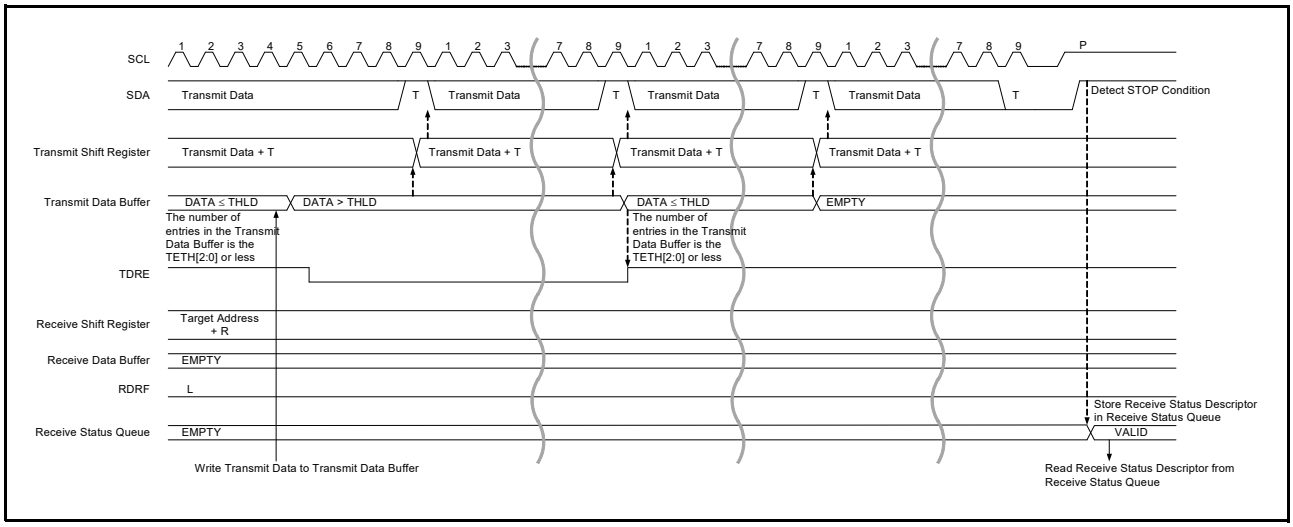


Figure 35.41 SDR Read Transfer Timing (2/2)

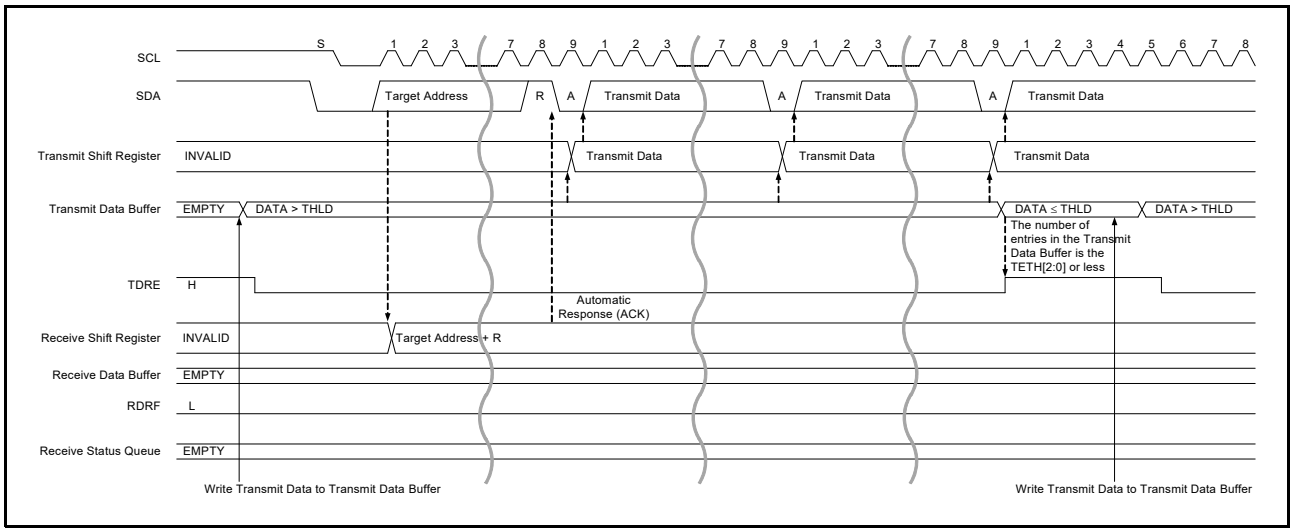


Figure 35.42 Legacy I²C Read Transfer Timing (1/2)

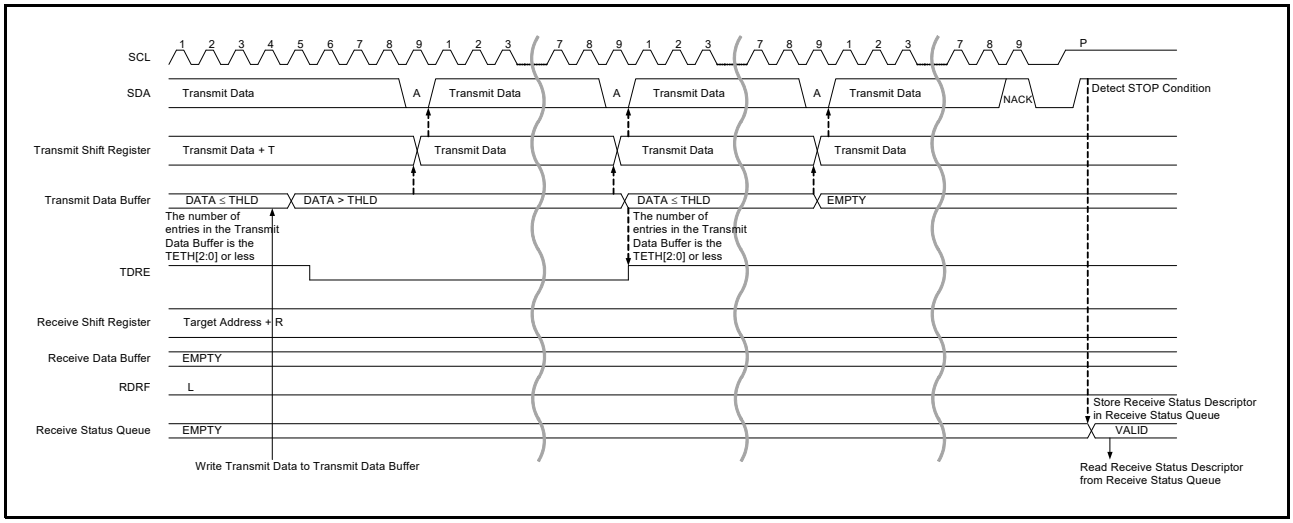


Figure 35.43 Legacy I²C Read Transfer Timing (2/2)

35.4.6.4 IBI Transfer

(1) When sending Target Interrupt Request.
When transmitting IBI Data, write IBI Data to the IBI Data Buffer via the ICIQR register.

(2) Write Command Descriptor (immediate data transfer command or regular data transfer command) to the command queue for IBI Transfer via the ICCQR register.

(3) When Command Descriptor is written to command queue, IBI Transaction is issued under the following conditions.

- Detect a START condition. (Does not apply a repeated START condition)
- If no START condition is forthcoming within the following bus condition, then this module issue a START Request by pulling the SDA line Low.
- (a) Target Interrupt Request, Controller Role Request: Bus Available
  (b) Hot-Join Event: Bus Idle

(4) In Target Address and RnW field in the Address header, if losing Arbitration by issuing a Transaction from I3C Controller, stop issuing Transaction.
When detecting repeated START condition or STOP condition, store the Response Descriptor into the response queue.

(5) When sending Target Interrupt Request:

- When IBI data for transmission still remain, write IBI data with an IBI queue empty/full interrupt to the IBI queue via the ICIQR register.
- When the transmission of IBI Data for the number of data length specified by the DATA_LENGTH filed of the Command Descriptor is completed, output Low to the T-bit following IBI Data and notify the I3C Controller that it is the final IBI Data.

(6) When detecting repeated START condition or STOP condition, store the Response Descriptor into the response queue.

(7) Read the Response Descriptor form the response queue with the ICRQR register and check the status. If NACK is responded, repeat steps (1) to (7).

(8) When sending Target Interrupt Request:
Check that the value of the DATA_LENGTH field of the Response Descriptor is 0.

The CRR processing flow is shown in Figure 35.46. When joining the I3C bus by Hot-Join after the I3C bus has already been configured, issue the Hot-Join according to the flow shown in Figure 35.51.

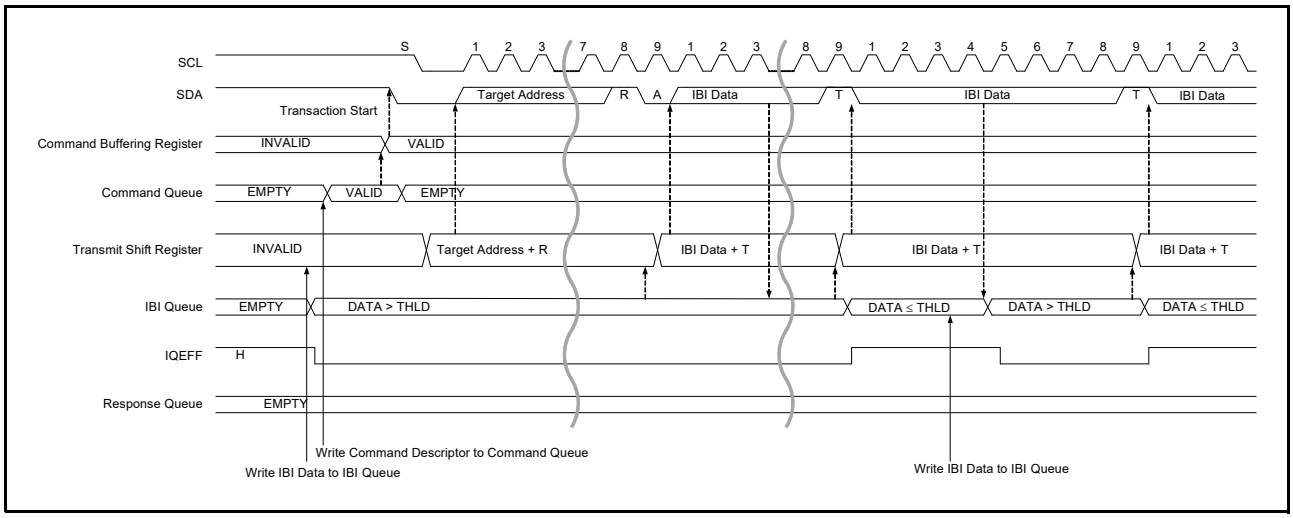


Figure 35.44 I3C Target IBI Transfer Timing (1/2)

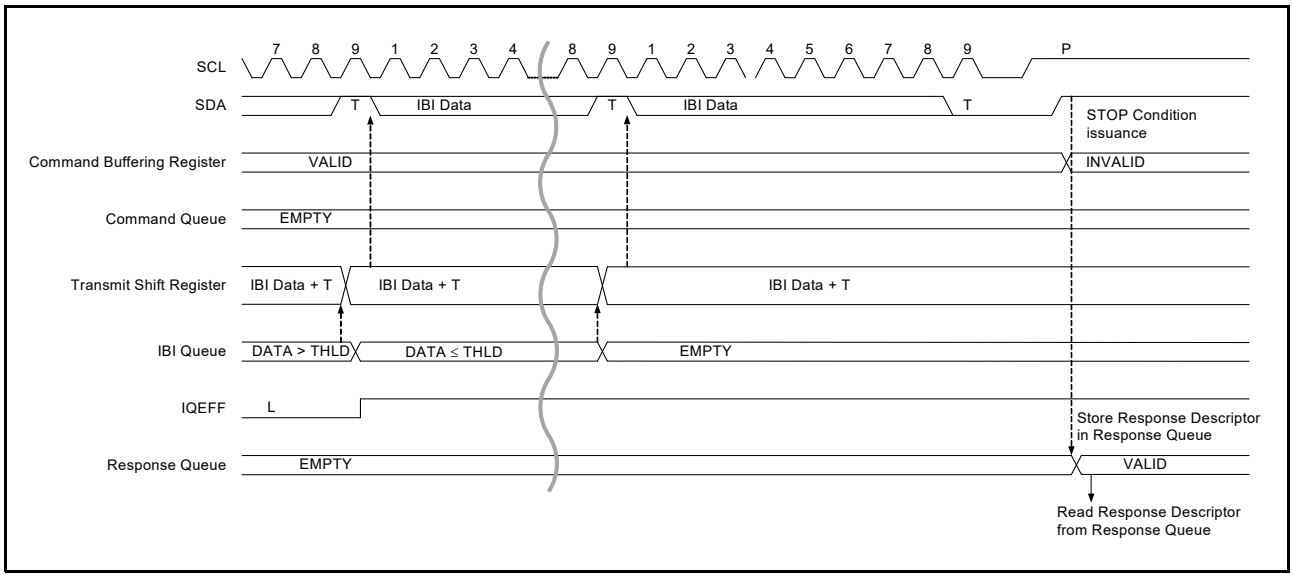


Figure 35.45 I3C Target IBI Transfer Timing (2/2)

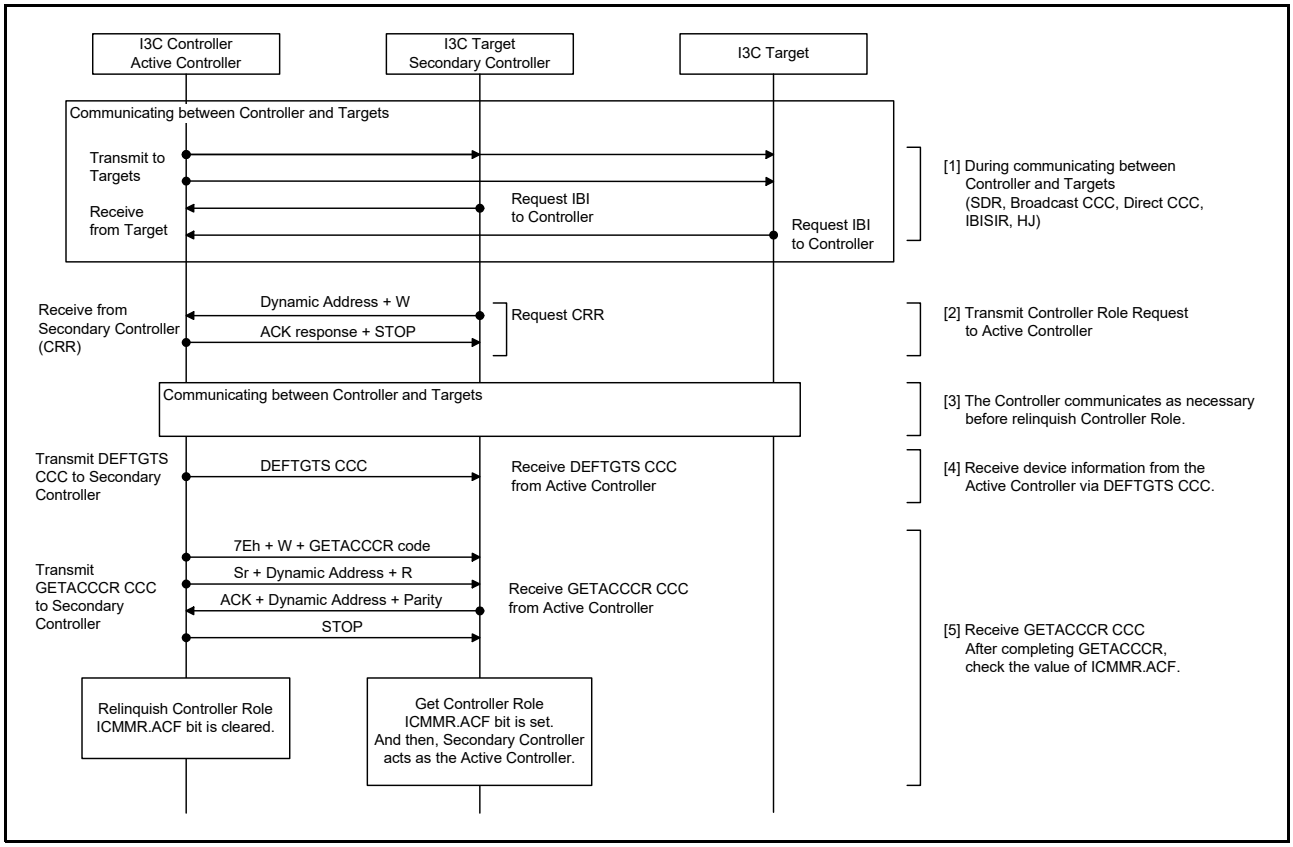
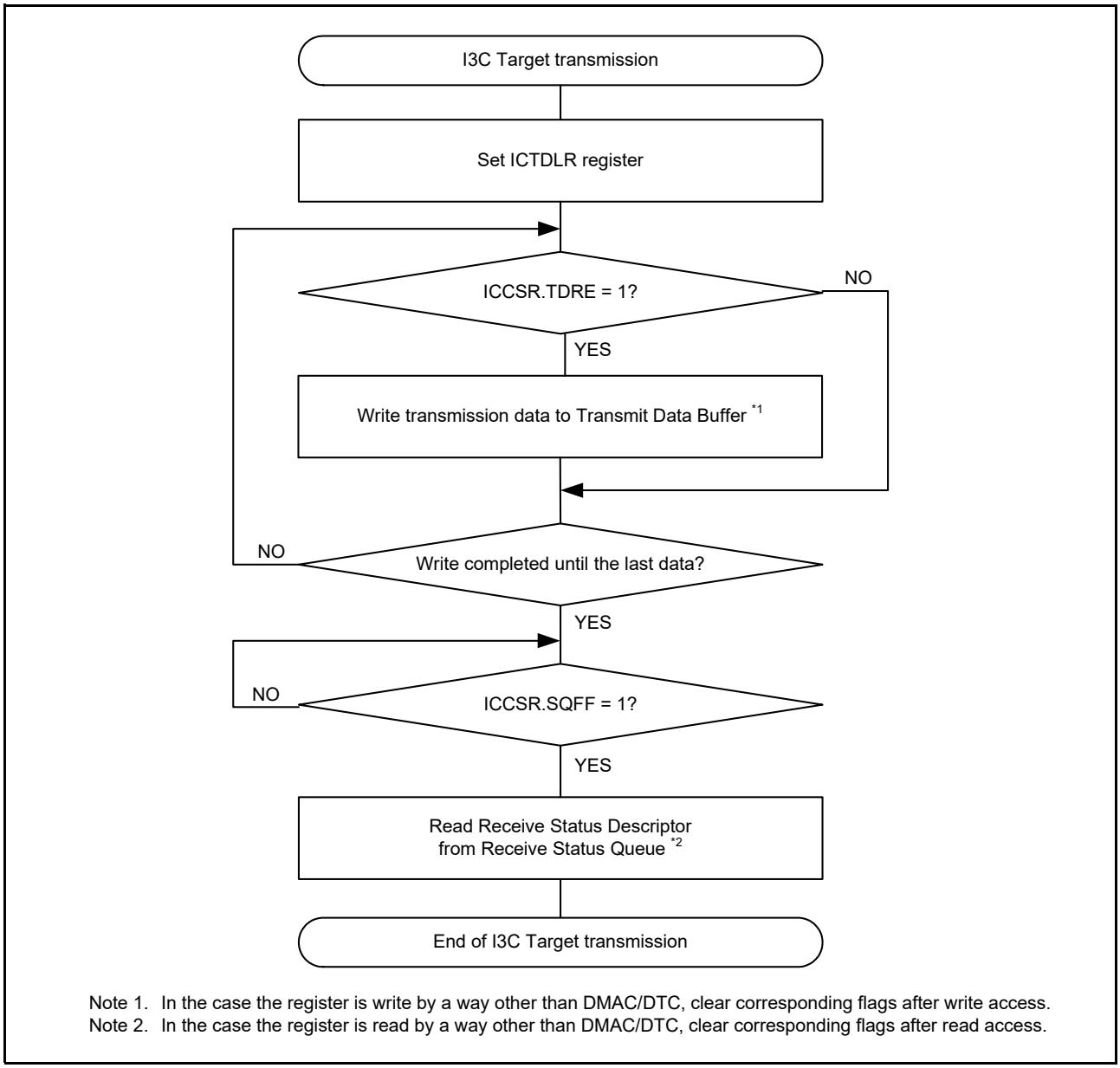


Figure 35.46 I3C Target CRR Processing Flow

35.4.6.5 I3C Target Transmission Flow (FIFO Buffer Transfer)

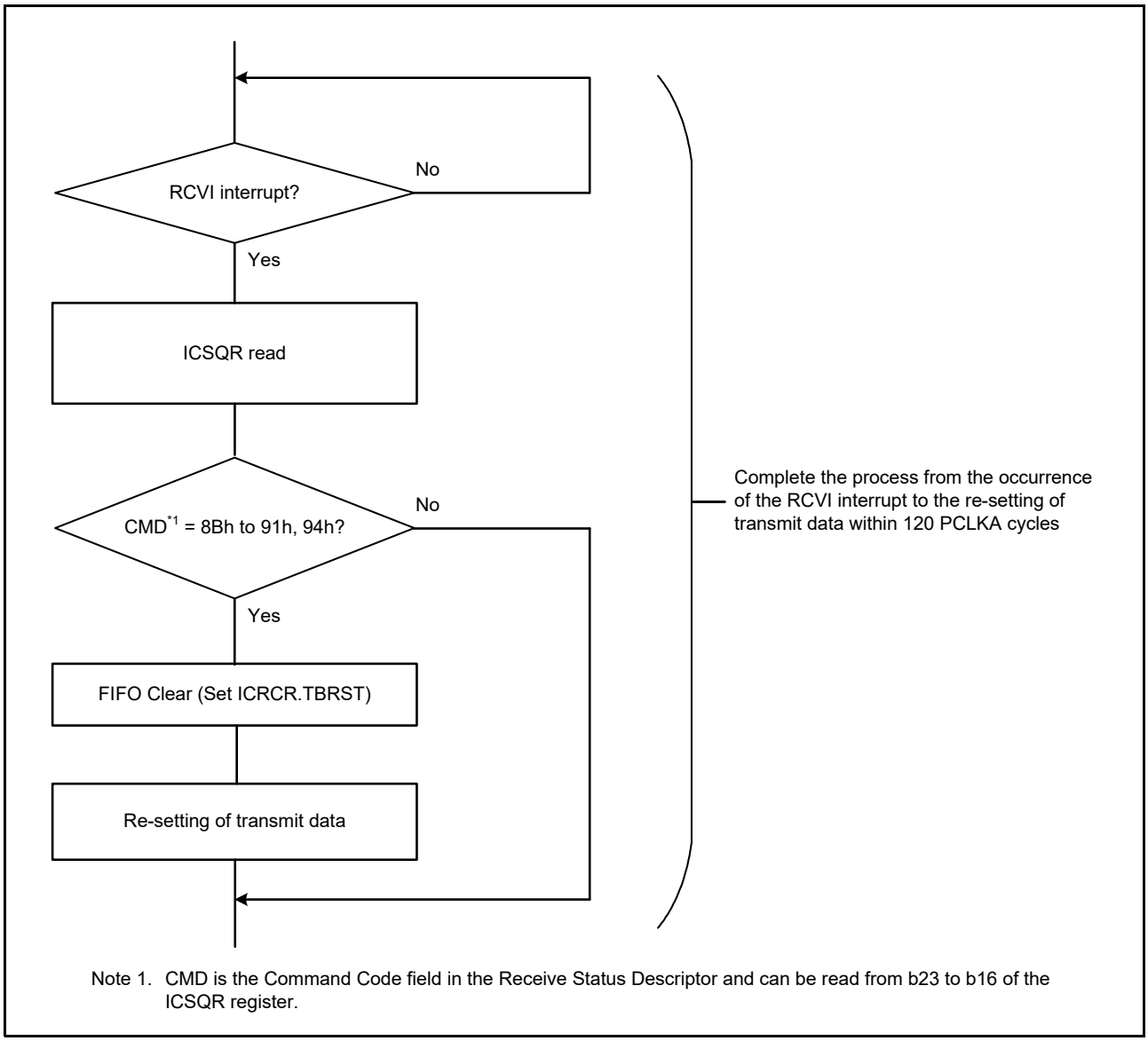
Target Transmission Flow in FIFO buffer transfer is common to Legacy I^2C, SDR (Private Transfer, Broadcast CCC, Direct CCC).



Note 1. In the case the register is write by a way other than DMAC/DTC, clear corresponding flags after write access.
Note 2. In the case the register is read by a way other than DMAC/DTC, clear corresponding flags after read access.

Figure 35.47 Example of I3C Target Transmission Flowchart (FIFO Buffer Transfer)

When using the RI3C as an I3C Target, if I3C Target receives GET CCC while data exists by writing from the ICDR register to the transmit buffer, follow the flow below.

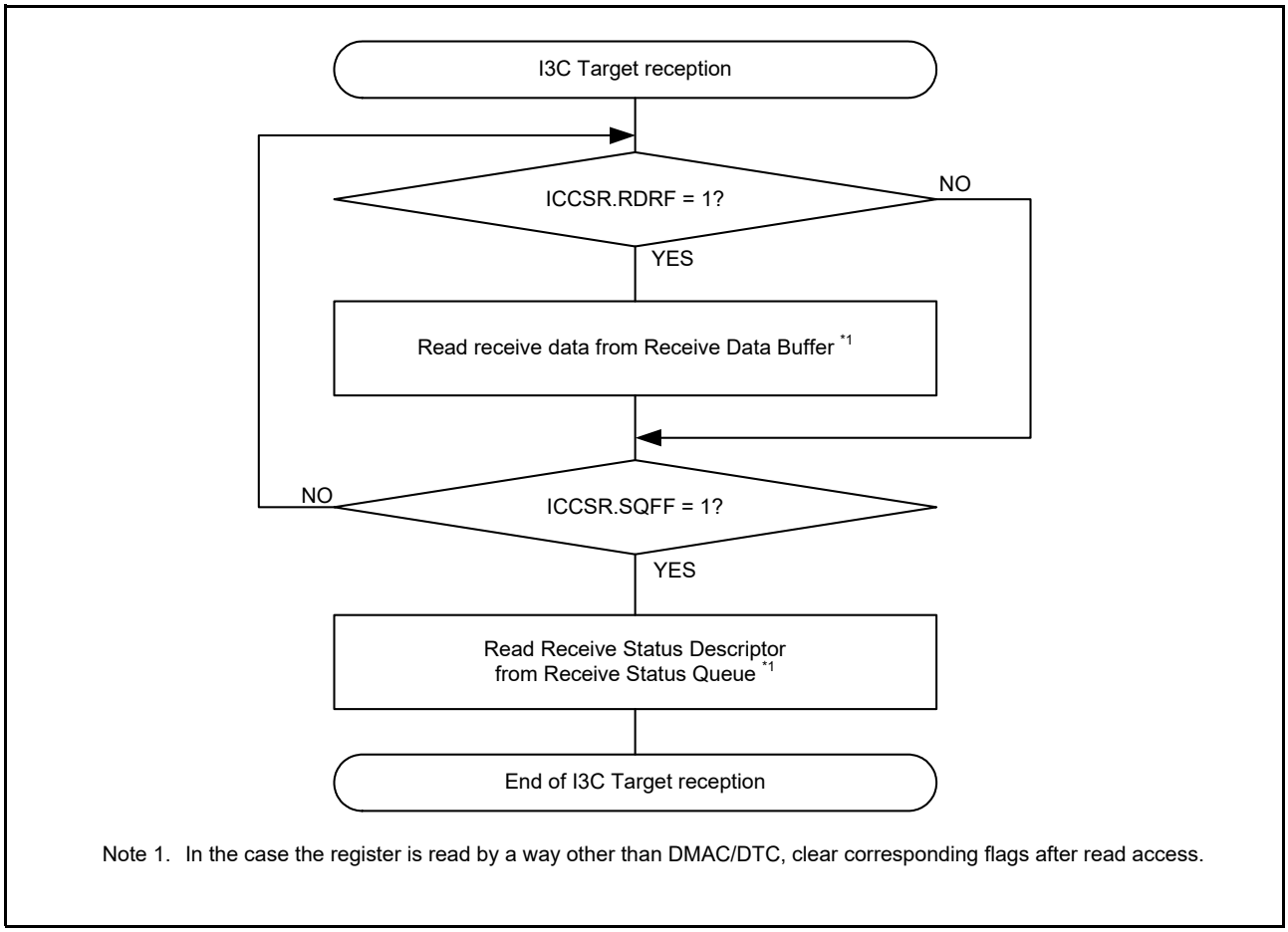


Note 1. CMD is the Command Code field in the Receive Status Descriptor and can be read from b23 to b16 of the ICSQR register.

Figure 35.48 I3C Target Receives GET CCC Command while Data Exists by Writing from the ICDR Register to the Transmit Buffer

35.4.6.6 I3C Target Reception Flow (FIFO Buffer Transfer)

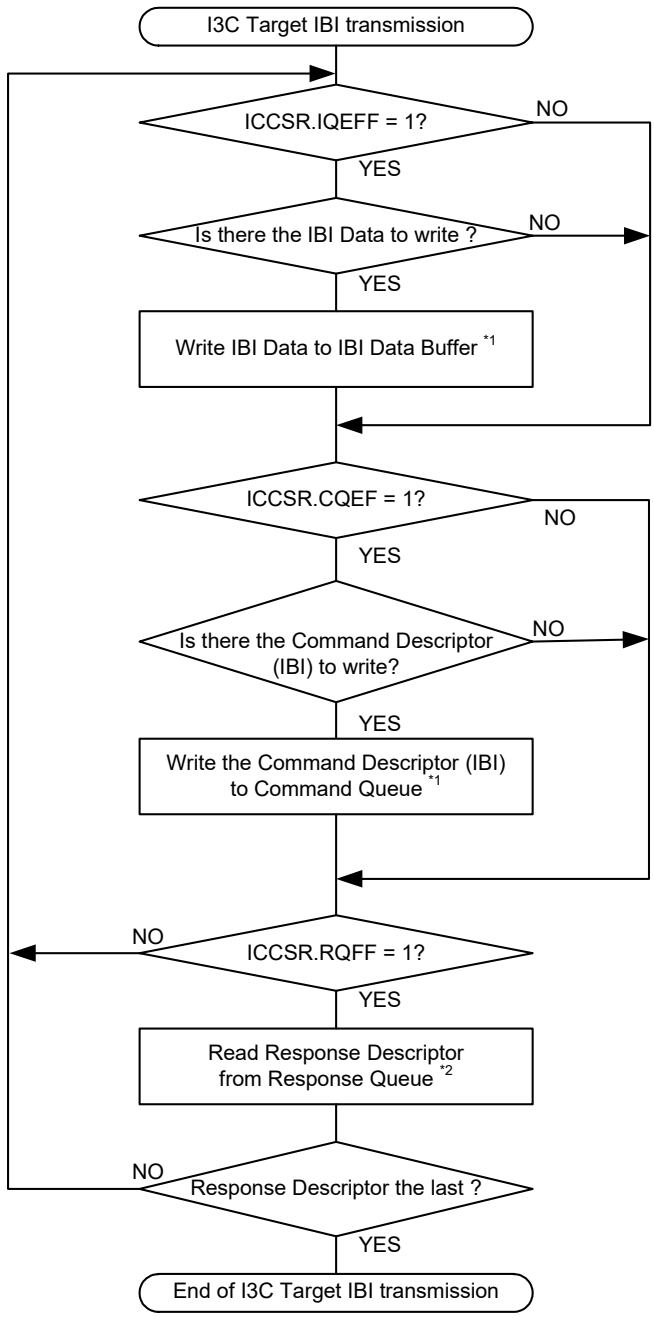
Target Reception Flow in FIFO buffer transfer is common to Legacy I^2C, SDR (Private Transfer, Broadcast CCC, Direct CCC).



Note 1. In the case the register is read by a way other than DMAC/DTC, clear corresponding flags after read access.

Figure 35.49 Example of I3C Target Reception Flowchart (FIFO Buffer Transfer)

35.4.6.7 I3C Target IBI Transmission Flow



Note 1. In the case the register is write by a way other than DMAC/DTC, clear corresponding flags after write access.
Note 2. In the case the register is read by a way other than DMAC/DTC, clear corresponding flags after read access.

Figure 35.50 Example of I3C Target IBI Transmission Flowchart

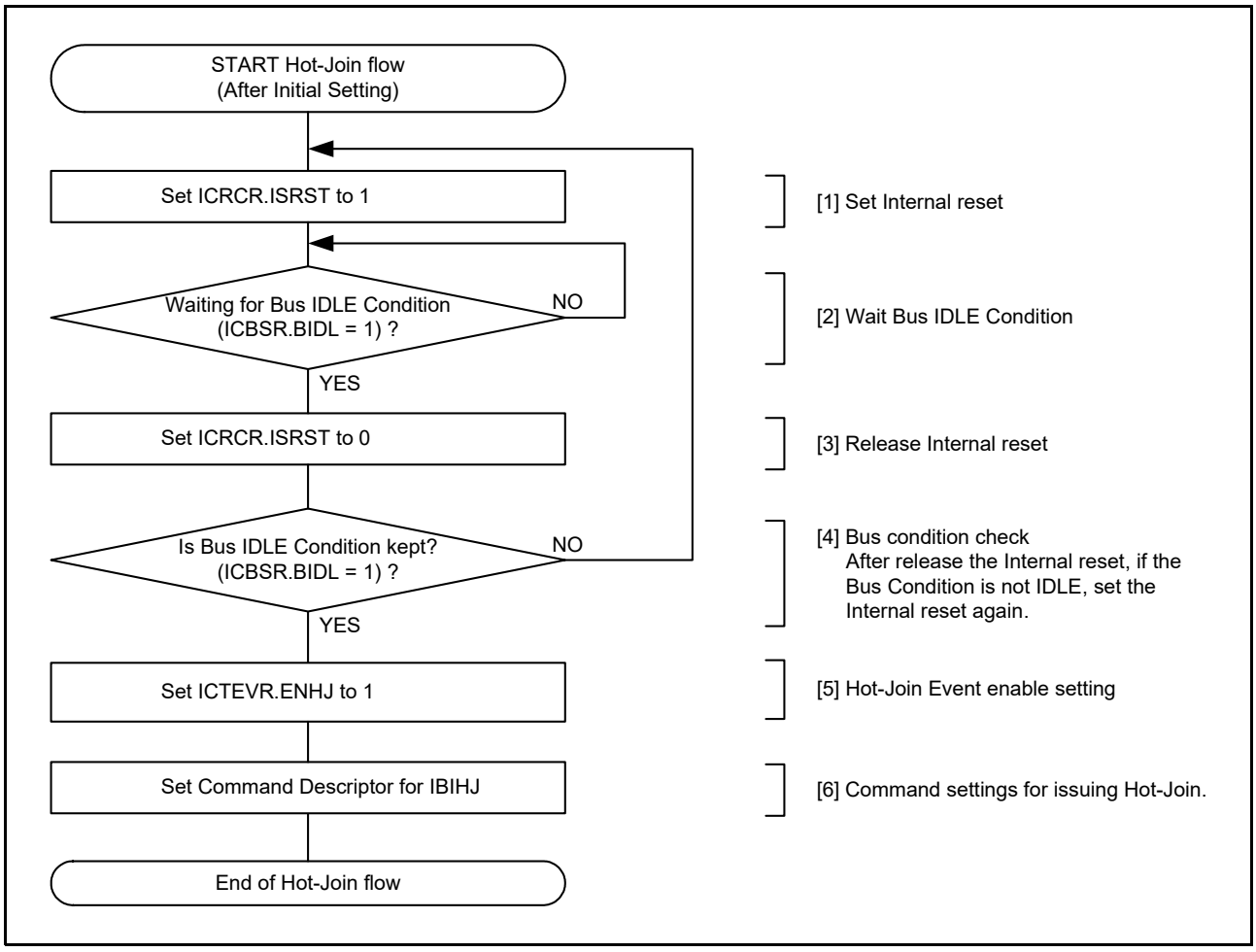


Figure 35.51 Hot-Join Flowchart after the I3C Bud has Already been Configured

35.5 Details of Function

35.5.1 CCC detection function

In case of Broadcast CCC

(1) Receive Broadcast Address (7Eh) and RnW = 0 after START condition or repeated START condition.
(2) Respond with ACK.
(3) Receive Common Command Code (CCC).
(4) In accordance with the CCC, the following data is stored. (Storage destination: refer to Table 35.6)
(5) Store the Receive Status Descriptor into the receive status queue.

In case of Broadcast CCC (ENTDAA)

(1) Receive Broadcast Address (7Eh) and RnW = 0 after START condition.
(2) Respond with ACK.
(3) Receive ENTDAA.
(4) If receives Broadcast Address (7Eh) and RnW = 1 after repeated START condition.
(5) When the Dynamic Address is not assigned, ACK response is done.
(6) This Provisioned ID (the ICPIDHR register and bits b15 to b0 of the ICPIDLR register), BCR (bits b15 to b8 of the ICDCTR register), DCR (bits b7 to b0 of the ICDCTR register) are transmitted.
(7) When winning the arbitration in a transmission of the above Step 6, the Dynamic Address following that is received.
When losing arbitration in a transmission of the above Step 6, processing of Step 6 is repeated from Step 4.
(8) When parity of the Dynamic Address is valid, ACK response is done.
(9) When parity of the Dynamic Address is invalid, NACK replies, and repeat the process from Steps 4 to 7.
(10) ICDAR0.DADR[6:0] bits are renewed and the ICDAMR0.DAV flag is set to 1.
(11) Upon detecting the STOP condition, Store the Receive Status Descriptor into the receive status queue.

In case of Direct Write CCC

(1) Receive Broadcast Address (7Eh) and RnW = 0 after START condition or repeated START condition.
(2) Respond with ACK.
(3) Receive Common Command Code (CCC).
(4) Receive Dynamic Address and RnW = 0 after repeated START condition.
(5) Compare the received Dynamic Address with the assigned Dynamic Address, and if it matches, RI3C responds with ACK.
If they do not match, it responds with NACK and waits for repeated START condition or STOP condition.
(6) In accordance with the CCC, the following data is stored. (Storage destination: refer to Table 35.6)
(7) Store the Receive Status Descriptor into the receive status queue.

In case of Direct Read CCC

(1) It receives Broadcast Address (7Eh) and RnW = 1 after START condition or repeated START condition.
(2) Respond with ACK.
(3) Receive Common Command Code (CCC).
(4) Receive Dynamic Address and RnW = 1 after repeated START condition.
(5) Compare the received Dynamic Address with the assigned Dynamic Address, and if it matches, RI3C responds with ACK.
If they do not match, it responds with NACK and waits for repeated START condition or STOP condition.
(6) Respond from register according to CCC. (Responding CCC: refer to Table 35.6)
(7) Store the Receive Status Descriptor into the receive status queue.

Table 35.6 Common Command Code Operation

Command Code	CCC Type	Command Name	With Data	Auto Response	Storage
00h	Broadcast	ENEC	Yes	—	Register
01h	Broadcast	DISEC	Yes	—	Register
02h	Broadcast	ENTAS0	No	—	Register
03h	Broadcast	ENTAS1	No	—	Register
04h	Broadcast	ENTAS2	No	—	Register
05h	Broadcast	ENTAS3	No	—	Register
06h	Broadcast	RSTDAA	No	—	Register
07h	Broadcast	ENTDAA	Yes	Yes	Register
08h	Broadcast	DEFTGTS	Yes	—	FIFO
09h	Broadcast	SETMWL	Yes	—	Register
0Ah	Broadcast	SETMRL	Yes	—	Register
0Bh	Broadcast	ENTTM	Yes	—	Register
29h	Broadcast	SETAASA	No	—	Register
80h	Direct Write	ENEC	Yes	—	Register
81h	Direct Write	DISEC	Yes	—	Register
82h	Direct Write	ENTAS0	No	—	Register
83h	Direct Write	ENTAS1	No	—	Register
84h	Direct Write	ENTAS2	No	—	Register
85h	Direct Write	ENTAS3	No	—	Register
86h	Direct Write	RSTDAA	No	—	Register
87h	Direct Write	SETDASA	Yes	—	Register
88h	Direct Write	SETNEWDA	Yes	—	Register
89h	Direct Write	SETMWL	Yes	—	Register
8Ah	Direct Write	SETMRL	Yes	—	Register
8Bh	Direct Read	GETMWL	—	Yes	Register
8Ch	Direct Read	GETMRL	—	Yes	Register
8Dh	Direct Read	GETPID	—	Yes	Register
8Eh	Direct Read	GETBCR	—	Yes	Register
8Fh	Direct Read	GETDCR	—	Yes	Register
90h	Direct Read	GETSTATUS	—	Yes	Register
91h	Direct Read	GETACCCR	—	Yes	Register
94h	Direct Read	GETMXDS	—	Yes	Register

35.5.2 Clock Stalling

RI3C has the function of stalling the SCL during the SCL Low period.
The SCL stall control is described in the table below.

Table 35.7 I3C Clock Stalling

Clock Stalling Condition	Clock Stalling Control	Clock Stalling Period
I3C Transfer, ACK/NACK Phase	ICSTCR.APSE bit setting	During the count period of ICSTCR.STT[15:0] value
	Transmit FIFO Empty	Until data is written to the transmit FIFO
	Receive FIFO Full	Until data is read from the receive FIFO
I3C Write Data Transfer, Parity Bit	ICSTCR.PBSE bit setting	During the count period of ICSTCR.STT[15:0] value
	Transmit FIFO Empty	Until data is written to the transmit FIFO
I3C Read Transfer, Transition Bit	Receive FIFO Full	Until data is read from the receive FIFO
Assigned Address Phase	ICSTCR.AASE bit setting	During the count period of ICSTCR.STT[15:0] value

The following figure shows the stalling timing of each condition.

(1) I3C Transfer, ACK/NACK Phase

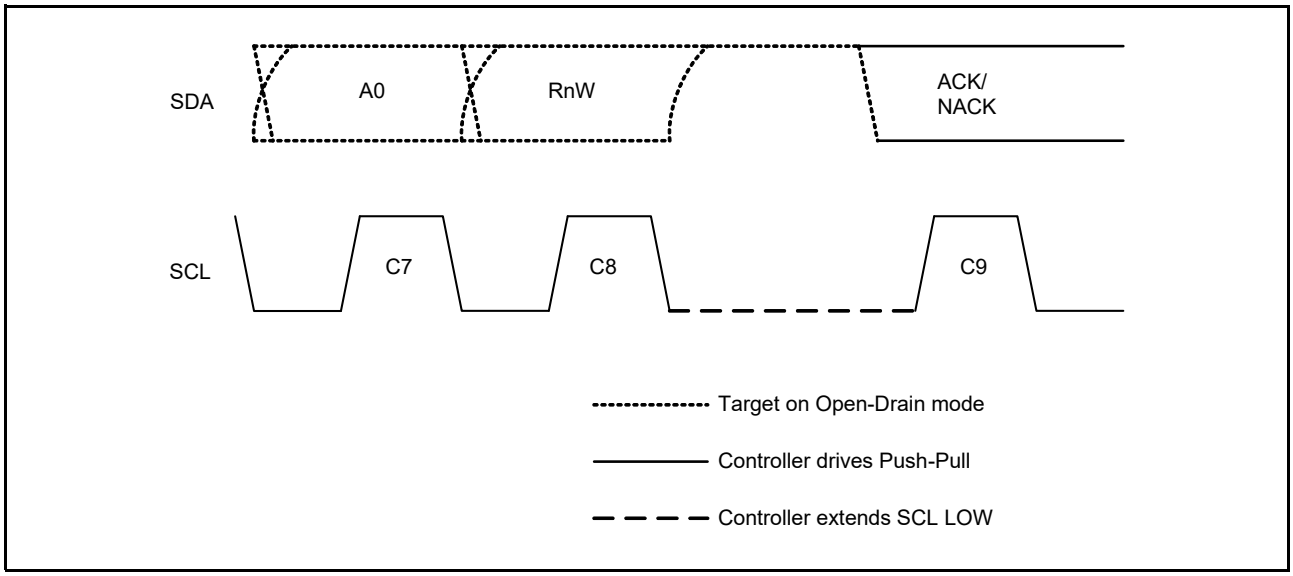


Figure 35.52 Controller Clock Stalling in ACK Phase

(2) I3C Write Data Transfer, Parity Bit

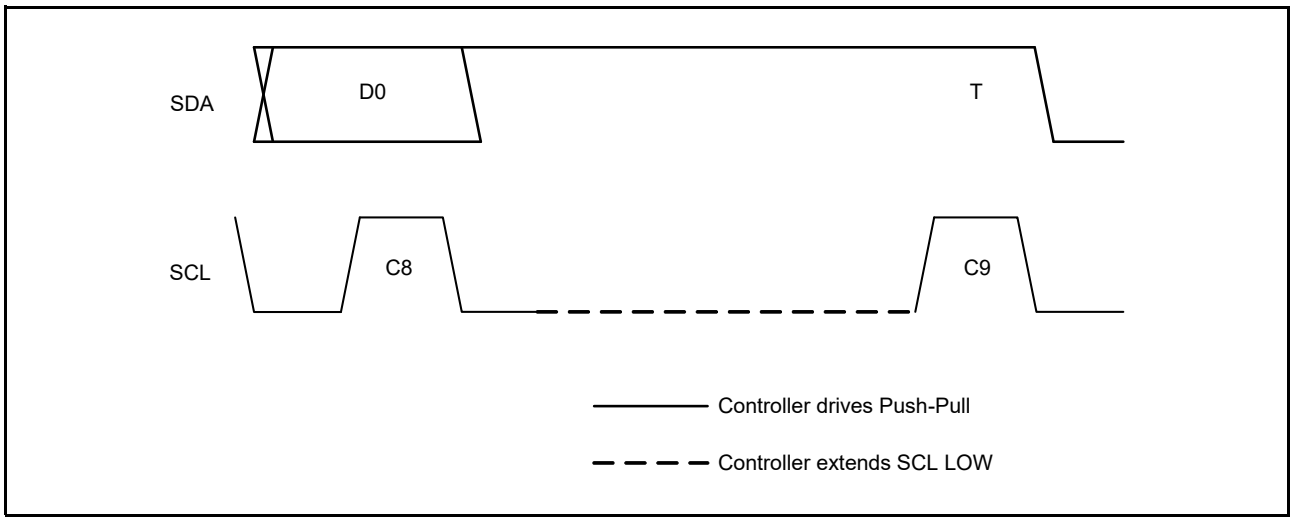


Figure 35.53 Controller Clock Stalling in Write Parity Bit

(3) I3C Read Transfer, Transition Bit

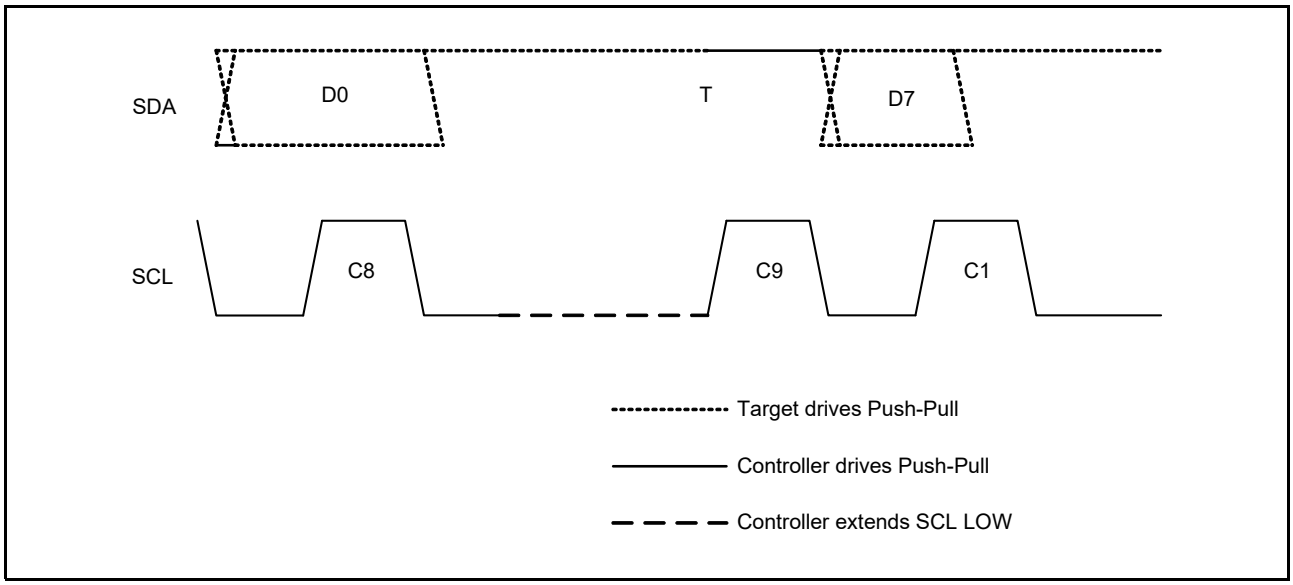


Figure 35.54 Controller Clock Stalling in T-bit Before Next Read Data

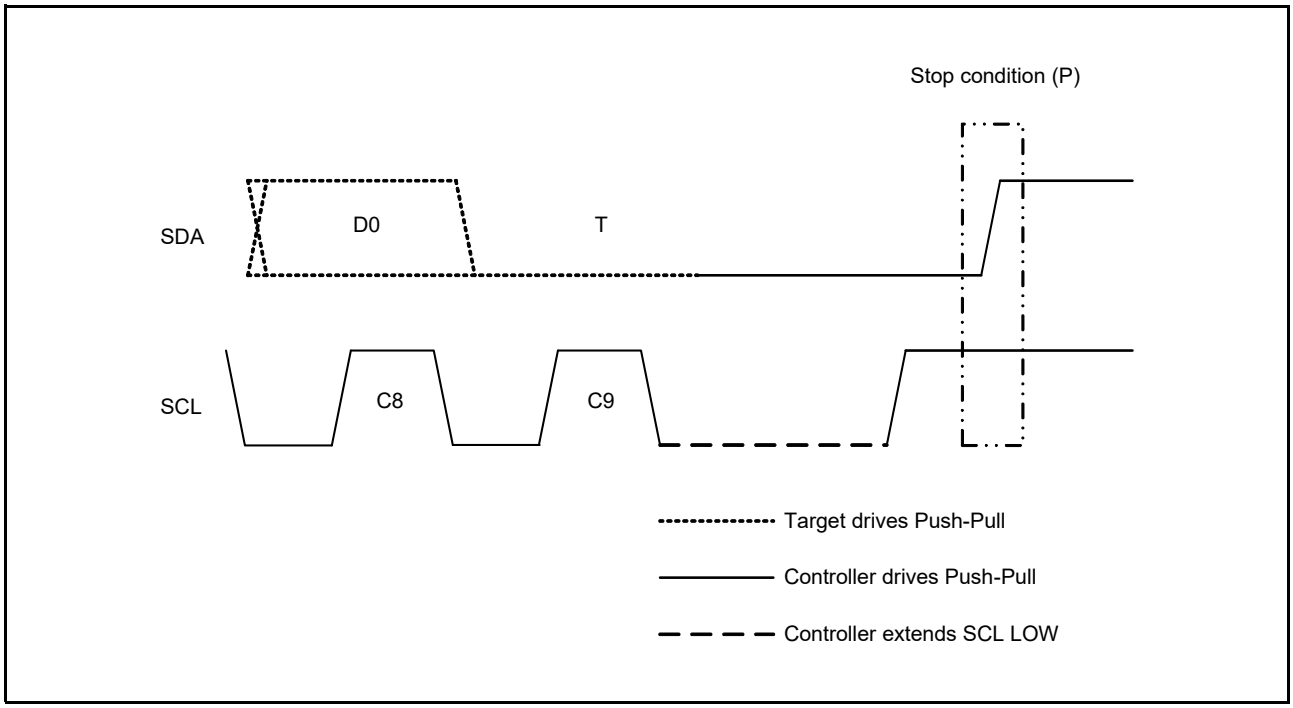


Figure 35.55 Controller Clock Stalling in T-bit Before STOP Condition

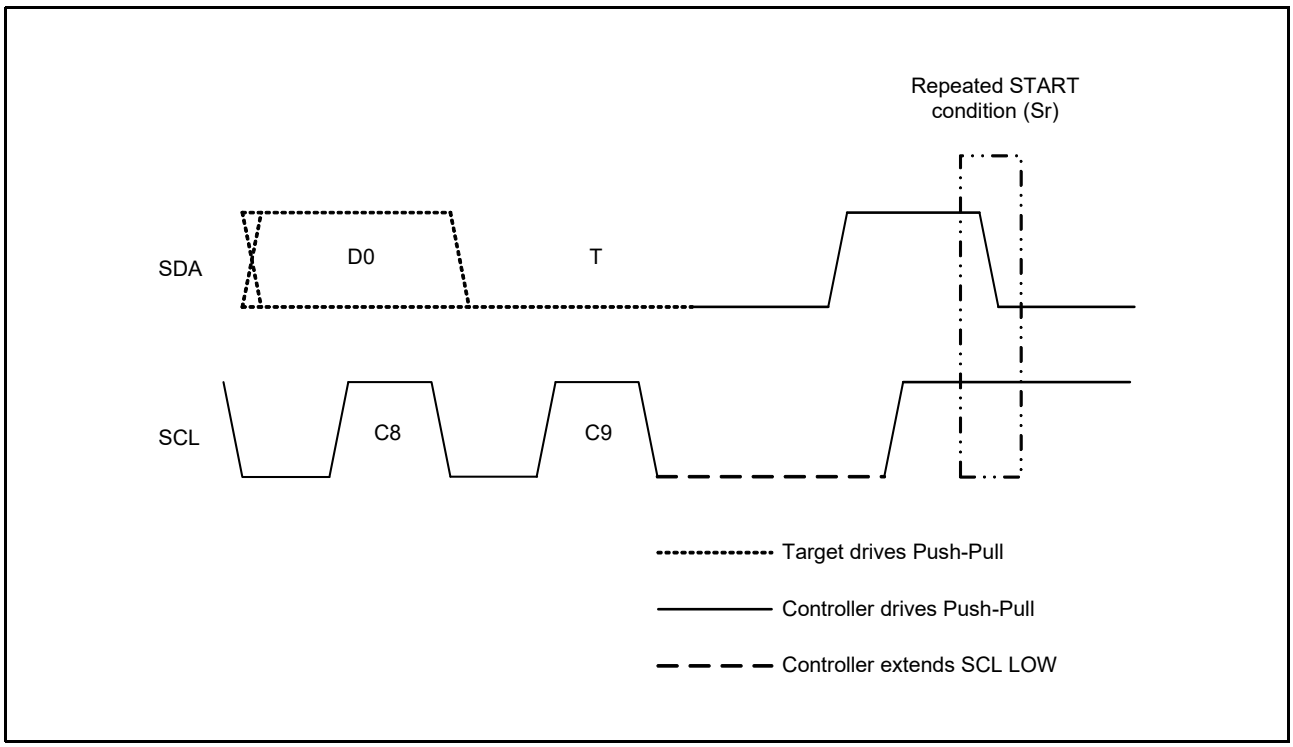


Figure 35.56 Controller Clock Stalling in Low T-bit before Repeated START Condition

(4) Dynamic Address Assignment, First Bit of Assigned Address

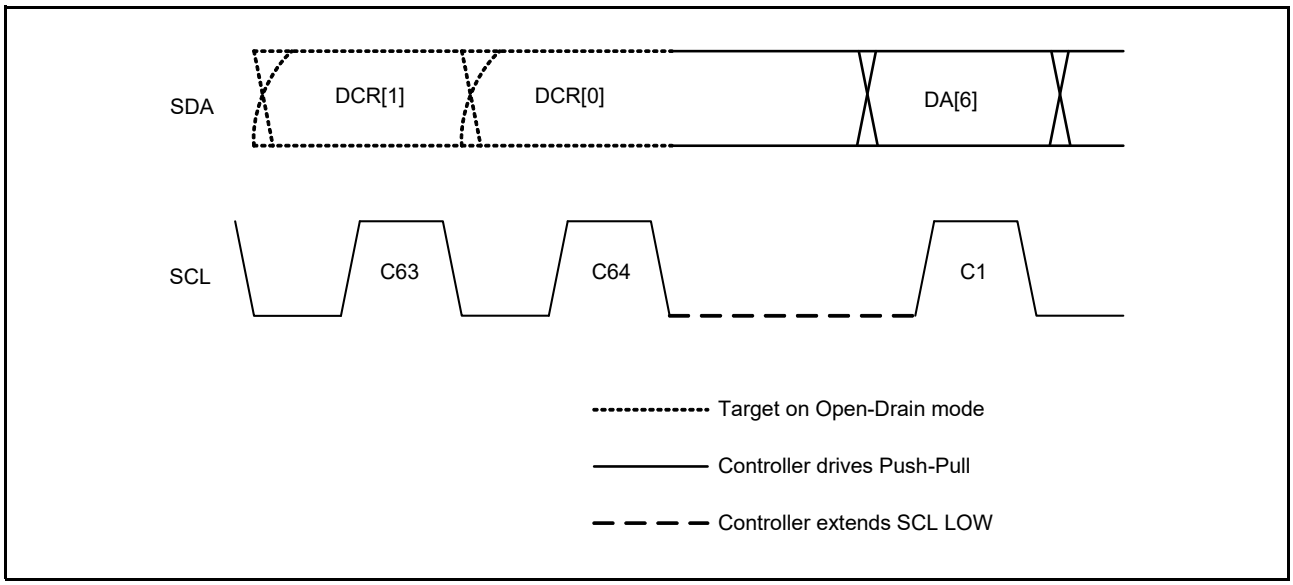


Figure 35.57 Controller Clock Stalling in Dynamic Address First Bit

35.5.3 IBI

RI3C detects IBI in the arbitrated Address header following a START condition (but not following a repeated START condition). If START Request (SDA Low Drive) is issued from Target Device, RI3C drives SCL low and completes START condition. After that, it supplies SCL and receives IBI Request.
The IBI to be detected is classified into the following three types.

- Target Interrupt Request
- Controller Role Request
- Hot-Join Event

The operation when detecting each IBI is described below.

35.5.3.1 Target Interrupt Request

(1) Detect Target Address with RnW bit High in Address header.
(2) Compare the detected Target Address with the DADR[7:0] bits in each DAT (the ICTDATRm register).
(3) When it does not match the ICTDATRm.DADR[7:0] bits:
• Responds with NACK, then issues a STOP condition.
When it matches the ICTDATRm.DADR[7:0] bits and the ICTDATRm.TIRRJ bit = 1:
• RI3C operates in the following order:
1. Responds with NACK.
2. Issues a repeated START condition, then automatically issues Direct DISEC CCC to the detected Target.
3. Issues a STOP condition.
When it matches the ICTDATRm.DADR[7:0] bits and the ICTDATRm.TIRRJ bit = 0:
• Responds with ACK.
(4) When the ICTDATRm.IBIPL bit is 0:
• Issues a STOP condition.
When the ICTDATRm.IBIPL bit is 1:
• RI3C operates in the following order:
1. Drives the SCL to receive the IBI Data from the Target following the ACK response and receives IBI Data.
2. It stores the received IBI Data into the IBI queue.
3. Each time IBI Data of the size set by the ICQBTCR.IDSS[7:0] bits is received, the IBI Status Descriptor is stored in the IBI queue.
(5) After detection of Low of T-bit following IBI Data, issues a STOP condition.
(6) After issuing the STOP condition,
NACK response:
• If the ICINCR.RTIRN bit is 0, the IBI Status Descriptor is not stored into the IBI queue.
• If the ICINCR.RTIRN bit is 1, the IBI Status Descriptor is stored into the IBI queue.
ACK response:
• Stores the IBI Status Descriptor into the IBI queue.

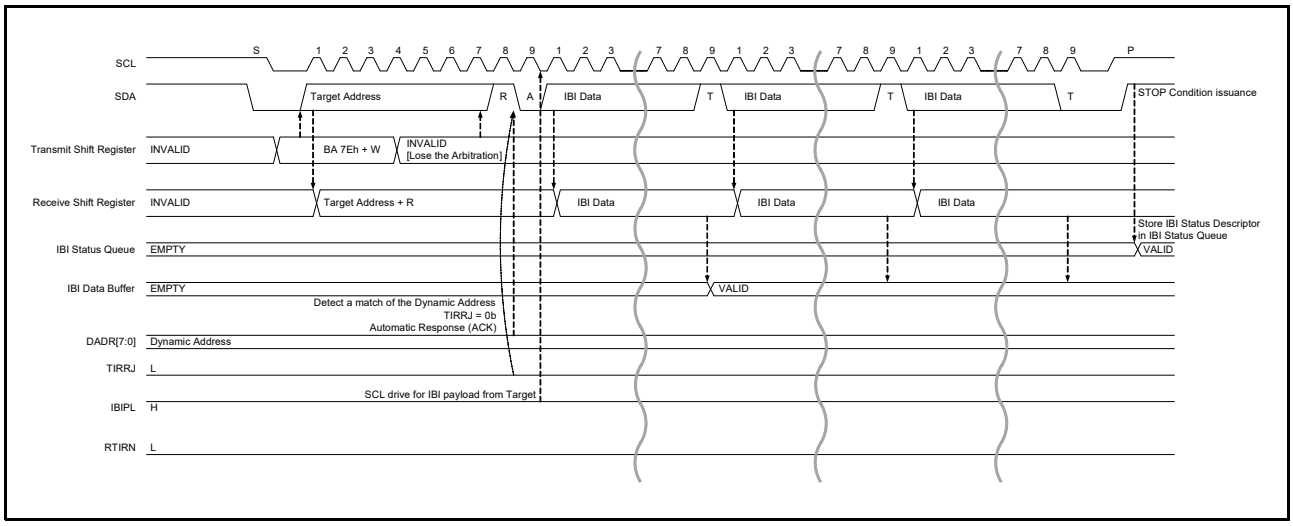


Figure 35.58 Target Interrupt Request: ACK and IBIPL = 1

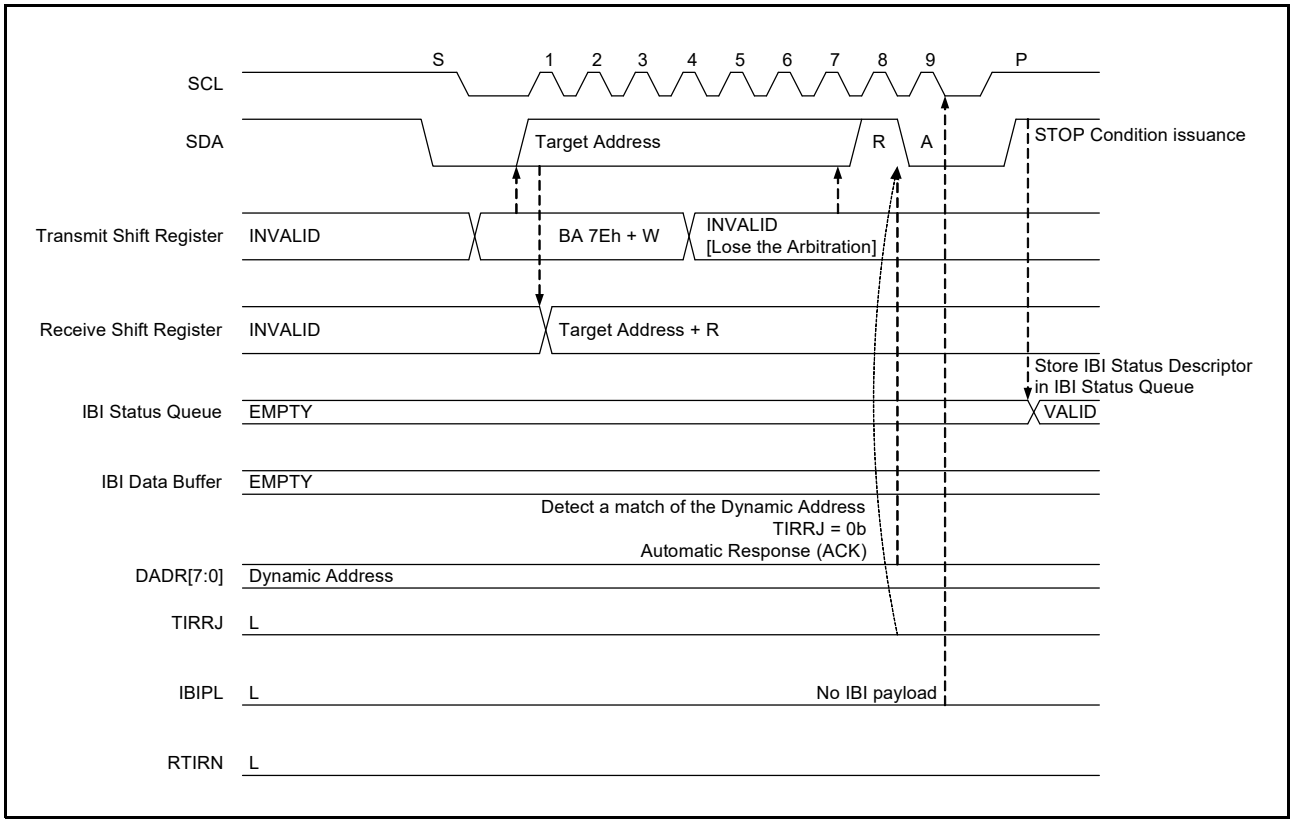


Figure 35.59 Target Interrupt Request: ACK and IBIPL = 0

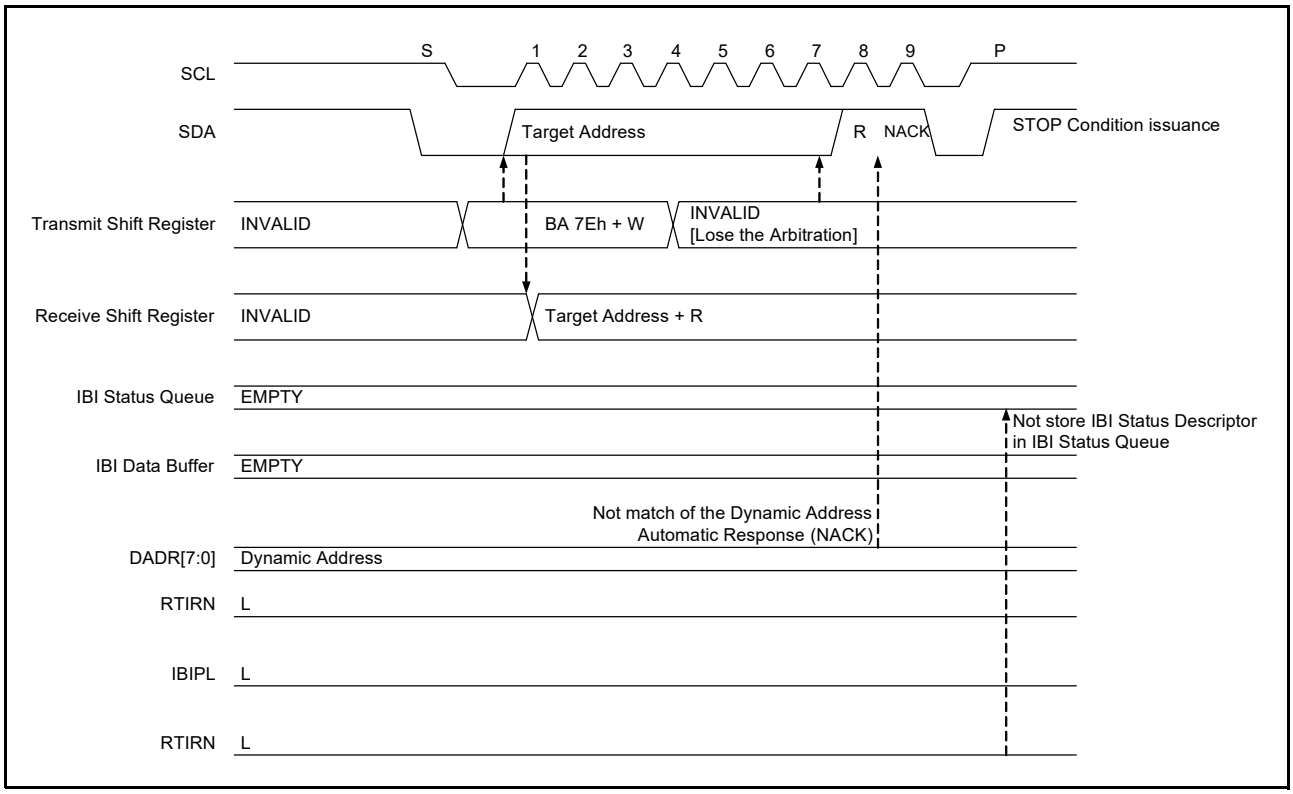


Figure 35.60 Target Interrupt Request: NACK (Not Match the DADR[7:0] Bits) and RTIRN = 0

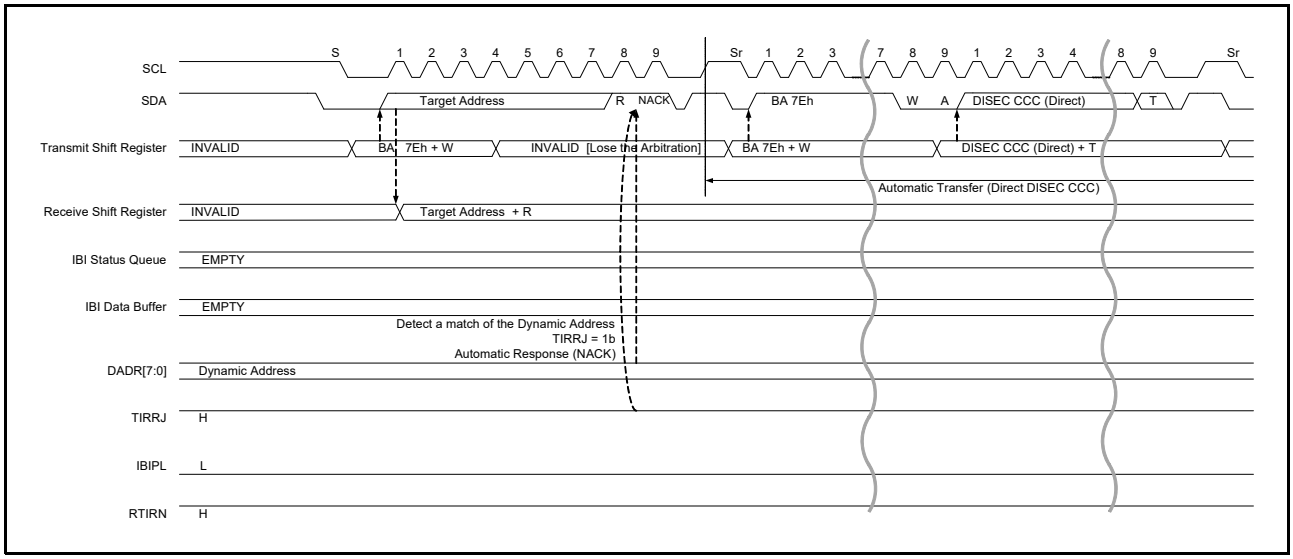


Figure 35.61 Target Interrupt Request: NACK (TIRRJ = 1) and RTIRN = 1 (1/2)

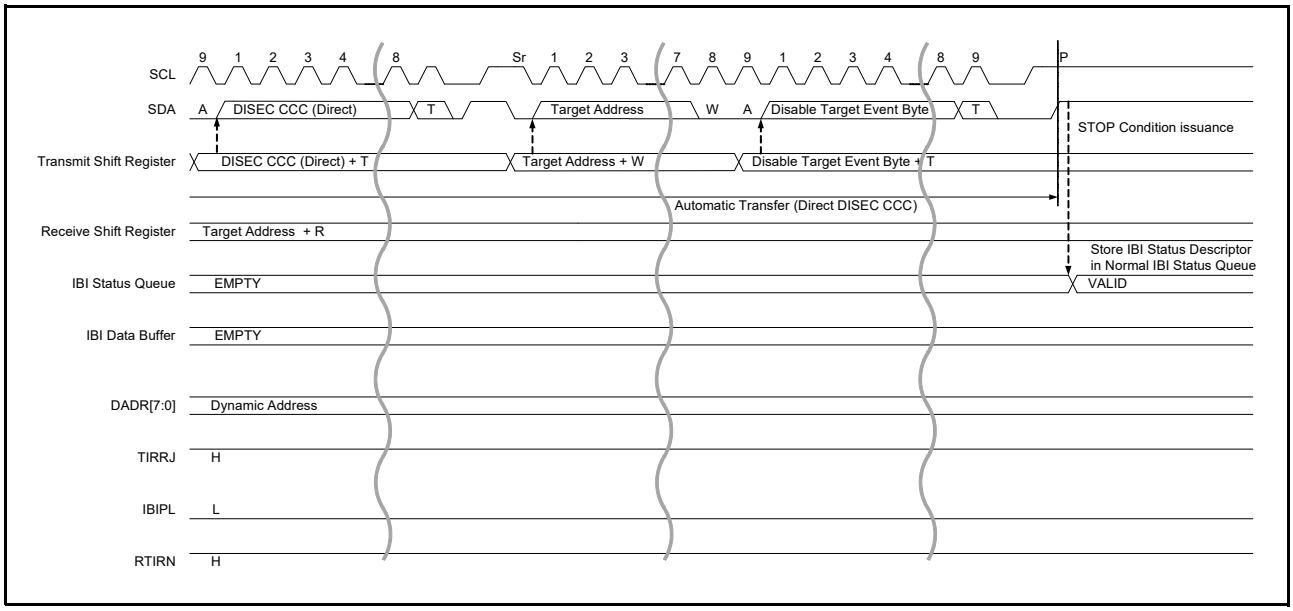


Figure 35.62 Target Interrupt Request: NACK (TIRRJ = 1) and RTIRN = 1 (2/2)

35.5.3.2 Controller Role Request

(1) Detect Target Address with RnW bit Low in Address header.

(2) Compare the detected Target Address with the DADR[7:0] bits in each DAT (the ICTDATRm register).

(3) When it does not match the ICTDATRm.DADR[7:0] bits:

• Responds with NACK, then issues a STOP condition.

When it matches the ICTDATRm.DADR[7:0] bits and the ICTDCTRm.ROLE[1:0] bits (BCR[7:6]) are “00b” (I3C Target):

• Responds with NACK, then issues a STOP condition.

When it matches the ICTDATRm.DADR[7:0] bits and the ICTDCTRm.ROLE[1:0] bits (BCR[7:6]) are “01b” (I3C Controller capable):

(a) When the ICTDATRm.CRRRJ bit is 1:

• RI3C operates in the following order.

1. Responds with NACK.
2. Issues a repeated START condition and automatically issues Direct DISEC CCC to the detected Target.
3. Issues a STOP condition.

(b) When the ICTDATRm.CRRRJ bit is 0:

• Responds with ACK, then issues a STOP condition.

(4) After issuing the STOP condition,

NACK response:

• If the ICINCR.RCRRN bit is 0, the IBI Status Descriptor is not stored into the IBI queue.

• If the ICINCR.RCRRN bit is 1, the IBI Status Descriptor is stored into the IBI queue.

ACK response:

• Stores the IBI Status Descriptor into the IBI queue.

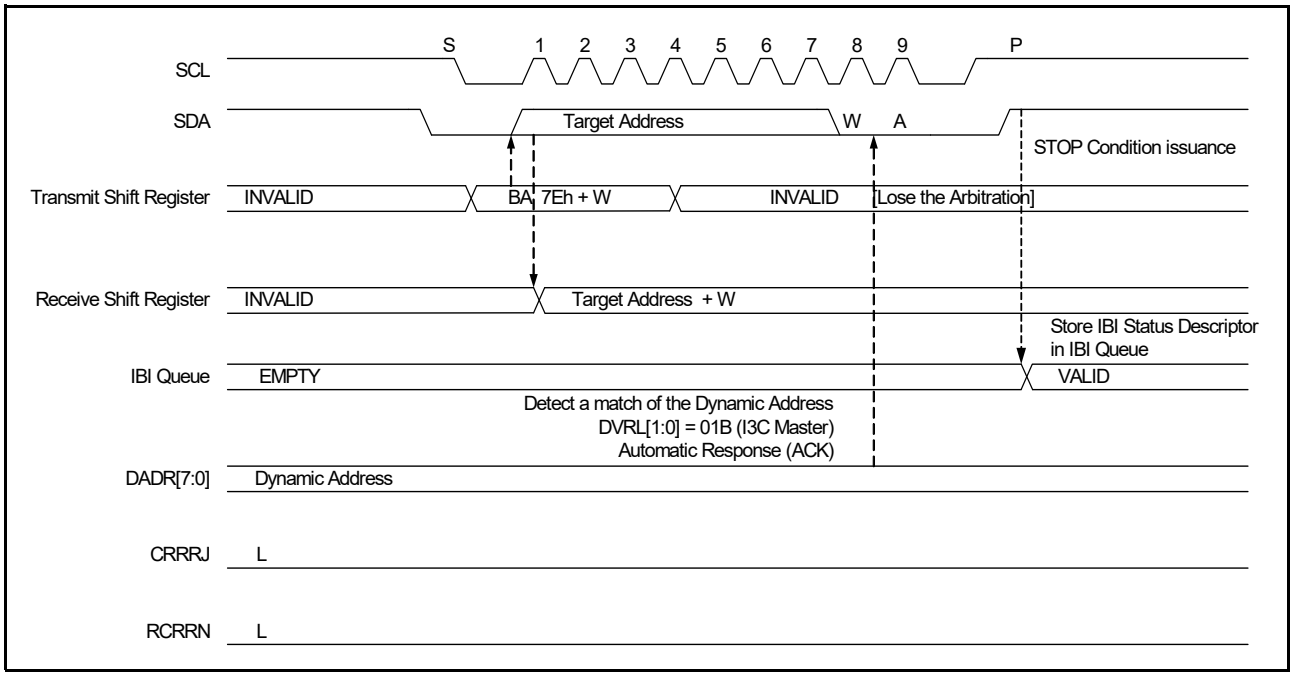


Figure 35.63 Controller Role Request: ACK

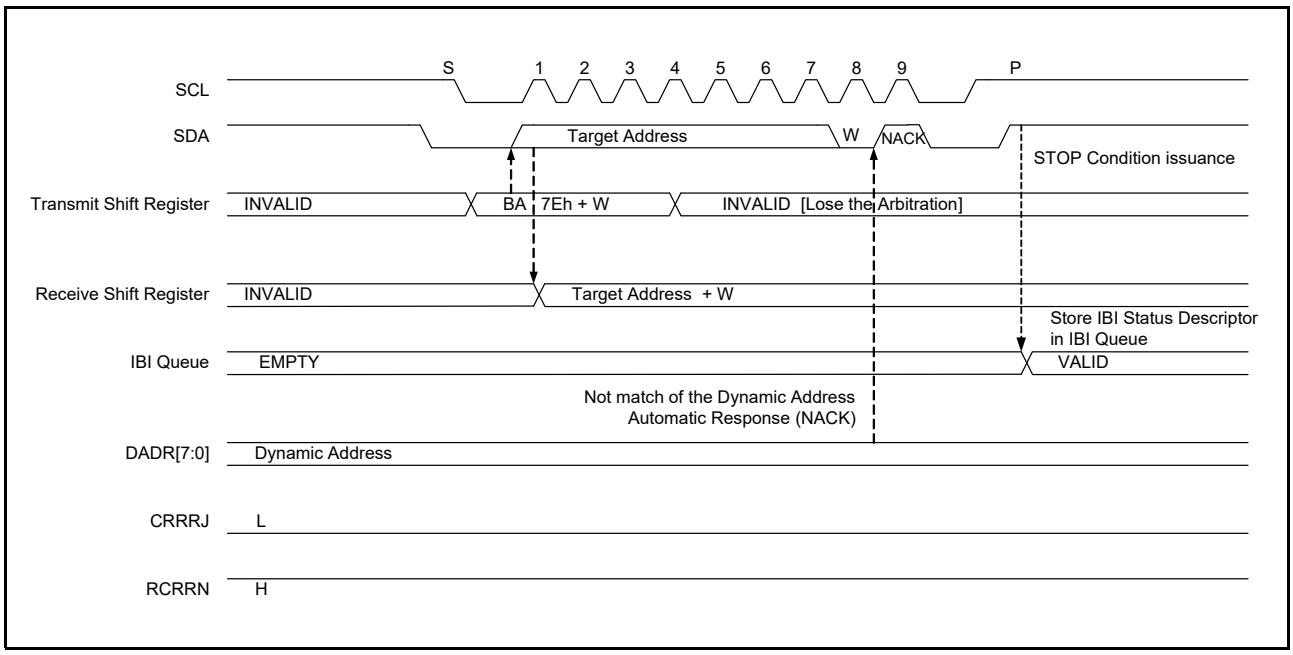


Figure 35.64 Controller Role Request: NACK (Not Match the DADR[7:0] Bits) and RCRRN = 1

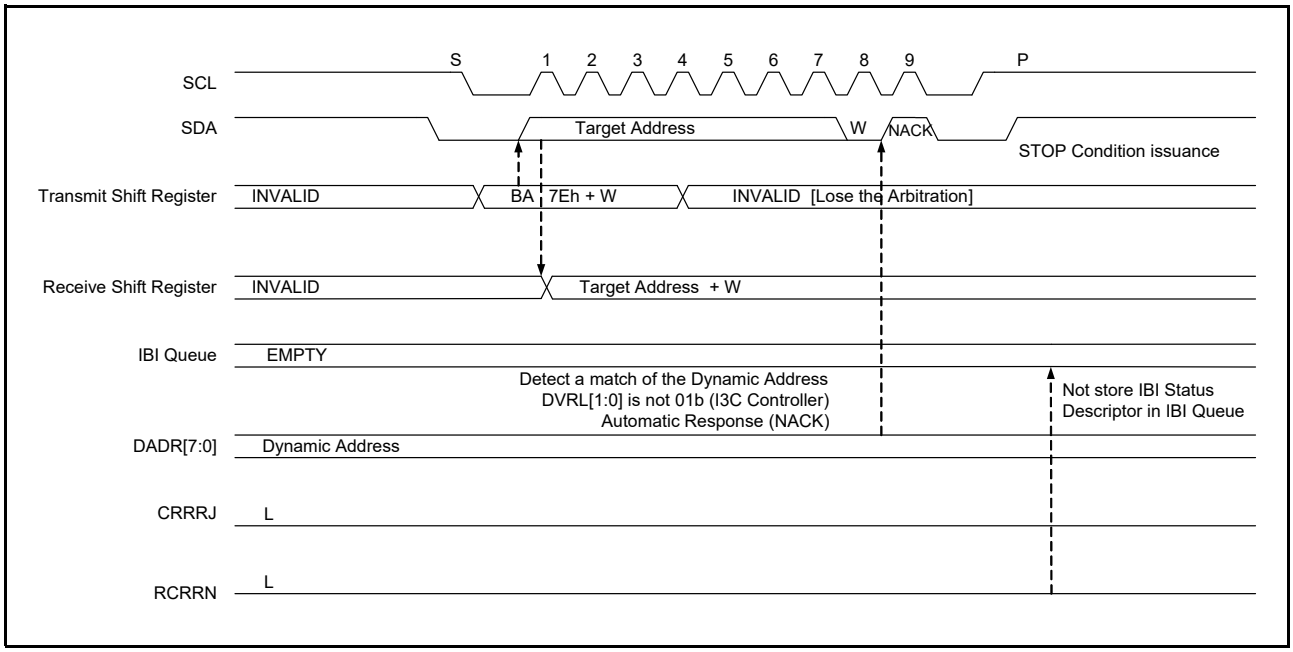


Figure 35.65 Controller Role Request: NACK (ROLE[1:0] = 00b (I3C Target)) and RCRRN = 0

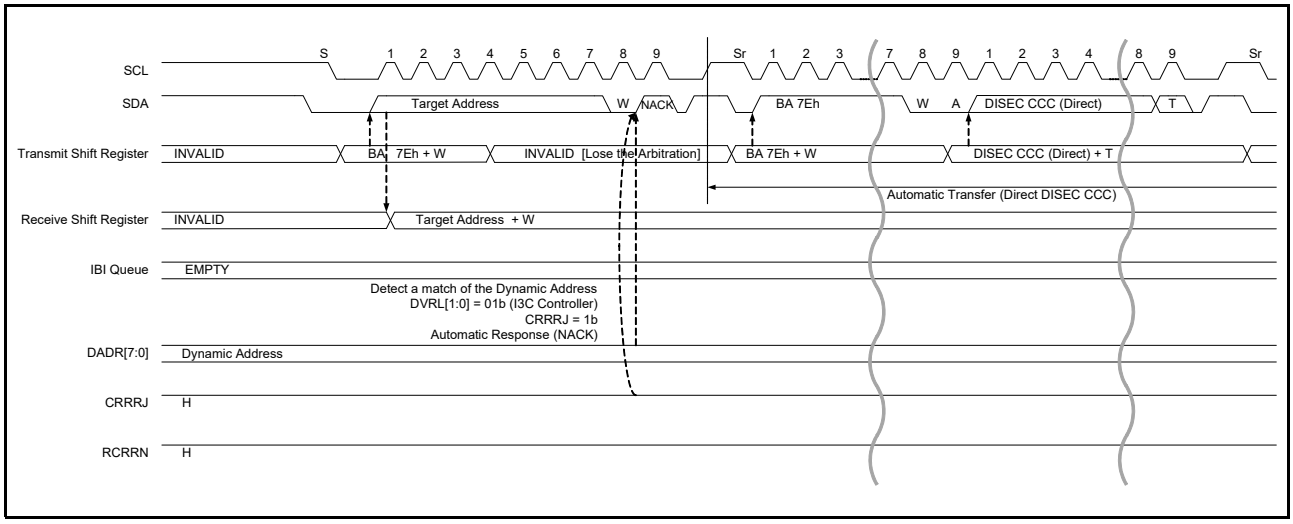


Figure 35.66 Controller Role Request: NACK (CRRRJ = 1) and RCRRN = 1 (1/2)

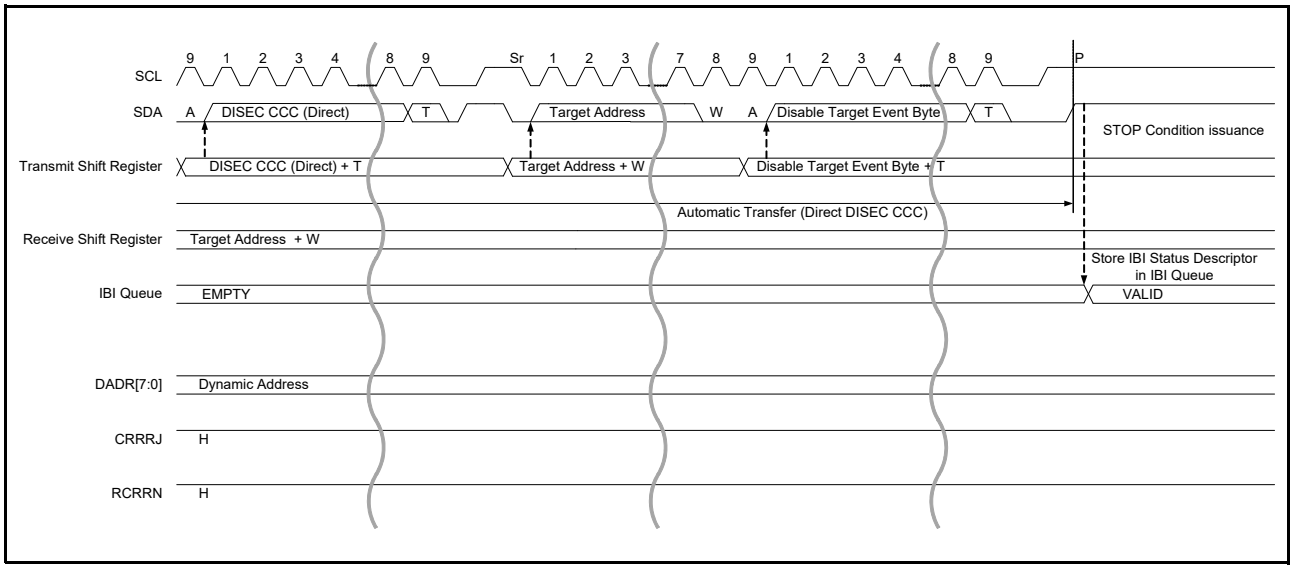


Figure 35.67 Controller Role Request: NACK (CRRRJ = 1) and RCRRN = 1 (2/2)

35.5.3.3 Hot-Join Event

(1) Detect the Hot-Join Address (02h) with RnW bit Low in the Address header.

(2) When the ICCR.HJC bit is 1:

• It operates in the following order.

1. Responds with NACK.
2. Issues a repeated START condition and automatically issues Broadcast DISEC CCC.
3. Issues a STOP condition.

When the ICCR.HJC bit is 0:

• Responds with ACK, then issues a STOP condition.

(3) After issuing the STOP condition,

NACK response:

• If the ICINCR.RHJRN bit is 0, the IBI Status Descriptor is not stored into the IBI queue.

• If the ICINCR.RHJRN bit is 1, store the IBI Status Descriptor into the IBI queue.

ACK response:

• Stores the IBI Status Descriptor into the IBI queue.

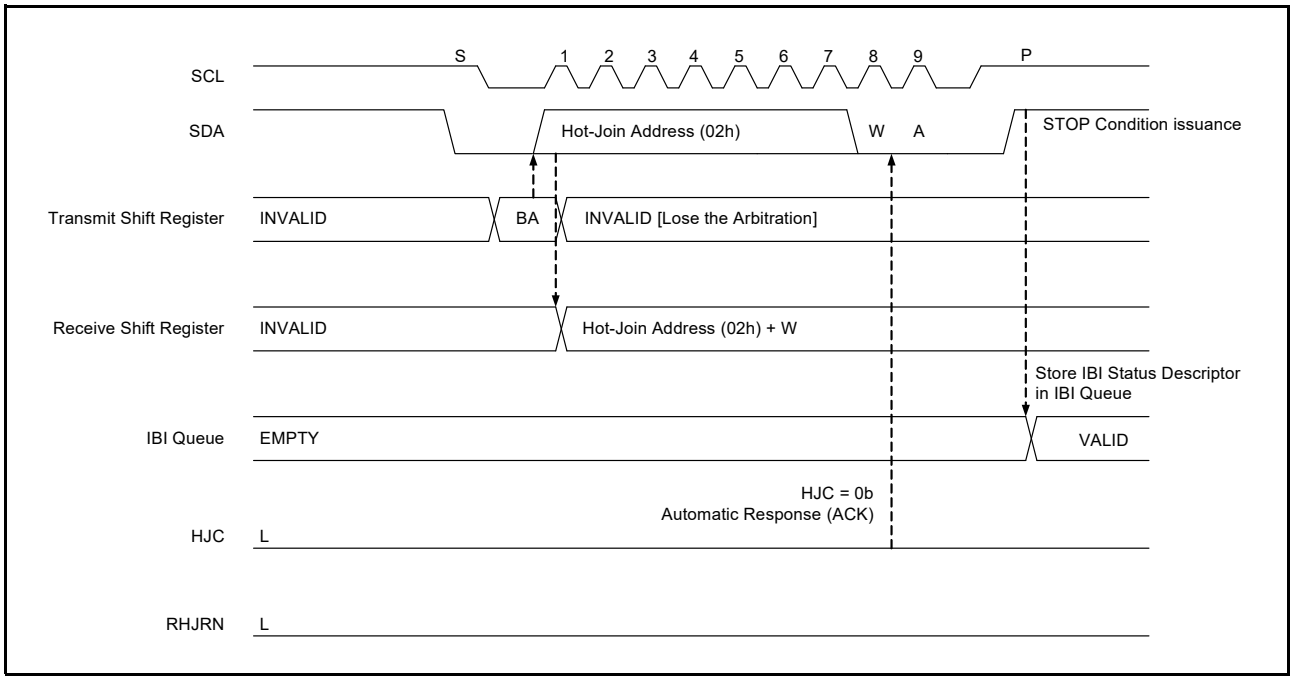


Figure 35.68 Hot-Join Event: ACK

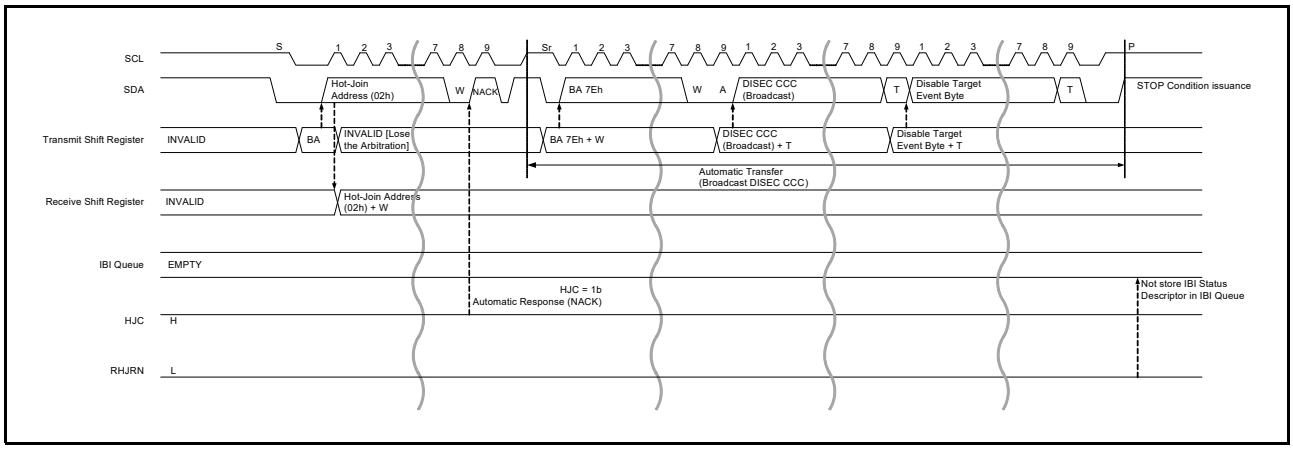


Figure 35.69 Hot-Join Event: NACK (HJC = 1) and RHJRN = 0

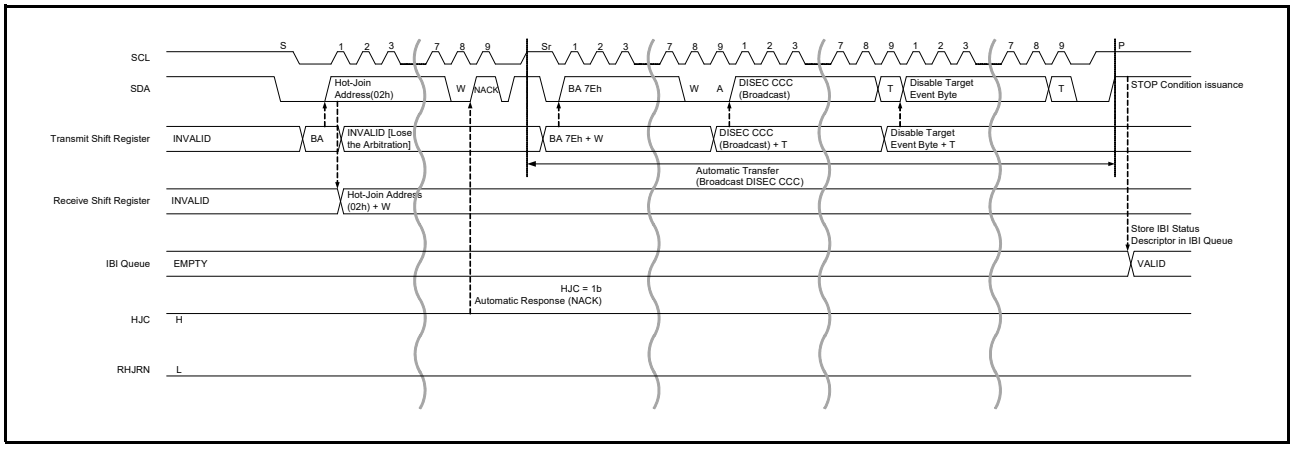


Figure 35.70 Hot-Join Event: NACK (HJC = 1) and RHJRN = 1

35.5.4 Common Command Codes (CCC)

Command Code E0h to FEh Vendor Extension-Direct CCCs defined are not supported.
The MIPI reserved area and Vendor Extension area of Command Code are not supported.
Do not use an unsupported CCC when using this module with I3C Target.
If the I3C Controller must use an unsupported CCC, use the added CCC after using ENTASx CCC to put this module to Sleep mode.

35.5.5 Error Detection

35.5.5.1 SDR Error Detection and Recovery Methods for I3C Target Devices

The seven error types summarized in Table 35.8 are supported for all I3C Target devices. Each error type is further explained below the table.

Table 35.8 SDR Target Error Types

Error Type	Description	Error Detection Method	Error Recovery Method
TE0	Broadcast Address/W (= 7Eh/W) or Dynamic Address/RW	Detect any of the following: 3Eh/W 5Eh/W 6Eh/W 76h/W 7Ah/W 7Ch/W 7Fh/W 7Eh/R	Enable HDR Exit Detector and ignore all other patterns
TE1	CCC code	Parity check, using T-Bit	Enable HDR Exit Detector and neglect other patterns
TE2	Write data	Parity check, using T-Bit	Enable STOP condition detector and neglect other patterns
TE3	Assigned address during Dynamic Address arbitration	Parity check, using PAR Bit	Generate NACK (after PAR), then wait for another repeated START condition and 7Eh/R to re-transmit the Provisioned ID
TE4	7Eh/R after Sr during Dynamic Address arbitration	Detect any value other than 7Eh/R after Sr during Dynamic Address Arbitration	Generate NACK (after 7Eh/R), then enable STOP condition detector and ignore all other patterns
TE5	Transaction after detecting CCC	Detect illegally formatted CCC	Generate NACK (after Target Address), then enable STOP condition detector and ignore all other patterns
TE6 (optional)	Monitoring error	Target detects (through monitoring) that transmitted Data differs from what it intended to transmit (Does not apply during Dynamic Address arbitration)	Stop the transmission, then enable STOP condition detector and ignore all other patterns

35.5.5.2 SDR Error Detection and Recovery Methods for I3C Controller Devices

The two error types summarized in Table 35.9 are supported for all I3C Controller devices. Each error type is further explained below the table.

Table 35.9 SDR Controller Error Types

Error Type	Description	Error Detection Method	Error Recovery Method
CE0	Transaction after sending CCC	Detect illegally formatted CCC	Stop the transmission, then send STOP condition and retry the transmission.
CE1 (optional)	Monitoring error	Controller detects (through monitoring) transmitted data different from what it intended to transmit (does not apply during Dynamic Address arbitration)	Stop the transmission, then send STOP condition and retry the transmission.
CE2	No response to Broadcast Address (7Eh)	Controller detects NACK after Broadcast Address (7Eh) transmission	Upon detection of NACK, Controller transmits HDR Exit Pattern followed by STOP condition

35.5.5.3 Timeout Error Detection

RI3C includes a timeout function for detecting when the SCL line has been stuck longer than the predetermined time. RI3C can detect an abnormal bus state by monitoring that the SCL line is stuck low or high for a predetermined time. The timeout function monitors the SCL line state and counts the low-level period or high-level period using the internal counter. The timeout function resets the internal counter each time the SCL line changes (rising or falling), but continues to count unless the SCL line changes. If the internal counter overflows due to no SCL line change, RI3C can detect the timeout and report the bus hung state.

This timeout function is enabled when ICSER.TMOE = 1. It detects a hung state that the SCL line is stuck low or high during the following conditions: (When ICTOR.TMOM[1:0] = 00b)

- The bus is busy (ICBSR.BFREE = 0) in Controller mode (ICMMR.ACF = 1).
- The bus is busy (ICBSR.BFREE = 0) in Target mode (ICMMR.ACF = 0).

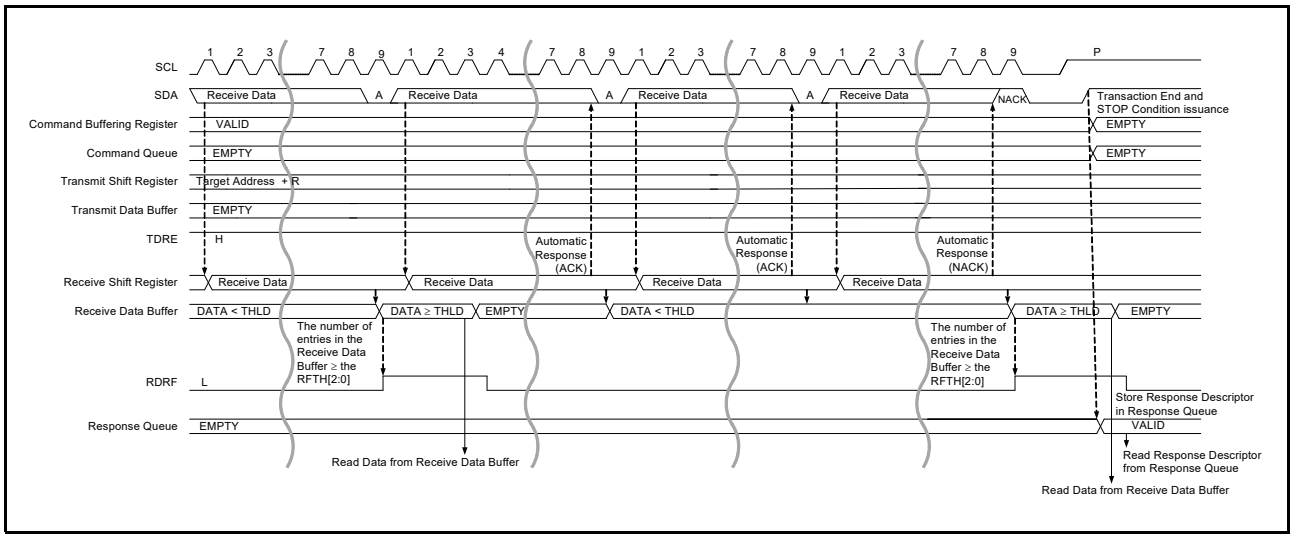


Figure 35.71 Timeout Error Detection (TMOE, TMOS[1:0], TMOH, and TMOL bits)

35.5.5.4 Resume Operation

I3C enters the Halt state as a result of any type of error occurring in a transfer.
The error type is indicated by the field ERR_STATUS in Response Descriptor or Receive Status Descriptor. After I3C has entered the Halt state, the user must write the value 1 to the RESUME bit to resume operation. I3C shall auto-clear the RESUME bit once it has initiated the next Command transfer or detected the START condition.

35.5.5.5 Abort Operation

When the ICCR.ABORT bit is set to 1, RI3C relinquish control of the bus before completing the currently issued transfer. In response to an Abort request, RI3C issues the STOP condition on the bus after the complete data byte is transmitted or received. After RI3C has aborted, the user shall clear the ICCR.ABORT bit to allow operation on the bus.

Note: For Read transaction, received data are stored in the receive data buffer even if the ICCR.ABORT bit is set to 1.

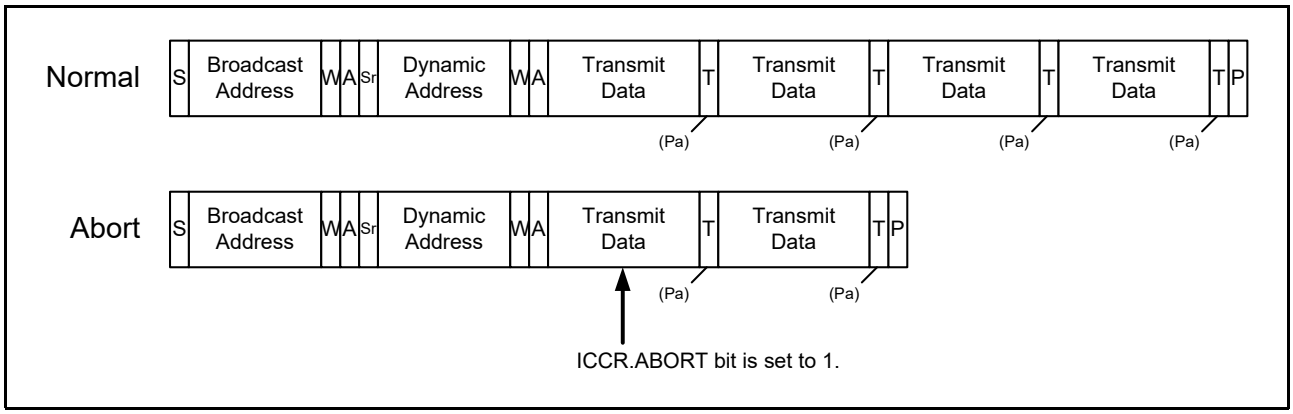


Figure 35.72 Abort Operation of SDR Write Transfer

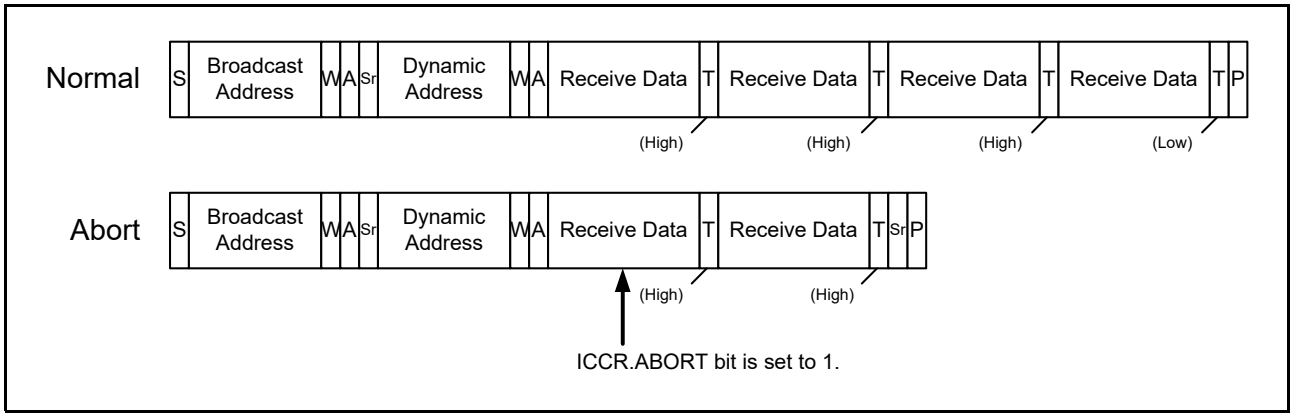


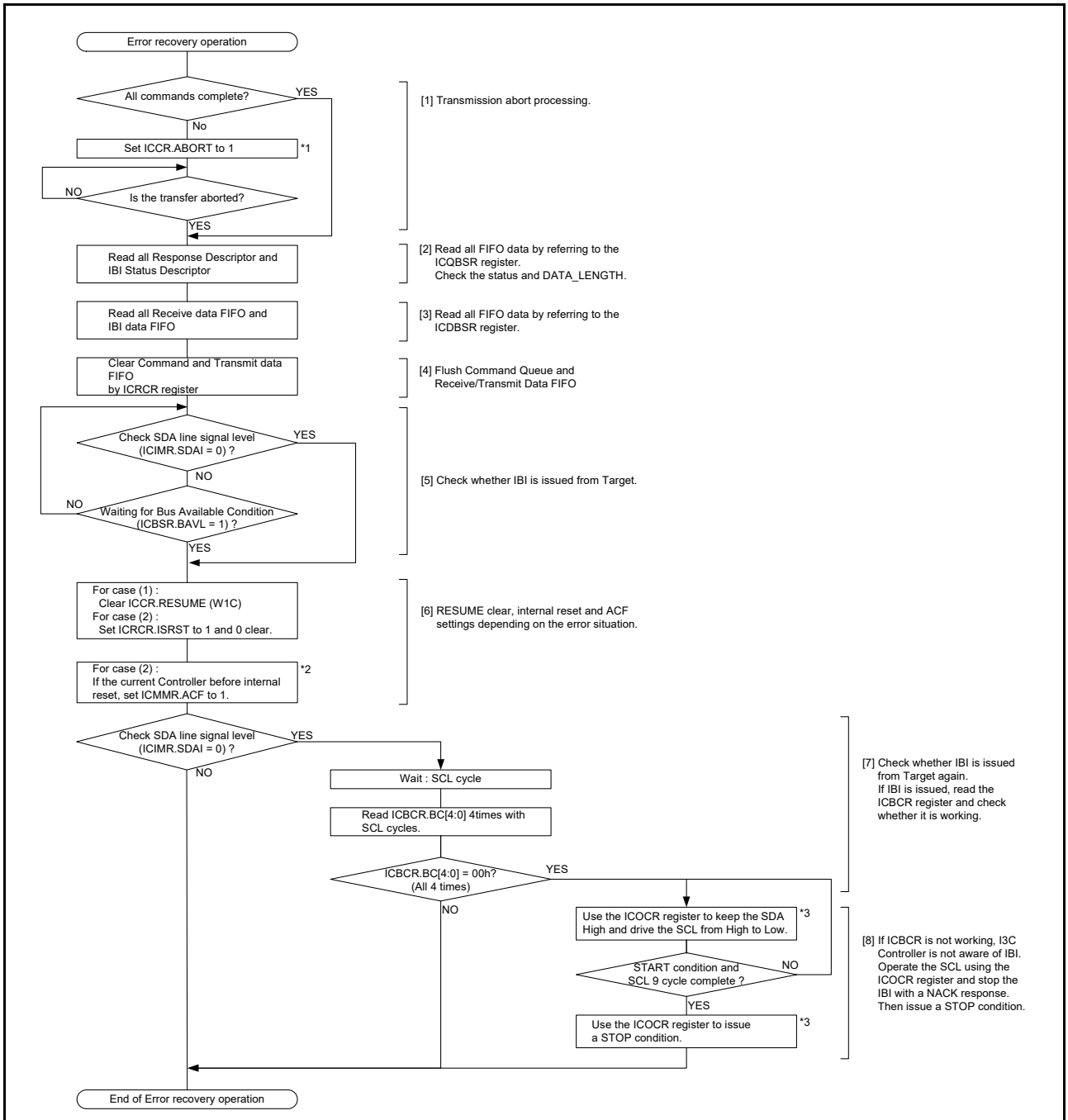
Figure 35.73 Abort Operation of SDR Read Transfer

35.5.5.6 Error Recovery Operation

When an error occurs, the ICISR.BERF, ICCSR.DTEF, and ICCSR.DTAF flags are set to 1 according to the cause of the error, or the interrupts associated with each flag are asserted (when detection and interrupts are enabled).
There is a possibility of communication error or internal module error.
The I3C Controller must perform an error recovery flow according to the following case:

- When DTEF is detected.

Figure 35.74 and Figure 35.75 show the error recovery flow.



Note 1. If the transfer is not aborted due to an error, set the ICCR.ABORT bit to 1 and then abort the transfer.
Note 2. When writing to the ACF flag, set ICMMR.WP bit to 1 at the same time.
Note 3. When writing to the SDAO and SCLO bit, set ICOCR.SOWP bit to 1 at the same time.

Figure 35.74 Example of Error Recovery Operation Flowchart for I3C Controller

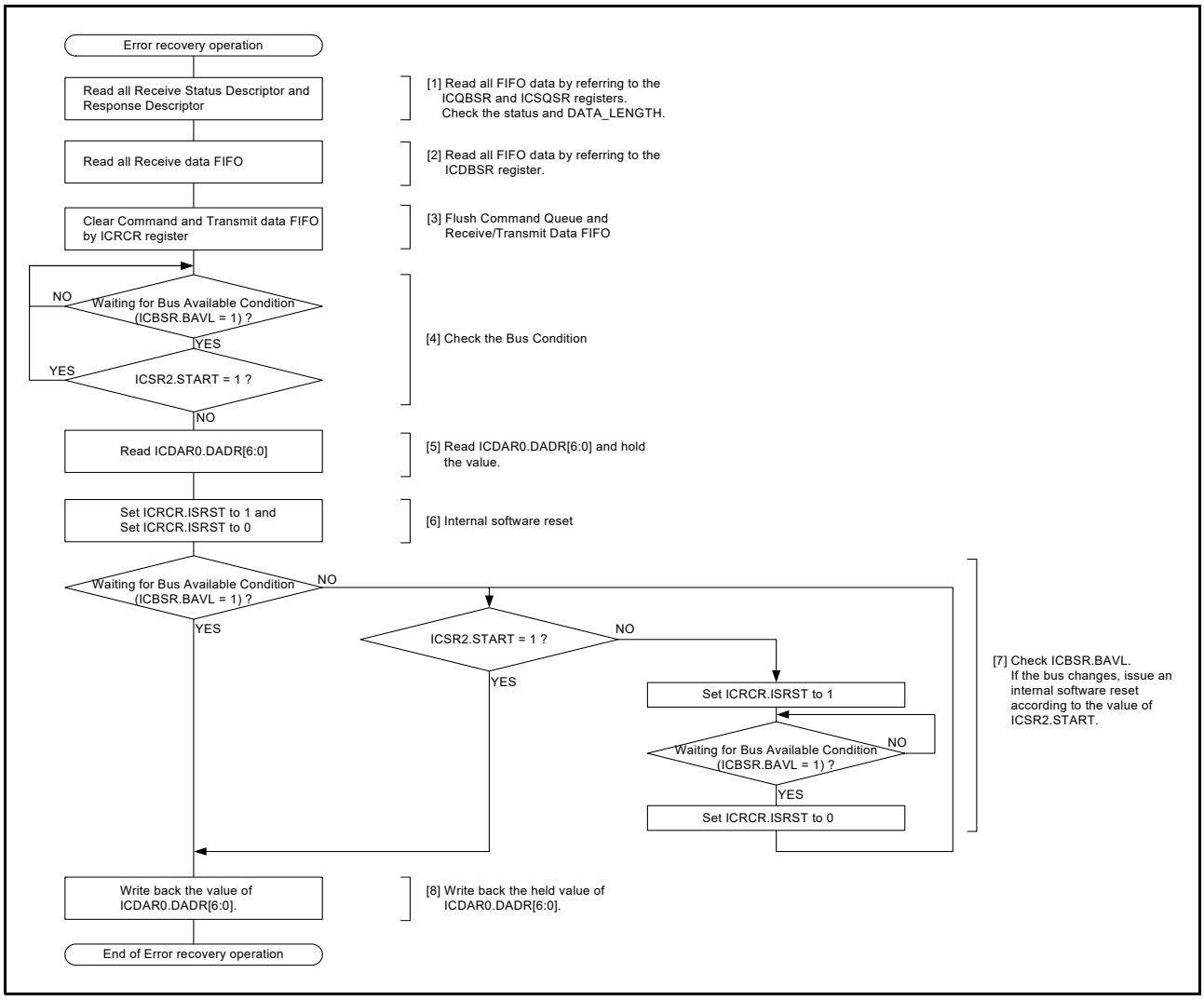


Figure 35.75 Example of Error Recovery Operation Flowchart for I3C Target

35.6 Interrupt Sources

The RI3C generates seven types of interrupt requests: transmit data empty, receive data full, IBI queue empty/full, command queue empty, response queue full, receive status queue full, and communication error or communication event (START condition detection, STOP condition detection, HDR Exit Pattern detection, timeout detection, buffer access error detection, data transfer abort, and data transfer error).

Table 35.10 lists details of the interrupt requests. The transmit data empty, receive data full, IBI queue empty/full, command queue empty, response queue full, and receive status queue full interrupt requests allow the DTC or DMAC to start data transfer.

Table 35.10 Interrupt Sources

Symbol	Interrupt Source	Interrupt Flag	Interrupt Enable Bit	Start DTC/DMA Transfer	Interrupt Generation Condition
TXI*1	Transmit data empty	ICCSR.TDRE	ICCSIER.TIE	Possible	TDRE = 1 and TIE = 1
RXI*2	Receive data full	ICCSR.RDRF	ICCSIER.RIE	Possible	RDRF = 1 and RIE = 1
IBII*3	IBI queue empty/full	ICCSR.IQEFF	ICCSIER.IQEFIE	Possible	IQEFF = 1 and IQEFIE = 1
CMDI*3	Command queue empty	ICCSR.CQEF	ICCSIER.CQEIE	Possible	CQEF = 1 and CQEIE = 1
RESPI*3	Response queue full	ICCSR.RQFF	ICCSIER.RQFIE	Possible	RQFF = 1 and RQFIE = 1
RCVI*3	Receive status queue full	ICCSR.SQFF	ICCSIER.SQFIE	Possible	SQFF = 1 and SQFIE = 1
EEI	Communication error/ communication event	ICSR2.START	ICSIER.STIE	Not possible	START = 1 and STIE = 1
		ICSR2.STOP	ICSIER.SPIE		STOP = 1 and SPIE = 1
		ICSR2.HDRXDF	ICSIER.HDRXIE		HDRXDF = 1 and HDRXIE = 1
		ICSR2.TMOF	ICSIER.TMOIE		TMOF = 1 and TMOIE = 1
		ICISR.BERF	ICISIER.BERIE		BERF = 1 and BERIE = 1
		ICCSR.DTAF	ICCSIER.DTAIE		DTAF = 1 and DTAIE = 1
		ICCSR.DTEF	ICCSIER.DTEIE		DTEF = 1 and DTEIE = 1

Note: There is a delay between the execution of a write instruction for a peripheral module by the CPU and actual writing to the module. Thus, when an interrupt flag has been cleared or an interrupt request has been masked, read the relevant flag again to check whether clearing or masking has been completed, and then return from interrupt handling. Returning from interrupt handling without checking that writing to the module has been completed creates a possibility of repeated processing of the same interrupt.

Note 1. Because TXI is an edge-detected interrupted, it does not require clearing. Furthermore, the ICCSR.TDRE flag (a condition for TXI) is automatically set to 0 when transmit data is written to the ICDR register or when a STOP condition is detected (ICSR2.STOP flag is 1).

Note 2. Because RXI is an edge-detected interrupted, it does not require clearing. Furthermore, the ICCSR.RDRF flag (a condition for RXI) is automatically set to 0 when data is read from the ICDR register.

Note 3. Because these interrupts are edge-detected interrupted, they do not require clearing. Furthermore, the each flag (a condition for interrupt) is automatically set to 0 when the corresponding queue is read or written to.

Clear each flag or mask the interrupt request during interrupt handling.

35.6.1 Buffer Operation for Empty and Full Interrupts

If the conditions for generating a Empty or Full interrupt is satisfied while the corresponding ICU.IRn.IR flag is 1, the interrupt request is not output to the ICU but retained internally (up to one request can be retained internally for each source).

When the IR flag becomes 0, the retained interrupt request is output to the ICU. The retained interrupt request is automatically cleared under normal conditions of usage.

The retained interrupt request can also be cleared by writing 0 to the corresponding interrupt enable bit in the ICCSIER register.

35.7 Event Link Function

The RI3C outputs the corresponding event signal to the event link controller (ELC) when any of the following events occur.

- Transmit data empty
- Receive data full
- Communication error/communication event

35.7.1 Interrupt Handling and Event Linking

Each of the RI3C interrupts: transmit data empty, receive data full, and communication error/communication event (START condition detection, STOP condition detection, and timeout detection) has an enable bit that controls whether the interrupt is enabled or disabled. When an interrupt request source occurs, an interrupt request signal is output to the ICU if the corresponding interrupt enable bit is 1.
On the other hand, the event signal is transmitted via the ELC to other modules when the corresponding interrupt request source occurs, regardless of the interrupt enable bit.
For details on interrupt sources, refer to Table 35.10.

35.8 Reset Description

Table 35.11 Reset Control of Registers (1/5)

Register	Bit	System Reset	ICRCR Register	
			MRST Bit	ISRST Bit
ICMR	OMS	To be reset	To be reset	Retained
ICCR	ICE	To be reset	To be reset	Retained
	RESUME	To be reset	To be reset	To be reset
	ABORT	To be reset	Retained	Retained
	HJC	To be reset	Retained	Retained
	IBAINC	To be reset	Retained	Retained
ICCAR	DAV	To be reset	Retained	Retained
	DADR[6:0]	To be reset	Retained	Retained
ICRCR	ISRST	To be reset	To be reset	Retained
	SQRST	To be reset	To be reset	To be reset
	IQRST	To be reset	To be reset	To be reset
	RBRST	To be reset	To be reset	To be reset
	TBRST	To be reset	To be reset	To be reset
	RQRST	To be reset	To be reset	To be reset
	CQRST	To be reset	To be reset	To be reset
	MRST	To be reset	Retained	Retained
ICMMR	WP	To be reset	To be reset	To be reset
	ACF	To be reset	To be reset	To be reset
ICISR	BERF	To be reset	To be reset	To be reset
ICISER	BERDE	To be reset	To be reset	Retained
ICISIER	BERIE	To be reset	To be reset	Retained
ICDCTIR	INDEX[4:0]	To be reset	To be reset	To be reset
ICINCR	RTIRN	To be reset	To be reset	Retained
	RCRRN	To be reset	To be reset	Retained
	RHJRN	To be reset	To be reset	Retained
ICTCR	TA0DE	To be reset	To be reset	Retained
ICSBR	ODDBL	To be reset	To be reset	Retained
	PPHW[5:0]	To be reset	To be reset	Retained
	PPLW[5:0]	To be reset	To be reset	Retained
	ODHW[7:0]	To be reset	To be reset	Retained
	ODLW[7:0]	To be reset	To be reset	Retained
ICEBR	PPHW[5:0]	To be reset	To be reset	Retained
	PPLW[5:0]	To be reset	To be reset	Retained
	ODHW[7:0]	To be reset	To be reset	Retained
	ODLW[7:0]	To be reset	To be reset	Retained
ICBFTR		To be reset	To be reset	Retained
ICBATR		To be reset	To be reset	Retained
ICBITR		To be reset	To be reset	Retained
ICOCR	SOWP	To be reset	To be reset	To be reset
	SCLO	To be reset	To be reset	Retained
	SDAO	To be reset	To be reset	Retained

Table 35.11 Reset Control of Registers (2/5)

Register	Bit	System Reset	ICRCR Register	
			MRST Bit	ISRST Bit
ICTOR	TMOM[1:0]	To be reset	To be reset	Retained
	TMOH	To be reset	To be reset	Retained
	TMOL	To be reset	To be reset	Retained
	TMOS[1:0]	To be reset	To be reset	Retained
ICSTCR	APSE	To be reset	To be reset	Retained
	PBSE	To be reset	To be reset	Retained
	AASE	To be reset	To be reset	Retained
	STT[15:0]	To be reset	To be reset	Retained
ICTDLR	DLGTH[15:0]	To be reset	To be reset	Retained
ICCQR		To be reset	To be reset	To be reset
ICRQR		To be reset	To be reset	To be reset
ICDR		To be reset	To be reset	To be reset
ICIQR		To be reset	To be reset	To be reset
ICSQR		To be reset	To be reset	To be reset
ICQBTCR	IQTH[7:0]	To be reset	To be reset	Retained
	IDSS[7:0]	To be reset	To be reset	Retained
	RQTH[7:0]	To be reset	To be reset	Retained
	CQTH[7:0]	To be reset	To be reset	Retained
ICDBTCR	RSTH[2:0]	To be reset	To be reset	Retained
	TSTH[2:0]	To be reset	To be reset	Retained
	RFTH[2:0]	To be reset	To be reset	Retained
	TETH[2:0]	To be reset	To be reset	Retained
ICSQTCR	SQTH[7:0]	To be reset	To be reset	Retained
ICSR2	TMOF	To be reset	To be reset	To be reset
	HDRXDF	To be reset	To be reset	To be reset
	STOP	To be reset	To be reset	To be reset
	START	To be reset	To be reset	To be reset
ICSER	TMOE	To be reset	To be reset	Retained
	HDRXDE	To be reset	To be reset	Retained
	SPDE	To be reset	To be reset	Retained
	STDE	To be reset	To be reset	Retained
ICSIER	TMOIE	To be reset	To be reset	Retained
	HDRXIE	To be reset	To be reset	Retained
	SPIE	To be reset	To be reset	Retained
	STIE	To be reset	To be reset	Retained
ICCSR	SQFF	To be reset	To be reset	To be reset
	DTEF	To be reset	To be reset	To be reset
	DTAF	To be reset	To be reset	To be reset
	RQFF	To be reset	To be reset	To be reset
	CQEF	To be reset	To be reset	To be reset
	IQEFF	To be reset	To be reset	To be reset
	RDRF	To be reset	To be reset	To be reset
	TDRE	To be reset	To be reset	To be reset

Table 35.11 Reset Control of Registers (3/5)

Register	Bit	System Reset	ICRCR Register	
			MRST Bit	ISRST Bit
ICCSER	SQFDE	To be reset	To be reset	Retained
	DTEDE	To be reset	To be reset	Retained
	DTADE	To be reset	To be reset	Retained
	RQFDE	To be reset	To be reset	Retained
	CQEDE	To be reset	To be reset	Retained
	IQEFDE	To be reset	To be reset	Retained
	RDE	To be reset	To be reset	Retained
	TDE	To be reset	To be reset	Retained
ICCSIER	SQFIE	To be reset	To be reset	Retained
	DTEIE	To be reset	To be reset	Retained
	DTAIE	To be reset	To be reset	Retained
	RQFIE	To be reset	To be reset	Retained
	CQEIE	To be reset	To be reset	Retained
	IQEFIE	To be reset	To be reset	Retained
	RIE	To be reset	To be reset	Retained
	TIE	To be reset	To be reset	Retained
ICBSR	BIDL	To be reset	To be reset	Retained
	BAVL	To be reset	To be reset	Retained
	BFREE	To be reset	To be reset	Retained
ICTDATRm (m = 0 to 3)	TYPE	To be reset	To be reset	Retained
	NACKRC[1:0]	To be reset	To be reset	Retained
	DADR[7:0]	To be reset	To be reset	Retained
	IBITSE	To be reset	To be reset	Retained
	CRRRJ	To be reset	To be reset	Retained
	TIRRJ	To be reset	To be reset	Retained
	IBIPL	To be reset	To be reset	Retained
	SADR[6:0]	To be reset	To be reset	Retained
ICEDATR	TYPE	To be reset	To be reset	Retained
	NACKRC[1:0]	To be reset	To be reset	Retained
	DADR[7:0]	To be reset	To be reset	Retained
	SADR[6:0]	To be reset	To be reset	Retained
ICDAR0	DADR[6:0]	To be reset	To be reset	Retained
	IBIPL	To be reset	To be reset	Retained
	SADR[9:0]	To be reset	To be reset	Retained
ICTDCTRm (m = 0 to 3)	ROLE[1:0]	To be reset	To be reset	Retained
	OFLC	To be reset	To be reset	Retained
	IBIPL	To be reset	To be reset	Retained
	IBIRQC	To be reset	To be reset	Retained
	LIMIT	To be reset	To be reset	Retained

Table 35.11 Reset Control of Registers (4/5)

Register	Bit	System Reset	ICRCR Register	
			MRST Bit	ISRST Bit
ICDCTR	ROLE[1:0]	To be reset	To be reset	Retained
	OFLC	To be reset	To be reset	Retained
	IBIPL	To be reset	To be reset	Retained
	IBIRQC	To be reset	To be reset	Retained
	LIMIT	To be reset	To be reset	Retained
	DCR[7:0]	To be reset	To be reset	Retained
ICPIDLR		To be reset	To be reset	Retained
ICPIDHR		To be reset	To be reset	Retained
ICDAMR0	DAV	To be reset	To be reset	Retained
	SAV	To be reset	To be reset	Retained
	TADR[9:0]	To be reset	To be reset	Retained
ICTEVR	ENHJ	To be reset	To be reset	Retained
	ENCR	To be reset	To be reset	Retained
	ENINT	To be reset	To be reset	Retained
ICASR	ENTAS3	To be reset	To be reset	Retained
	ENTAS2	To be reset	To be reset	Retained
	ENTAS1	To be reset	To be reset	Retained
	ENTAS0	To be reset	To be reset	Retained
ICMWLR	MWL[15:0]	To be reset	To be reset	Retained
ICMRLR	IBIPL[7:0]	To be reset	To be reset	Retained
	MRL[15:0]	To be reset	To be reset	Retained
ICTMR	TM[7:0]	To be reset	To be reset	Retained
ICDSR	VRSV[7:0]	To be reset	To be reset	Retained
	CAS[1:0]	To be reset	To be reset	Retained
	PERR	To be reset	To be reset	Retained
	PNDINT[3:0]	To be reset	To be reset	To be reset
ICMWSR	MSWDR[2:0]	To be reset	To be reset	Retained
ICMRSR	TSCO[2:0]	To be reset	To be reset	Retained
	MSRDR[2:0]	To be reset	To be reset	Retained
ICMTTR	MRTTE	To be reset	To be reset	Retained
	MRTT[23:0]	To be reset	To be reset	Retained
ICTSIR	INAC[7:0]	To be reset	To be reset	Retained
	FREQ[7:0]	To be reset	To be reset	Retained
ICBCR	BC[4:0]	To be reset	To be reset	To be reset
ICQBSR	ISC[4:0]	To be reset	To be reset	To be reset
	IQFL[7:0]	To be reset	To be reset	To be reset
	RQFL[7:0]	To be reset	To be reset	To be reset
	CQFL[7:0]	To be reset	To be reset	To be reset
ICDBSR	RBFL[7:0]	To be reset	To be reset	To be reset
	TBFL[7:0]	To be reset	To be reset	To be reset
ICSQSR	SQFL[7:0]	To be reset	To be reset	To be reset

Table 35.11 Reset Control of Registers (5/5)

Register	Bit	System Reset	ICRCR Register	
			MRST Bit	ISRST Bit
ICIMR	SDAO	To be reset	To be reset	To be reset
	SCLO	To be reset	To be reset	To be reset
	SDAI	To be reset	To be reset	Retained
	SCLI	To be reset	To be reset	Retained
ICCECR	CE2C[7:0]	To be reset	To be reset	To be reset

35.9 Usage Notes

35.9.1 Settings of the Module Stop Function

RI3C operation can be disabled or enabled by module stop control register D (MSTPCRD).
After a reset, the RI3C is stopped. The registers become accessible when it is released from the module stop state. For details, refer to section 11, Low Power Consumption.

36. CAN FD Module (CANFD)

36.1 Overview

This MCU implements one channel of the CAN FD (Controller Area Network with Flexible Data Rate) module that complies with the ISO 11898-1:2015 Standard.

Table 36.1 lists the specifications of the CAN FD module, and Figure 36.1 shows a block diagram of the CAN FD module.

Table 36.1 CAN FD Module Specifications

Item	Description
Protocol	ISO 11898-1:2015 compliant
Data transfer rate	Arbitration phase: up to 1 Mbps Data phase: up to 8 Mbps*1
Operating frequency*2	Register block: up to 60 MHz (PCLKB) Message buffer RAM: up to 120 MHz (PCLKA)
Operating clock for data link layer (DLL clock)	Up to 60 MHz (either CANFDMCLK or CANFDCLK can be selected)
Frame types	Classic CAN (CAN 2.0) • Data frame in base format (11-bit ID) • Data frame in extended format (29-bit ID) • Remote frame in base format (11-bit ID) • Remote frame in extended format (29-bit ID) CAN FD*1 • Data frame in base format (11-bit ID) • Data frame in extended format (29-bit ID)
Data length	Classic CAN: 0 to 8 bytes CAN FD: 0 to 8, 12, 16, 20, 24, 32, 48, or 64 bytes*1
Message buffers	• 32 receive message buffers • Four transmit message buffers • One transmit queue Messages can be automatically transfered to the transmit queue.
FIFOs	Variable size FIFO buffers • Two receive FIFOs • One common FIFO that can be configured as a receive FIFO or transmit FIFO
Automatic transmission interval adjustment	Available when the common FIFO is configured as a transmit FIFO The interval between messages sent from the FIFO can be adjusted.
Acceptance filter	Filterable in the following fields: • IDE bit (base format, extended format, or both) • ID field • RTR bit (data frame or remote frame) (only for Classic CAN) • DLC field (data length) Protection function when the payload size is exceeded Acceptance filter list (AFL) entries can be updated during communication.
Software support	Label information is automatically added to received messages
Timer	Transmission and reception timestamp function
Power down function	Module start/stop function for each CAN node (CH_SLEEP and GL_SLEEP mode) Transition to module stop state is possible.
RAM	RAM with ECC protection

Note 1. This is only available for products that support the CAN FD protocol.
Note 2. The frequency ratio of PCLKA and PCLKB should be 2 : 1. Also, the frequency of PCLKB should be equal to or higher than that of the DLL clock.

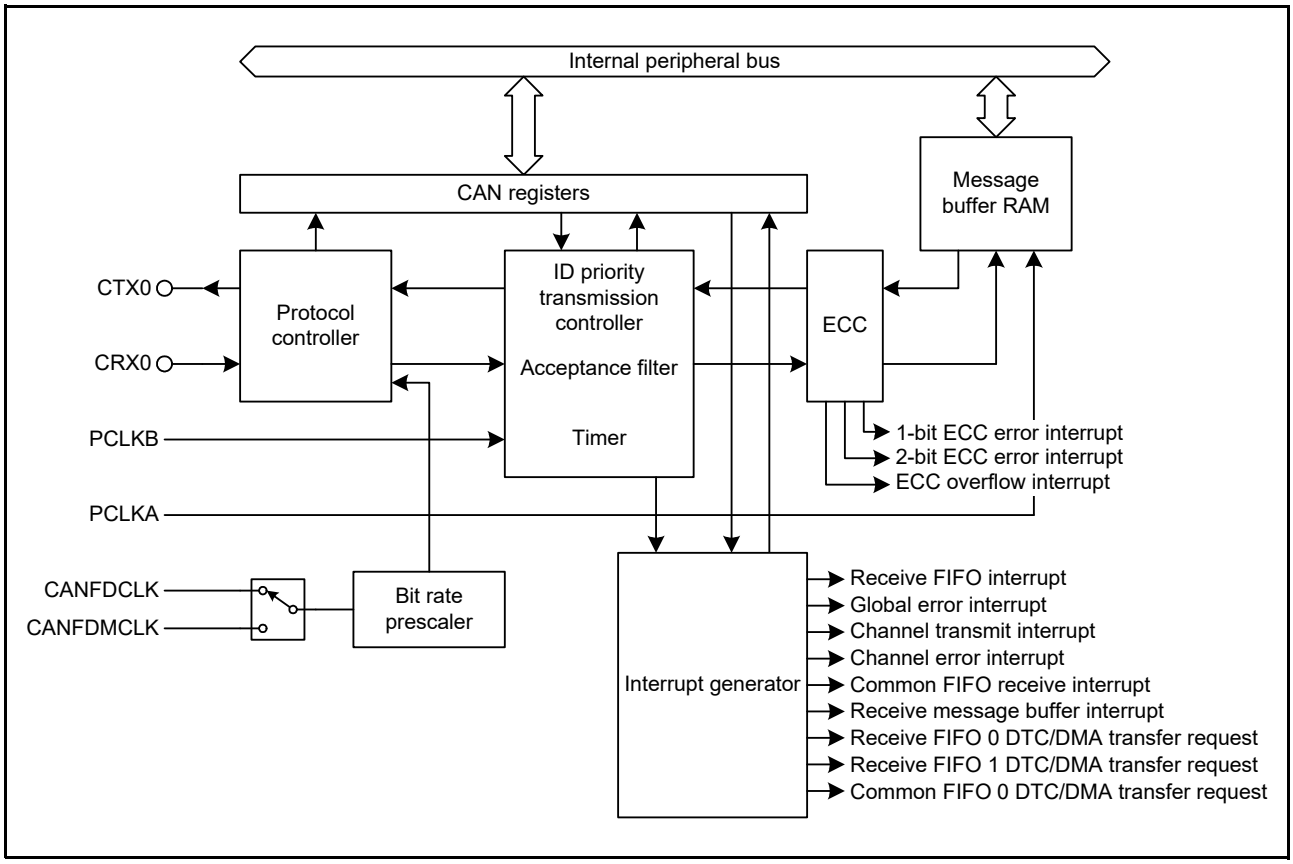


Figure 36.1 CAN FD Module Block Diagram

- CRX0, CTX0
  I/O pins of the CAN FD module
- Protocol controller
  Handles CAN FD protocol processing such as bus arbitration, bit timing at transmission and reception, stuffing, error handling, etc.
- Message buffer RAM
  Used for message buffers or FIFOs for transmitting and receiving messages. Each message has an individual identifier, data length code, a data field, a message pointer for upper layer application, and a timestamp.
- Acceptance filter
  Performs filtering of received messages. The entries set in the acceptance filter list are used for the filtering process.
- Timers
  Two timers: one used for the receive timestamp function and the other for adjusting the message transmission interval from the transmit FIFO.

Table 36.2 lists the pin configuration of the CAN FD module.

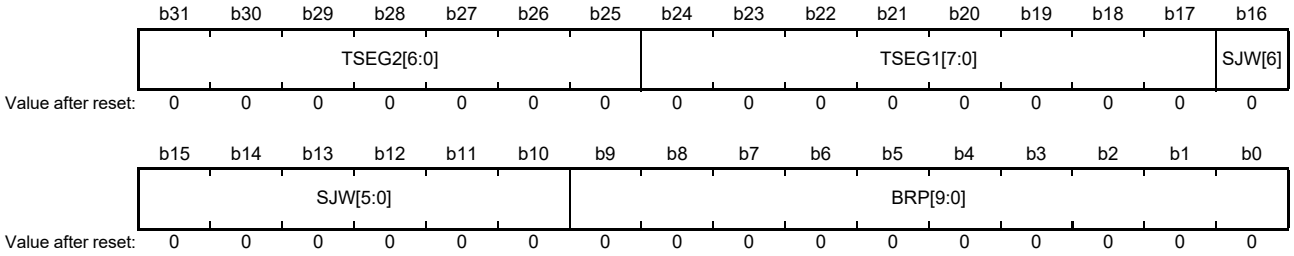
Table 36.2 CAN FD Module Pin Configuration

Pin Name	I/O	Function
CRX0	Input	Receive data input
CTX0	Output	Transmit data output

36.2 Register Descriptions

36.2.1 Nominal Bit Rate Configuration Register (NBCR)

Address(es): CANFD0.NBCR 000A 8000h



Bit	Symbol	Bit Name	Description	R/W
b9 to b0	BRP[9:0]	Bit Rate Prescaler Setting	000h: No division 001h: Divided by 2 : : 3FEh: Divided by 1023 3FFh: Divided by 1024	R/W
b16 to b10	SJW[6:0]	Resynchronization Jump Width Control	00h: 1 Tq 01h: 2 Tq : : 7Eh: 127 Tq 7Fh: 128 Tq	R/W
b24 to b17	TSEG1[7:0]	Time Segment 1 Control	00h: Setting prohibited 01h: 2 Tq 02h: 3 Tq 03h: 4 Tq : : FEh: 255 Tq FFh: 256 Tq	R/W
b31 to b25	TSEG2[6:0]	Time Segment 2 Control	00h: Setting prohibited 01h: 2 Tq 02h: 3 Tq : : 7Eh: 127 Tq 7Fh: 128 Tq	R/W

This register is used to set the nominal bit rate during transmission and reception.
The value cannot be changed in CH_OPERATION or CH_SLEEP mode. Rewrite this register in CH_RESET or CH_HALT mode.
For details on the setting values, refer to section 36.4.1.2, Bit Timing.

BRP[9:0] Bits (Bit Rate Prescaler Setting)

These bits are used to define a period of 1 Tq (Time Quantum) that is the basis for CAN communication. Set the division ratio for the operating clock of the data link layer (DLL clock) selected by the GCFG.DLLCS bit. If the set value is n, the bit rate prescaler divides the DLL clock by n + 1.

SJW[6:0] Bits (Resynchronization Jump Width Control)

These bits are used to specify the resynchronization jump width with a Tq value. A value from 1 to 128 Tq can be set. Set a value less than or equal to that of the TSEG2[6:0] bits.

TSEG1[7:0] Bits (Time Segment 1 Control)

These bits are used to specify the total value (TSEG1) of the propagation time segment (PROP_SEG) and phase buffer segment 1 (PHASE_SEG1) with a Tq value. A value from 2 to 256 Tq can be set.

TSEG2[6:0] Bits (Time Segment 2 Control)

These bits are used to specify the value (TSEG2) of the phase buffer segment 2 (PHASE_SEG2) with a Tq value. A value from 2 to 128 Tq can be set. Set a value less than that of the TSEG1[7:0] bits.

36.2.2 Channel Control Register (CHCR)

Address(es): CANFD0.CHCR 000A 8004h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ROME	BFT	—	—	—	CTMS[1:0]		CTME	EDM	BOM[1:0]		—	TDCVIE	SCOVIE	ECOVIE	TAIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ALIE	BLIE	OLIE	BORIE	BOEIE	EPIE	EWIE	BEIE	—	—	—	—	RTBO	SLPRQ	MDC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	MDC[1:0]	Channel Mode Control	b1 b0 0 0: Requests transition to CH_OPERATION mode 0 1: Requests transition to CH_RESET mode 1 0: Requests transition to CH_HALT mode 1 1: Keep current mode	R/W
b2	SLPRQ	CH_SLEEP Mode Request	0: Requests release from CH_SLEEP mode 1: Requests transition to CH_SLEEP mode	R/W
b3	RTBO	Forced Recovery from Bus-Off*1	0: Forced recovery from bus-off state is disabled. 1: Forced recovery from bus-off state is enabled.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	BEIE	Bus Error Interrupt Enable*2	0: Bus error interrupt is disabled. 1: Bus error interrupt is enabled.	R/W
b9	EWIE	Error Warning Interrupt Enable*2	0: Error warning interrupt is disabled. 1: Error warning interrupt is enabled.	R/W
b10	EPIE	Error Passive Interrupt Enable*2	0: Error passive interrupt is disabled. 1: Error passive interrupt is enabled.	R/W
b11	BOEIE	Bus-Off Entry Interrupt Enable*2	0: Bus-off entry interrupt is disabled. 1: Bus-off entry interrupt is enabled.	R/W
b12	BORIE	Bus-Off Recovery Interrupt Enable*2	0: Bus-off recovery interrupt is disabled. 1: Bus-off recovery interrupt is enabled.	R/W
b13	OLIE	Overload Interrupt Enable*2	0: Overload interrupt is disabled. 1: Overload interrupt is enabled.	R/W
b14	BLIE	Bus Lock Interrupt Enable*2	0: Bus lock interrupt is disabled. 1: Bus lock interrupt is enabled.	R/W
b15	ALIE	Arbitration Lost Interrupt Enable*2	0: Arbitration lost interrupt is disabled. 1: Arbitration lost interrupt is enabled.	R/W
b16	TAIE	Transmission Abort Interrupt Enable*2	0: Transmission abort interrupt is disabled. 1: Transmission abort interrupt is enabled.	R/W
b17	ECOVIE	Error Occurrence Counter Overflow Interrupt Enable*2	0: Error occurrence counter overflow interrupt is disabled. 1: Error occurrence counter overflow interrupt is enabled.	R/W
b18	SCOVIE	Success Occurrence Counter Overflow Interrupt Enable*2	0: Success occurrence counter overflow interrupt is disabled. 1: Success occurrence counter overflow interrupt is enabled.	R/W
b19	TDCVIE	Transceiver Delay Compensation Violation Interrupt Enable*2, *3	0: Transceiver delay compensation violation interrupt is disabled. 1: Transceiver delay compensation violation interrupt is enabled.	R/W
b20	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit Name	Description	R/W
b22, b21	BOM[1:0]	Bus-Off Recovery Mode Select*2	b22 b21 0 0: Normal mode (ISO 11898-1 compliant) 0 1: Automatically enters CH_HALT mode at bus-off entry 1 0: Automatically enters CH_HALT mode at bus-off end 1 1: Enters CH_HALT mode by software (during bus-off recovery)	R/W
b23	EDM	Error Display Mode Select*4	0: Only the first error detected is indicated. 1: All detected errors are indicated.	R/W
b24	CTME	Channel Test Mode Enable*5	0: Channel test mode is disabled. 1: Channel test mode is enabled.	R/W
b26, b25	CTMS[1:0]	Channel Test Mode Select*5	b26 b25 0 0: Basic test mode 0 1: Listen-only mode 1 0: Self-test mode 0 (external loop back mode) 1 1: Self-test mode 1 (internal loop back mode)	R/W
b29 to b27	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b30	BFT	Bit Flip Test*5	0: First bit of received data stream is not inverted. 1: First bit of received data stream is inverted.	R/W
b31	ROME	Restricted Operation Mode Enable *3, *5	0: Limited operation mode is disabled. 1: Limited operation mode is enabled.	R/W

Note 1. Set this bit in CH_OPERATION mode.
Note 2. Rewrite these bits in CH_RESET mode.
Note 3. Do not set this bit to 1 in Classic Only mode.
Note 4. Set this bit in CH_RESET or CH_HALT mode.
Note 5. Set these bits in CH_HALT mode.

This register controls the modes of the channel. It is also used to enable interrupt generation when an error is detected on the CAN bus and to set the test mode.
In CH_SLEEP mode, values other than the SLPRQ bit cannot be changed.

MDC[1:0] Bits (Channel Mode Control)

The MDC[1:0] bits are used to specify the mode of the CAN channel. For the CAN mode transition, refer to section 36.3.2, Channel Modes.
When the CANFD module is in GL_HALT mode, these bits can only be set to 10b (CH_HALT mode) or 01b (CH_RESET mode).
These bits automatically become 10b when the mode is changed to CH_HALT mode by the setting of the CHCR.BOM[1:0] bits.
When writing to these bits and transition to CH_HALT mode (at the entry to bus-off state when the CHCR.BOM[1:0] bits are 01b, and at the end of bus-off state when the CHCR.BOM[1:0] bits are 10b) occur at the same time, priority is given to writing from the CPU. The value of these bits are automatically updated only if the above event occurs when these bits are 00b (CH_OPERATION mode).

SLPRQ Bit (CH_SLEEP Mode Request)

This bit is used to control the transition to CH_SLEEP mode and the return from CH_SLEEP mode.
Setting this bit to 1 in CH_RESET mode requests the channel to transition to CH_SLEEP mode. Setting this bit to 0 in CH_SLEEP mode requests the channel to transition to CH_RESET mode. These bits cannot be changed in other modes.

RTBO Bit (Forced Recovery from Bus-Off)

This bit is used to forcibly recover from the bus-off state. Use this bit only when the CHCR.BOM[1:0] bits are 00b.
This bit is automatically set to 0. The read value is always 0.
When this bit is set to 1 during bus-off state, the channel transitions from bus-off state to integrating state within 1 bit time. Also the CHSR.REC[7:0] and TEC[7:0] bits are set to 00h and the CHSR.BOST flag is set to 0. Other registers and

bits do not change. In this case, even if the bus-off recovery interrupt is enabled, the bus-off recovery interrupt does not generated.
If this bit is set to 1 except in the bus-off state, nothing occurs.

BEIE Bit (Bus Error Interrupt Enable)

When the CHESR.BEDF flag is set to 1 while this bit is 1, a channel error interrupt request is generated.

EWIE Bit (Error Warning Interrupt Enable)

When the CHESR.EWDF flag is set to 1 while this bit is 1, a channel error interrupt request is generated.

EPIE Bit (Error Passive Interrupt Enable)

When the CHESR.EPDF flag is set to 1 while this bit is 1, a channel error interrupt request is generated.

BOEIE Bit (Bus-Off Entry Interrupt Enable)

When the CHESR.BOEDF flag is set to 1 while this bit is 1, a channel error interrupt request is generated.

BORIE Bit (Bus-Off Recovery Interrupt Enable)

When the CHESR.BORDF flag is set to 1 while this bit is 1, a channel error interrupt request is generated.

OLIE Bit (Overload Interrupt Enable)

When the CHESR.OLDF flag is set to 1 while this bit is 1, a channel error interrupt request is generated.

BLIE Bit (Bus Lock Interrupt Enable)

When the CHESR.BLDF flag is set to 1 while this bit is 1, a channel error interrupt request is generated.

ALIE Bit (Arbitration Lost Interrupt Enable)

When the CHESR.ALDF flag is set to 1 while this bit is 1, a channel error interrupt request is generated.

TAIE Bit (Transmission Abort Interrupt Enable)

When the transmission from the transmit message buffer of the channel is successfully aborted while this bit is 1, a channel transmit interrupt request is generated.

ECOVIE Bit (Error Occurrence Counter Overflow Interrupt Enable)

When the FDSTS.ECOV flag is set to 1 while this bit is 1, a channel error interrupt request is generated.

SCOVIE Bit (Success Occurrence Counter Overflow Interrupt Enable)

When the FDSTS.SCOV flag is set to 1 while this bit is 1, a channel error interrupt request is generated.

TDCVIE Bit (Transceiver Delay Compensation Violation Interrupt Enable)

When the FDSTS.TDCFVF flag is set to 1 while this bit is 1, a channel error interrupt request is generated. Do not set this bit to 1 in Classic only mode.

BOM[1:0] Bits (Bus-Off Recovery Mode Select)

These bits control the recovery timing from the bus-off state.

EDM Bit (Error Display Mode Select)

This bit controls the indication mode of the error flags (b14 to b8) in the CHESR register.
When this bit is 0, only the flag corresponding to the first error detected becomes 1. If multiple errors are detected at the

same time, all corresponding flags become 1. The other flags do not become 1 until all the flags in b14 to b8 have been cleared.
When this bit is 1, the error flags are updated each time an error is detected.

CTME Bit (Channel Test Mode Enable)

This bit is used to enable the channel test mode.
When the channel transitions to CH_RESET mode, this bit becomes 0.

CTMS[1:0] Bits (Channel Test Mode Select)

These bits are used to select the test mode.
When the channel transitions to CH_RESET mode, these bits become 00b.

BFT Bit (Bit Flip Test)

This bit is used to check the CRC generator in the protocol controller.
Setting this bit to 1 inverts the first bit of the message data stream (ID bit) so that the result of the internally generated CRC does not match the received CRC value.
Note that as a result of bit inversion, a stuff error may be detected instead of a CRC error. Refer to the bit stuffing rule when using this function.
The CRC value generated internally can be checked in the following bits.

- The CHESR.CRC15[14:0] bits (for Classical CAN frames)
- The FDCRC.CRC21[20:0] bits (for CAN FD frames)

When using the BFT bit, it is required for other CAN nodes to send the reference message.

Note: Since the transmit and receive modes share the same CRC generator, there is no need to test individually CRC errors in transmit mode.

Bit flip test mode is enabled when both the BFT and CTME bits are 1 and the CTMS[1:0] bits are 00b (basic test mode).
Using this function on the transmit node causes a bit error or arbitration lost.
When the channel transitions to CH_RESET mode, this bit becomes 0.

ROME Bit (Restricted Operation Mode Enable)

When both this bit and the CTME bit are 1, the limited operation mode is enabled. Use this mode only in basic test mode (CTMS[1:0] bits = 00b). Also, do not set this bit to 1 in Classic only mode.
When the channel transitions to CH_RESET mode, this bit becomes 0.

36.2.3 Channel Status Register (CHSR)

Address(es): CANFD0.CHSR 000A 8008h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TEC[7:0]								REC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	RESI	CRDY	RECST	TRMST	BOST	EPST	SLPST	HLTST	RSTST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	RSTST	CH_RESET Status Flag	0: Not in CH_RESET mode 1: In CH_RESET mode	R
b1	HLTST	CH_HALT Status Flag	0: Not in CH_HALT mode 1: In CH_HALT mode	R
b2	SLPST	CH_SLEEP Status Flag	0: Not in CH_SLEEP mode 1: In CH_SLEEP mode	R
b3	EPST	Error Passive Status Flag	0: Not in error passive state 1: In error passive state	R
b4	BOST	Bus-Off Status Flag	0: Not in bus-off state 1: In bus-off state	R
b5	TRMST	Transmit Status Flag	0: Channel is not transmitting 1: Transmission in progress	R
b6	RECST	Receive Status Flag	0: Channel is not receiving 1: Reception in progress	R
b7	CRDY	Communication Ready Flag	0: Channel is not ready for communication 1: Channel is ready for communication	R
b8	RESI	Receive ESI Flag*1	0: No message with ESI flag set to 1 was received. 1: At least 1 message with ESI flag set to 1 was received.	R/(W) *2
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23 to b16	REC[7:0]	Reception Error Count	These bits increment or decrement the counter value according to error status of the CAN channel during reception.	R
b31 to b24	TEC[7:0]	Transmission Error Count	These bits increment or decrement the counter value according to error status of the CAN channel during transmission.	R

Note 1. This flag can be set to 0 only in CH_OPERATION or CH_HALT mode.
Note 2. Writing 1 has no effect. This flag is cleared by writing 0 to it.

This register shows the mode, error and transmission or reception status of the CAN channel together with its reception and transmission error count values.

RSTST Flag (CH_RESET Status Flag)

The RSTST flag indicates whether the CAN channel is in CH_RESET mode.
This flag is automatically set to 1 when the CAN channel enters CH_RESET mode, and is automatically set to 0 when the CAN channel exits CH_RESET mode. When the mode is changed from CH_RESET mode to CH_SLEEP mode, the RSTST flag remains 1.

HLTST Flag (CH_HALT Status Flag)

The HLTST flag indicates whether the CAN channel is in CH_HALT mode.

This flag is automatically set to 1 when the CAN channel enters CH_HALT mode, and is automatically set to 0 when the CAN channel exits CH_HALT mode.

SLPST Flag (CH_SLEEP Status Flag)

The SLPST flag indicates whether the CAN channel is in CH_SLEEP mode.
This flag is automatically set to 1 when the CAN channel enters CH_SLEEP mode, and is automatically set to 0 when the CAN channel exits CH_SLEEP mode.

EPST Flag (Error Passive Status Flag)

The EPST flag indicates whether the CAN channel has entered the error passive state.
This flag is automatically set to 1 when the value of the REC[7:0] or TEC[7:0] bits exceeds 127.
This flag is automatically set to 0 when the CAN channel exits the error passive state or enters CH_RESET mode.

BOST Flag (Bus-Off Status Flag)

The BOST flag indicates whether the CAN channel has entered the bus-off state.
This flag is automatically set to 1 when the value of the TEC[7:0] bits exceeds 255 and the CAN channel is in the bus-off state.
This flag is automatically set to 0 when the CAN channel exits bus-off state.

TRMST Flag (Transmit Status Flag)

The TRMST flag indicates whether the CAN channel is transmitting a message.
This flag is automatically set to 1 when the CAN channel is operating as a transmitter node or is in the bus-off state, and is automatically set to 0 when the CAN channel is in the idle state or starts operating as a receiver node.

RECST Flag (Receive Status Flag)

The RECST flag indicates whether the CAN channel is receiving a message.
This flag is automatically set to 1 when the CAN channel is operating as a receiver node, and is automatically set to 0 when the CAN channel is in the idle state or starts operating as a transmitter node.

CRDY Flag (Communication Ready Flag)

The CRDY flag indicates whether the CAN channel is ready for communication.
This flag is automatically set to 1 when the CAN channel is ready to perform communication following the detection of 11 consecutive recessive bits after exiting CH_RESET or CH_HALT mode.
This flag is automatically set to 0 when the CAN channel is in CH_RESET or CH_HALT mode.

Note: This flag is 1 in the bus-off state.

RESI Flag (Receive ESI Flag)

The RESI flag is set to 1 when the ESI bit is sampled recessively for a received message without any error. When in Loopback or Mirror mode, the self-transmitted messages are considered reception messages.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
This flag is cleared by writing 0 to it. This flag is automatically set to 0 when the CAN channel is in CH_RESET mode.
Do not use the bit clear instruction to clear this flag. Set only this flag to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.

REC[7:0] Bits (Reception Error Count)

The REC[7:0] bits indicate the value of the reception error counter.
The value in bus-off state is indeterminate.
These bits are automatically set to 00h when the CANFD module enters GL_RESET or the CAN channel is in

CH_RESET mode.

TEC[7:0] Bits (Transmission Error Count)

The TEC[7:0] bits indicate the value of the transmission error counter.

These bits are automatically set to 00h when CANFD module is in GL_RESET or CAN channel is in CH_RESET mode.

36.2.4 Channel Error Status Register (CHESR)

Address(es): CANFD0.CHESR 000A 800Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	CRC15[14:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	ADEDF	B0EDF	B1EDF	CEDF	AEDF	FEDF	SEDF	ALDF	BLDF	OLDF	BORDF	BOEDF	EPDF	EWDF	BEDF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BEDF	Bus Error Detect Flag*1	0: Bus error is not detected 1: Bus error is detected	R/(W) *2
b1	EWDF	Error Warning Detect Flag*1	0: Error warning is not detected 1: Error warning is detected	R/(W) *2
b2	EPDF	Error Passive Detect Flag*1	0: Error passive is not detected 1: Error passive is detected	R/(W) *2
b3	BOEDF	Bus-Off Entry Detect Flag*1	0: Bus-off entry is not detected 1: Bus-off entry is detected	R/(W) *2
b4	BORDF	Bus-Off Recovery Detect Flag*1	0: Bus-off recovery is not detected 1: Bus-off recovery is detected	R/(W) *2
b5	OLDF	Overload Detect Flag*1	0: Overload is not detected 1: Overload is detected	R/(W) *2
b6	BLDF	Bus Lock Detect Flag*1	0: Bus lock is not detected 1: Bus lock is detected	R/(W) *2
b7	ALDF	Arbitration Lost Detect Flag*1	0: Arbitration lost is not detected 1: Arbitration lost is detected	R/(W) *2
b8	SEDF	Stuff Error Detect Flag*1	0: Stuff error is not detected 1: Stuff error is detected	R/(W) *2
b9	FEDF	Form Error Detect Flag*1	0: Form error is not detected 1: Form error is detected	R/(W) *2
b10	AEDF	Acknowledge Error Detect Flag*1	0: Acknowledge error is not detected 1: Acknowledge error is detected	R/(W) *2
b11	CEDF	CRC Error Detect Flag*1	0: CRC error is not detected 1: CRC error is detected	R/(W) *2
b12	B1EDF	Bit 1 Error Detect Flag*1	0: Bit 1 error is not detected 1: Bit 1 error is detected	R/(W) *2
b13	B0EDF	Bit 0 Error Detect Flag*1	0: Bit 0 error is not detected 1: Bit 0 error is detected	R/(W) *2
b14	ADEDF	ACK Delimiter Error Detect Flag*1	0: Acknowledge delimiter error is not detected 1: Acknowledge delimiter error is detected	R/(W) *2
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b30-b16	CRC15[14:0]	CRC_15 Test	These bits show the CRC_15 value calculated for the CAN2.0 CAN frame.	R
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. This flag can be set to 0 only in CH_OPERATION or CH_HALT mode.
Note 2. Writing 1 has no effect. This flag is cleared by writing 0 to it.

This register shows the status of various error conditions detectable regardless of the interrupt enable/disable setting of the Channel Control Register (CHCR). It also shows the status of the various bus errors detectable by the CAN channel. Refer to the CAN specification (ISO 11898-1) for the conditions under which each error occurs.

Only a single bit can be cleared at a time. Do not use the bit clear instruction to clear the flag. Set only the flag to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.

BEDF Flag (Bus Error Detect Flag)

The BEDF flag indicates a detection of an error state, flagged by b14 to b8 in this register.
This flag is automatically set to 1 when a bus error is detected, and is automatically set to 0 when the CAN channel is in CH_RESET mode.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.

EWDF Flag (Error Warning Detect Flag)

The EWDF flag indicates whether an error warning condition has been detected for the CAN channel.
This flag is automatically set to 1 when the value of either the CHSR.TEC[7:0] or REC[7:0] bits exceeds 95.
The setting of this flag only occurs when the value of the TEC[7:0] or REC[7:0] bits initially exceeds 95. Therefore, if the TEC[7:0] or REC[7:0] bits remains > 95 and the EWDF flag is cleared by software, it is not set to 1 again until the value of both the TEC[7:0] and REC[7:0] bits go below 96 and either TEC[7:0] or REC[7:0] bits crosses over again from a value ≤ 95 to a value > 95.
If a set condition occurs simultaneously with a clear condition, then the flag is set to 1. It is automatically set to 0 when the CAN channel is in CH_RESET mode.

EPDF Flag (Error Passive Detect Flag)

The EPDF flag indicates a detection of a CAN channel error passive state.
This flag is automatically set to 1 when the CAN error state becomes error passive state.
The setting of this flag only occurs when the value of either the CHSR.TEC[7:0] or REC[7:0] bits initially exceeds 127. Therefore, if the value of either the TEC[7:0] or REC[7:0] bits remains > 127 and the flag is cleared by software, it is not set to 1 again until the value of both the TEC[7:0] and REC[7:0] bits go below 128 and either TEC[7:0] or REC[7:0] bits crosses over again from a value ≤ 127 to a value > 127.
If a set condition occurs simultaneously with a clear condition, then the flag is set to 1. It is automatically set to 0 when the CAN channel is in CH_RESET mode.

BOEDF Flag (Bus-Off Entry Detect Flag)

The BOEDF flag indicates a detection of a CAN channel bus-off entry state.
This flag is automatically set to 1 when the CAN error state enters the bus-off state.
This flag is automatically set to 0 when the CAN channel is in CH_RESET mode. If a set condition occurs simultaneously with a clear condition, then the flag is set to 1.

BORDF Flag (Bus-Off Recovery Detect Flag)

The BORDF flag indicates a detection of a CAN channel bus-off recovery state.
This flag is automatically set to 1 if CAN channel recovers from bus-off state in the following conditions:

- When the CHCR.BOM[1:0] bits are 00b and normal recovery (128 occurrence of 11 consecutive recessive bits are detected) occurs
- When the CHCR.BOM[1:0] bits are 10b and normal recovery (128 occurrence of 11 consecutive recessive bits are detected) occurs
- When the CHCR.BOM[1:0] bits are 11b and normal recovery (128 occurrence of 11 consecutive recessive bits are detected) occurs.

The flag is not set to 1 if CAN channel recovers from bus-off state in the following conditions:

- When the CH_RESET mode is requested
- When setting the CHCR.RTBO bit to 1 (the CAN channel returns to error active)
- When the CHCR.BOM[1:0] bits are 01b

- When the CHCR.BOM[1:0] bits are 11b and a halt request is asserted before the CAN channel reaches the end of the bus-off state.

This flag is automatically set to 0 when the CAN channel is in CH_RESET mode. If a set condition occurs simultaneously with a clear condition, the flag is set to 1.

OLDF Flag (Overload Detect Flag)

The OLDF flag indicates a detection of a CAN channel overload state.
This flag is automatically set to 1 when an overload condition is detected. If a set condition occurs simultaneously with a clear condition, then the flag is set to 1.
This flag is automatically set to 0 when the CAN channel is in CH_RESET mode.

BLDF Flag (Bus Lock Detect Flag)

The BLDF flag indicates a detection of a CAN channel bus lock condition.
This flag is automatically set to 1 when 32 consecutive dominant bits are detected on the CAN bus while the CAN channel is in Operation mode.
If a set condition occurs simultaneously with a clear condition, then the flag is set to 1. This flag is automatically set to 0 when the CAN channel is in CH_RESET mode.

ALDF Flag (Arbitration Lost Detect Flag)

The ALDF flag indicates a detection of a CAN channel bus arbitration lost condition.
The flag is automatically set to 1 when an arbitration lost condition is detected on the CAN bus while the CAN channel is in Operation mode.
If a set condition occurs simultaneously with a clear condition, then the flag is set to 1. This flag is automatically set to 0 when the CAN channel is in CH_RESET mode.

SEDF Flag (Stuff Error Detect Flag)

The SEDF flag indicates a detection of a CAN stuff error.
After clearing this flag, confirm that the flag is actually changed to 0 and execute subsequent instructions.
This flag is automatically set to 1 when a stuff error is detected. If the CHCR.EDM bit is 1 and if a set condition occurs simultaneously with a clear condition, then the flag is set to 1.
This flag is automatically set to 0 when the CAN channel is in CH_RESET mode. If the CHCR.EDM bit is 0 and if a set condition occurs simultaneously with a clear condition, then this flag is set to 0 if even one of the error flags (b14 to b8) in the CHESR register is already 1. Otherwise, this flag is set to 1 if the setting of the error flags (b14 to b8) in the CHESR register is 0000000b.

FEDF Flag (Form Error Detect Flag)

The FEDF flag indicates a detection of a CAN form error.
After clearing this flag, confirm that the flag is actually changed to 0 and execute subsequent instructions.
This flag is automatically set to 1 when a form error is detected. If the CHCR.EDM bit is 1 and if a set condition occurs simultaneously with a clear condition, then the flag is set to 1.
This flag is automatically set to 0 when the CAN channel is in CH_RESET mode. If the CHCR.EDM bit is 0 and if a set condition occurs simultaneously with a clear condition, then this flag is set to 0 if even one of the error flags (b14 to b8) in the CHESR register is already 1. Otherwise, this flag is set to 1 if the setting of the error flags (b14 to b8) in the CHESR register is 0000000b.

AEDF Flag (Acknowledge Error Detect Flag)

The AEDF flag indicates a detection of a CAN acknowledge error.
After clearing this flag, confirm that the flag is actually changed to 0 and execute subsequent instructions.

This flag is automatically set to 1 when an acknowledge error is detected. If the CHCR.EDM bit is 1 and if a set condition occurs simultaneously with a clear condition, then the flag is set to 1.
This flag is automatically set to 0 when the CAN channel is in CH_RESET mode. If the CHCR.EDM bit is 0 and if a set condition occurs simultaneously with a clear condition, then this flag is set to 0 if even one of the error flags (b14 to b8) in the CHESR register is already 1. Otherwise, this flag is set to 1 if the setting of the error flags (b14 to b8) in the CHESR register is 0000000b.

CEDF Flag (CRC Error Detect Flag)

The CEDF flag indicates a detection of a CAN CRC error.
After clearing this flag, confirm that the flag is actually changed to 0 and execute subsequent instructions.
This flag is automatically set to 1 when a CRC error is detected. If the CHCR.EDM bit is 1 and if a set condition occurs simultaneously with a clear condition, then the flag is set to 1.
This flag is automatically set to 0 when the CAN channel is in CH_RESET mode. If the CHCR.EDM bit is 0 and if a set condition occurs simultaneously with a clear condition, then this flag is set to 0 if even one of the error flags (b14 to b8) in the CHESR register is already 1. Otherwise, this flag is set to 1 if the setting of the error flags (b14 to b8) in the CHESR register is 0000000b.

B1EDF Flag (Bit 1 Error Detect Flag)

The B1EDF flag indicates a detection of a recessive bit error.
After clearing this flag, confirm that the flag is actually changed to 0 and execute subsequent instructions.
This flag is automatically set to 1 when a recessive bit error (expected recessive bit, sampled as dominant bit) is detected. If the CHCR.EDM bit is 1 and if a set condition occurs simultaneously with a clear condition, then the flag is set to 1.
This flag is automatically set to 0 when the CAN channel is in CH_RESET mode. If the CHCR.EDM bit is 0 and if a set condition occurs simultaneously with a clear condition, then this flag is set to 0 if even one of the error flags (b14 to b8) in the CHESR register is already 1. Otherwise, this flag is set to 1 if the setting of the error flags (b14 to b8) in the CHESR register is 0000000b.

B0EDF Flag (Bit 0 Error Detect Flag)

The B0EDF flag indicates a detection of a dominant bit error.
After clearing this flag, confirm that the flag is actually changed to 0 and execute subsequent instructions.
This flag is automatically set to 1 when a dominant bit error (expected dominant bit, sampled as recessive bit) is detected. If the CHCR.EDM bit is 1 and if a set condition occurs simultaneously with a clear condition, then the flag is set to 1.
This flag is automatically set to 0 when the CAN channel is in CH_RESET mode. If the CHCR.EDM bit is 0 and if a set condition occurs simultaneously with a clear condition, then this flag is set to 0 if even one of the error flags (b14 to b8) in the CHESR register is already 1. Otherwise, this flag is set to 1 if the setting of the error flags (b14 to b8) in the CHESR register is 0000000b.

ADEDF Flag (ACK Delimiter Error Detect Flag)

The ADEDF flag indicates a detection of an acknowledge delimiter bit error.
After clearing this flag, confirm that the flag is actually changed to 0 and execute subsequent instructions.
This flag is automatically set to 1 when a form error is detected during the acknowledge delimiter state of frame transmission. If the CHCR.EDM bit is 1 and if a set condition occurs simultaneously with a clear condition, then the flag is set to 1.
This flag is automatically set to 0 when the CAN channel is in CH_RESET mode. If the CHCR.EDM bit is 0 and if a set condition occurs simultaneously with a clear condition, then this flag is set to 0 if even one of the error flags (b14 to b8) in the CHESR register is already 1. Otherwise, this flag is set to 1 if the setting of the error flags (b14 to b8) in the CHESR register is 0000000b.

CRC15[14:0] Bits (CRC_15 Test)

The CRC15[14:0] bits indicate the calculated CRC_15 value when the CHCR.CTME bit is 1 (channel test mode enabled). If the CHCR.CTME bit is 0, then these bits are always read as 0000h.
The CRC_15 value which is read from these bits show the CAN2.0 CRC value calculated by the CAN channel logic.
The value of the CRC15[14:0] bits is updated in the first bit of the CRC field of the Classical CAN frame.
These bits are automatically set to 0000h when the CAN channel is in CH_RESET mode.

36.2.5 Data Bit Rate Configuration Register (DBCR)

Address(es): CANFD0.DBCR 000A 8100h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	SJW[3:0]				—	—	—	—	TSEG2[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	TSEG1[4:0]					BRP[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	BRP[7:0]	Bit Rate Prescaler Setting	00h: No division 01h: Divided by 2 : : FEh: Divided by 255 FFh: Divided by 256	R/W
b12 to b8	TSEG1[4:0]	Time Segment 1 Control	00h: Setting prohibited 01h: 2 Tq 02h: 3 Tq 03h: 4 Tq : : 1Eh: 31 Tq 1Fh: 32 Tq	R/W
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b19 to b16	TSEG2[3:0]	Time Segment 2 Control	0h: Setting prohibited 1h: 2 Tq : : Eh: 15 Tq Fh: 16 Tq	R/W
b23 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b27 to b24	SJW[3:0]	Resynchronization Jump Width Control	0h: 1 Tq 1h: 2 Tq : : Fh: 16 Tq	R/W
b31 to b28	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set this register in CH_RESET or CH_HALT mode.

This register configures the transmission/reception data bit rate parameters of the channel.
In Classic only mode, there is no need to configure this register.

BRP[7:0] Bits (Bit Rate Prescaler Setting)

The BRP[7:0] bits define the number of the DLL clock contained in 1 Tq (Time Quantum).

TSEG1[4:0] Bits (Time Segment 1 Control)

The TSEG1[4:0] bits set the segment TSEG1 to compensate for edges on the CAN bus with a positive phase error. A value from 2 to 32 Tq can be set.
The TSEG1[4:0] bits are also used to set the propagation time segment.
Do not write any other value to these bits. Refer to section 36.4.1.2, Bit Timing for more details.

TSEG2[3:0] Bits (Time Segment 2 Control)

The TSEG2[3:0] bits set the segment TSEG2 to compensate for edges on the CAN bus with a negative phase error. A

value from 2 to 16 Tq can be set.
Do not write any other value to these bits.

SJW[3:0] Bits (Resynchronization Jump Width Control)

The SJW[3:0] bits set the resynchronization jump width. A value from 1 to 16 Tq can be set.

36.2.6 CAN FD Configuration Register (FDCFG)

Address(es): CANFD0.FDCFG 000A 8104h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	CLOE	REFE	FDOE	—	—	—	—	TDCO[7:0]							
Value after reset:	0	0/1*1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TESI	TDCE	SSPC	—	—	—	—	—	ECC[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	ECC[2:0]	Error Occurrence Counter Configuration*2	b2 b0 0 0 0: All CAN transmitter or receiver frames 0 0 1: All CAN transmitter frames 0 1 0: All CAN receiver frames 0 1 1: Setting prohibited 1 0 0: Only transmitter or receiver CAN FD data-phase 1 0 1: Only transmitter CAN FD data-phase 1 1 0: Only receiver CAN FD data-phase 1 1 1: Setting prohibited	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	SSPC	Secondary Sample Point Configuration*2	0: Measured + offset 1: Offset-only	R/W
b9	TDCE	Transceiver Delay Compensation Enable*2	0: Transceiver delay compensation is disabled 1: Transceiver delay compensation is enabled	R/W
b10	TESI	Transmit ESI Configuration*2	0: The ESI flag in the transmission frame reflects the error status of the node itself. 1: The ESI flag in the transmission frame reflects the ESI bit in the message buffer if the node is not error passive, and the error status of the node itself If the node is error passive.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23 to b16	TDCO[7:0]	Transceiver Delay Compensation Offset*2	Sets the offset value for the transceiver delay compensation	R/W
b27 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b28	FDOE	FD Only Mode Enable*3	0: FD only mode is disabled 1: FD only mode is enabled	R/W
b29	REFE	Receive Edge Filter Enable*3	0: Reception edge filter is disabled 1: Reception edge filter is enabled	R/W
b30	CLOE	Classic Only Mode Enable*3, *4	0: Classic only mode is disabled 1: Classic only mode is enabled	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. The value after reset is 0 for products that support the CAN FD protocol, and 1 for products that support only CAN 2.0 protocol.
Note 2. Set these bits in CH_RESET or CH_HALT mode.
Note 3. Set these bits in CH_RESET mode.
Note 4. This bit can only be written for products that support the CAN FD protocol. For products that support only CAN 2.0 protocol, this bit is reserved and fixed to 1.

This register configures which communication direction (transmitter/receiver) errors are counted.

ECC[2:0] Bits (Error Occurrence Counter Configuration)

The ECC[2:0] bits select which type of CAN frame the protocol errors should be counted for.

SSPC Bit (Secondary Sample Point Configuration)

The SSPC bit selects which offset is used when defining the position of the secondary sample point (SSP) for the CAN channel. If the bit is 0, the position of the SSP is the measured transceiver delay plus the fixed offset. If the bit is 1, the position of the SSP is defined only by the offset.
Do not set this bit to 1 in Classic only mode.

TDCE Bit (Transceiver Delay Compensation Enable)

The TDCE bit enables the transceiver delay compensation for the CAN channel.
Do not set this bit to 1 in Classic only mode.

TESI Bit (Transmit ESI Configuration)

The TESI bit selects whether to reflect the error status of the node itself or the value of the ESI bit in the message buffer (CFB0.HF2.ESI bit or TMBn.HF2.ESI bit) in the ESI flag of the transmission message.
Do not set this bit to 1 in Classic only mode.

TDCO[7:0] Bits (Transceiver Delay Compensation Offset)

The TDCO[7:0] bits set the offset of the secondary sample point. How this value is used, depends on the SSPC setting. If the SSPC bit is 0, the transceiver delay compensation result is equal to the Trv_Delay (measured delay) + the value in the TDCO[7:0] bits, rounded down to the nearest integer number of Tq. Otherwise, the result is equal to the value in the TDCO[7:0] bits. Refer to section 36.4.1.5, Transceiver Delay Compensation for details.
The actual offset value is interpreted as TDCO[7:0] + 1. For example, if 4 is set in TDCO[7:0], the offset is 5 clock cycles. Clock cycle is 1 cycle of CAN channel DLL clock.
Do not set to these bits in Classic only mode.

FDOE Bit (FD Only Mode Enable)

The FDOE bit enables the transmission and reception of CAN FD frames only. If enabled, communication in Classical CAN frame format is disabled. The value of the FDF bit in the message buffer (CFB0.HF2.FDF bit or TMBn.HF2.FDF bit) is arbitrary because transmission of Classical CAN frames is not possible.
If messages with Classical CAN frame format are received, the protocol controller treats them as invalid frames and responds with error frames. If a Classical CAN frame is configured for transmission, the FDF bit is transmitted recessive, therefore a CAN FD frame is transmitted. If the data length code (DLC) is configured to be 9 bytes or more, the remaining data bytes are padded with CCh.
Do not set the FDOE and CLOE bits to 1 simultaneously.

REFE Bit (Receive Edge Filter Enable)

The REFE bit enables the reception edge filter during the integrating state. When this bit is 1, two or more consecutive dominant Tq are required to detect an edge for hard synchronization.
Do not set this bit to 1 in Classic only mode.

CLOE Bit (Classic Only Mode Enable)

The CLOE bit enables the Classic only mode. If this bit is set to 1, the protocol controller can only transmit Classical CAN frames and responds to CAN FD frames with a form error or CRC error.
Do not set the CLOE and FDOE bits to 1 simultaneously.

Table 36.3 Operation Mode Configuration

CLOE bit	FDOE bit	Operation mode
0	0	CAN FD mode
0	1	FD only mode
1	0	Classic only mode
1	1	Setting prohibited

36.2.7 CAN FD Control Register (FDCTR)

Address(es): CANFD0.FDCTR 000A 8108h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SCCL	ECCL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ECCL	Error Occurrence Counter Clear	When 1 is written to this bit, the error occurrence counter is cleared. This bit is read as 0.	R/W
b1	SCCL	Success Occurrence Counter Clear	When 1 is written to this bit, the success occurrence counter is cleared. This bit is read as 0.	R/W
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register controls the error and success occurrence counters.

ECCL Bit (Error Occurrence Counter Clear)

The ECCL bit is used to clear the error occurrence counter.
This bit cannot be written in CH_SLEEP or CH_RESET mode.
This bit is automatically set to 0 and is also set to 0 when the CAN channel is in CH_RESET mode.

SCCL Bit (Success Occurrence Counter Clear)

The SCCL bit is used to clear the success occurrence counter.
This bit cannot be written in CH_SLEEP or CH_RESET mode.
This bit is automatically set to 0 and is also set to 0 when the CAN channel is in CH_RESET mode.

36.2.8 CAN FD Status Register (FDSTS)

Address(es): CANFD0.FDSTS 000A 810Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SC[7:0]								EC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TDCV	—	—	—	—	—	SCOV	ECOV	TDCR[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	TDCR[7:0]	Transceiver Delay Compensation Result	Indicates the transceiver delay compensation result when the transceiver delay has been measured	R
b8	ECOV	Error Occurrence Counter Overflow Flag*1	0: Error occurrence counter has not overflowed 1: Error occurrence counter has overflowed	R/(W) *2
b9	SCOV	Success Occurrence Counter Overflow Flag*1	0: Success occurrence counter has not overflowed 1: Success occurrence counter has overflowed	R/(W) *2
b14 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	TDCV	Transceiver Delay Compensation Violation Flag*1	0: Transceiver delay compensation violation has not occurred 1: Transceiver delay compensation violation has occurred	R/(W) *2
b23 to b16	EC[7:0]	Error Occurrence Counter	These bits show the error occurrence counter value.	R
b31 to b24	SC[7:0]	Success Occurrence Counter	These bits show the success occurrence counter value.	R

Note 1. Set this flag to 0 only in CH_OPERATION or CH_HALT mode.
Note 2. Writing 1 has no effect. This flag is cleared by writing 0 to it.

This register indicates the transceiver compensation delay result and its related FIFO message lost status.

TDCR[7:0] Bits (Transceiver Delay Compensation Result)

The TDCR[7:0] bits are set when the transceiver delay has been measured.
The measured delay is represented by the number of the DLL clock cycles. The result depends on the configuration of the FDCFG.SSPC bit and the offset value in the FDCFG.TDCO[7:0] bits. Refer to section 36.4.1.5, Transceiver Delay Compensation for details.
The TDCR[7:0] bits are updated at the falling edge between the FDF bit and res bit when the FDCFG.SSPC bit is 0 and the FDCFG.TDCE bit is 1 (transceiver delay compensation is enabled).
These bits are automatically set to 0 when the CAN channel is in CH_RESET mode.

ECOV Flag (Error Occurrence Counter Overflow Flag)

The ECOV flag indicates whether the CAN channel error occurrence counter has overflowed.
This flag is set to 1 if a CAN bus error specified in the FDCFG.ECC[2:0] bits is detected when the EC[7:0] bits are FFh.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
This flag is automatically set to 0 when the CAN channel is in CH_RESET mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.

SCOV Flag (Success Occurrence Counter Overflow Flag)

The SCOV flag indicates whether the CAN channel success occurrence counter has overflowed.

This flag is set to 1 if a successful message reception or successful message transmission occurs when the SC[7:0] bits are FFh.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
This flag is automatically set to 0 when the CAN channel is in CH_RESET mode.
Do not use the bit clear instruction to clear this flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.

TDCV Flag (Transceiver Delay Compensation Violation Flag)

The CANFD module captures internally the transmitted data bit-by-bit. This data is then compared against the received CAN bus level which is delayed by the transceiver loop delay.
The transceiver delay has some variations depending on the physical parameters such as temperature. The TDCR[7:0] bits are updated by each message. However, temporary maximum delay violation could be missed. Therefore, the TDCV flag captures this violation.
This flag is set to 1 when the transceiver delay compensation is greater than the maximum delay compensation (6 data bit times – 2 DLL clock) and the internal bit overruns.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
This flag is automatically set to 0 when the CAN channel is in CH_RESET mode.
Do not use the bit clear instruction to clear this flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.

EC[7:0] Bits (Error Occurrence Counter)

The EC[7:0] bits are used together with the SC[7:0] bits to support an option for host-controlled fall-back to payload bit rate identical to arbitration bit rate when messages utilizing the reduced payload bit length have significant higher error rates compared to other messages.
This higher error rate can be detected depending on the configuration of the FDCFG.ECC[2:0] bits.
The EC[7:0] bits are set only by CANFD module logic. These bits are cleared by writing 1 to the FDCTR.ECCL bit.
These bits are updated when an error occurs, according to the configuration of the FDCFG.ECC[2:0] bits. When the counter reaches the value of FFh, the update stops.
These bits are automatically set to 00h when the CAN channel is in CH_RESET mode.

SC[7:0] Bits (Success Occurrence Counter)

The SC[7:0] bits are used together with the EC[7:0] bits to support an option for host-controlled fall-back to payload bit rate identical to arbitration bit rate when messages utilizing the reduced payload bit length have significant higher error rates compared to other messages.
The SC[7:0] bits are set only by CANFD module logic. These bits are cleared by writing 1 to the FDCTR.SCCL bit.
These bits are updated when the occurrence of any error-free messages on the bus is detected through reception or transmission. When the counter reaches the value of FFh, the update stops.
These bits are automatically set to 00h when the CAN channel is in CH_RESET mode.

Note: In Loopback mode, the counter is incremented twice.

36.2.9 CAN FD CRC Register (FDCRC)

Address(es): CANFD0.FDCRC 000A 8110h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	SBC[3:0]				—	—	—	CRC21[20:16]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CRC21[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b20 to b0	CRC21[20:0]	CRC_21 Test	These bits show the CRC_17 value or CRC_21 value calculated for the CAN FD frame.	R
b23 to b21	—	Reserved	These bits are read as 0.	R
b27 to b24	SBC[3:0]	Stuff Bit Counter	These bits shows the stuff bit count (Mod 8) for the CAN FD frame.	R
b31 to b28	—	Reserved	These bits are read as 0.	R

This register holds the CRC value calculated for the CAN FD frame.

CRC21[20:0] Bits (CRC_21 Test)

The calculated CRC_17 value or CRC_21 value can be read from this bits when the CHCR.CTME bit is 1 (channel test mode).
When the CHCR.CTME bit is 0, the CRC21[20:0] bits are read as 000000h.
The CRC21[20:0] bits is updated in the first bit of the CRC field of the CAN FD frame.
When the CRC_17 field is used, the CRC21[20:17] bits are read as 0.
These bits are automatically set to 000000h when the CAN channel is in CH_RESET mode.

SBC[3:0] Bits (Stuff Bit Counter)

The SBC[3:0] bits contain the stuff count value of the CAN FD frame. These bits indicate the number of inserted stuff bits (modulo 8, Graycoded) for a CAN FD frame when the CHCR.CTME bit is enabled in SBC[3:1]. SBC[0] is the parity bit.
When the CHCR.CTME bit is 0, the SBC[3:0] bits are always read as 0000b.
The value of the SBC[3:0] bits is updated in the first bit of CRC field of the CAN FD frame. These bits are automatically set to 0000b when the CAN channel is in CH_RESET mode.

36.2.10 Global Configuration Register (GCFG)

Address(es): CANFD.GCFG 000A 8014h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ITP[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	TSCS	TSP[3:0]				—	—	OMRC	DLLCS	MME	DRE	DCE	TPRI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TPRI	Transmission Priority Setting	0: ID priority 1: Message buffer number priority	R/W
b1	DCE	DLC Check Enable	0: DLC check is disabled 1: DLC check is enabled	R/W
b2	DRE	DLC Replacement Enable	0: DLC replacement is disabled 1: DLC replacement is enabled	R/W
b3	MME	Mirror Mode Enable	0: Mirror mode is disabled 1: Mirror mode is enabled	R/W
b4	DLLCS	DLL Clock Select	0: CANFDCLK 1: CANFDMCLK	R/W
b5	OMRC	Payload-Overflowed Message Reception Configuration	0: The message is discarded. 1: The message payload is cut to fit the specified message size	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11 to b8	TSP[3:0]	Timestamp Counter Prescaler Setting	b11 b8 0 0 0 0: No division 0 0 0 1: Divide-by-2 (= 2^1) 0 0 1 0: Divide-by-4 (= 2^2) 0 0 1 1: Divide-by-8 (= 2^3) : : 1 1 0 1: Divide-by-8192 (= 2^{13}) 1 1 1 0: Divide-by-16384 (= 2^{14}) 1 1 1 1: Divide-by-32768 (= 2^{15})	R/W
b12	TSCS	Timestamp Counter Source Select	0: Count source for timestamp counter is peripheral module clock (PCLKB) 1: Count source for timestamp counter is bit time clock	R/W
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31 to b16	ITP[15:0]	Interval Timer Prescaler Setting	FIFO interval timer prescaler value. Set the divided value for the peripheral module clock (PCLKB)	R/W

This register is used to select the transmission priority to be used for all the transmit message buffers and the clock source for the CAN protocol engine. The GCFG register is also used to select the source for the timestamp clock and to configure the frequency for the timestamp clock and interval timer reference clock.

TPRI Bit (Transmission Priority Setting)

The TPRI bit selects the transmission priority for the CAN channel.

This bit cannot be written in GL_SLEEP mode. Write to this bit only when CANFD module is in GL_RESET mode. Do not select message buffer number priority when using the transmit queue.

DCE Bit (DLC Check Enable)

The DCE bit enables data length code (DLC) check for the CAN channel.
This bit cannot be written in GL_SLEEP mode. Write to this bit only when CANFD module is in GL_RESET mode.

DRE Bit (DLC Replacement Enable)

When the DRE bit is 1 and the DCE bit is 1, the CANFD module stores the configured value (AFLn.PTR0.DLC[3:0]) of the DLC in the destination receive message buffer or FIFO buffer if the DLC check passes. Otherwise, the DLC value in the destination receive message buffer or FIFO buffer is unchanged.
This bit cannot be written in GL_SLEEP mode. Write to this bit only when CANFD module is in GL_RESET mode.

MME Bit (Mirror Mode Enable)

The MME bit enables the Mirror mode for the CAN channel.
This bit cannot be written in GL_SLEEP mode. Write to this bit only when CANFD module is in GL_RESET mode.

DLLCS Bit (DLL Clock Select)

The DLLCS bit selects the clock source for CAN communication.This bit cannot be written in GL_SLEEP or GL_OPERATION mode. Write to this bit only when CANFD module is in GL_RESET mode.

OMRC Bit (Payload-Overflowed Message Reception Configuration)

The OMRC bit controls the message payload acceptance mechanism when the received payload is higher than the message buffer payload size (RMCR.PLS[2:0], RFCRn.PLS[2:0], and CFCR0.PLS[2:0]). The received message payload is always compared with the available message payload size in the message buffer.
This bit cannot be written in GL_SLEEP or GL_OPERATION mode. Write to this bit only when CANFD module is in GL_RESET mode.
When this bit is set to 1 and payload overflow occurs, the DLC value is stored in the receive message buffer or FIFO buffer unchanged.

TSP[3:0] Bits (Timestamp Counter Prescaler Setting)

The value configured in the TSP[3:0] bits defines the period of the count source used for the timestamp counter.
These bits cannot be written in GL_SLEEP mode. Write to this bit only when CANFD module is in GL_RESET mode.

TSCS Bit (Timestamp Counter Source Select)

The TSCS bit allows the selection of the count source for the timestamp counter.
This bit cannot be written in GL_SLEEP mode. Write to this bit only when CANFD module is in GL_RESET mode.
Additionally, do not set this bit to 1 when CAN FD communication is used.

Note: The bit time clock varies depending on the nominal bit rate and data bit rate configuration.

ITP[15:0] Bits (Interval Timer Prescaler Setting)

The ITP[15:0] bits allow the definition of a reference clock for the FIFO interval timer count source. When these bits are 0000h, the timer is disabled.
These bits cannot be written in GL_SLEEP mode. Write to these bits only when CANFD module is in GL_RESET mode.

36.2.11 Global Control Register (GCR)

Address(es): CANFD.GCR 000A 8018h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSCR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	POIE	THLIE	MLIE	DEIE	—	—	—	—	—	SLPRQ	MDC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	MDC[1:0]	Global Mode Control	b1 b0 0 0: Request transition to GL_OPERATION mode 0 1: Request transition to GL_RESET mode 1 0: Request transition to GL_HALT mode 1 1: Keep current value	R/W
b2	SLPRQ	GL_SLEEP Mode Request	0: Request to release GL_SLEEP 1: Request transition to GL_SLEEP	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	DEIE	DLC Error Interrupt Enable	0: DLC error interrupt is disabled 1: DLC error interrupt is enabled	R/W
b9	MLIE	Message Lost Interrupt Enable	0: Message lost interrupt is disabled 1: Message lost interrupt is enabled	R/W
b10	THLIE	Transmission History Entry Lost Interrupt Enable	0: Transmission history entry lost interrupt is disabled 1: Transmission history entry lost interrupt is enabled	R/W
b11	POIE	Payload Overflow Interrupt Enable	0: Payload overflow interrupt is disabled 1: Payload overflow interrupt is enabled	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	TSCR	Timestamp Counter Reset	When 1 is written to this bit, the timestamp counter is reset. This bit is read as 0.	R/W
b31 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register controls the Global mode of the CANFD module and the timestamp function. The register also enables and disables the global error interrupts.

MDC[1:0] Bits (Global Mode Control)

The MDC[1:0] bits are used to specify the modes of the CANFD module. For the mode transition of the CANFD module, refer to section 36.3.1, Global Modes.

These bits cannot be written in GL_SLEEP mode.

To request the CANFD module to transition to GL_SLEEP mode, first set these bits to 01b to enter GL_RESET mode, then set the GCR.SLPRQ bit to 1.

SLPRQ Bit (GL_SLEEP Mode Request)

The SLPRQ bit is used to control the transition to GL_SLEEP mode and the return from GL_SLEEP mode.

When this bit is set to 1, the transition to CH_SLEEP mode is also requested for CAN channel.

This bit can only be written when the CANFD module is in GL_RESET or GL_SLEEP mode.

DEIE Bit (DLC Error Interrupt Enable)

When the DEIE bit is 1, an interrupt is generated if a DLC error is detected in the received frames.

This bit cannot be written in GL_SLEEP mode.

MLIE Bit (Message Lost Interrupt Enable)

When the MLIE bit is 1, an interrupt is generated if a message lost condition occurs.
This bit cannot be written in GL_SLEEP mode.

THLIE Bit (Transmission History Entry Lost Interrupt Enable)

When the THLIE bit is 1, an interrupt is generated if a transmission history entry lost condition occurs.
This bit cannot be written in GL_SLEEP mode.

POIE Bit (Payload Overflow Interrupt Enable)

When the POIE bit is 1, an interrupt is generated when a message payload overflow condition occurs.
This bit cannot be written in GL_SLEEP mode.

TSCR Bit (Timestamp Counter Reset)

When the TSCR bit is 1, the Timestamp Counter Register (TSCR) is reset to 00000000h.
This bit cannot be written in GL_SLEEP mode. Do not write to this bit in GL_RESET mode.
This bit is automatically set to 0.

36.2.12 Global Status Register (GSR)

Address(es): CANFD.GSR 000A 801Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	RAMST	SLPST	HLTST	RSTST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	RSTST	GL_RESET Status Flag	0: Not in GL_RESET mode 1: In GL_RESET mode	R
b1	HLTST	GL_HALT Status Flag	0: Not in GL_HALT mode 1: In GL_HALT mode	R
b2	SLPST	GL_SLEEP Status Flag	0: Not in GL_SLEEP mode 1: In GL_SLEEP mode	R
b3	RAMST	RAM Initialization Status Flag	0: RAM initialization is completed 1: RAM initialization is in progress	R
b31 to b4	—	Reserved	These bits are read as 0.	R

This register indicates the global status of the CANFD module.

RSTST Flag (GL_RESET Status Flag)

The RSTST flag indicates the status of the CANFD module in GL_RESET mode.
This flag is automatically set to 1 when the CANFD module enters GL_RESET mode. When the mode changes from GL_RESET mode to GL_SLEEP mode, this bit remains 1. This flag is automatically set to 0 when the CANFD module enters GL_HALT or GL_OPERATION mode.

HLTST Flag (GL_HALT Status Flag)

The HLTST flag indicates the status of the CANFD module in GL_HALT mode.
This flag is automatically set to 1 when the CANFD module enters GL_HALT mode. This flag is automatically set to 0 when the CANFD module exits the GL_HALT mode.

SLPST Flag (GL_SLEEP Status Flag)

The SLPST flag indicates the status of the CANFD module in GL_SLEEP mode.
This flag is automatically set to 1 when the CANFD module enters GL_SLEEP mode. This flag is automatically set to 0 when the CANFD module exits the GL_SLEEP mode.

RAMST Flag (RAM Initialization Status Flag)

The RAMST flag indicates the status of the CANFD module's RAM initialization.
This flag is automatically set to 1 when the CANFD module enters GL_SLEEP mode after release from MCU reset. This flag is automatically set to 0 when the CANFD module completed RAM initialization.

36.2.13 Global Error Status Register (GESR)

Address(es): CANFD.GESR 000A 8020h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	EEDF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	PODF	THLDF	MLDF	DEDF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DEDF	DLC Error Detect Flag	0: DLC error is not detected 1: DLC error is detected	R/(W) *1
b1	MLDF	Message Lost Detect Flag	0: Message lost error is not detected 1: Message lost error is detected	R
b2	THLDF	Transmission History Entry Lost Detect Flag	0: Transmission history entry lost is not detected 1: Transmission history entry lost is detected	R
b3	PODF	Payload Overflow Detect Flag	0: Payload overflow is not detected 1: Payload overflow is detected	R/(W) *1
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	EEDF0	Channel 0 ECC Error Detect Flag	0: ECC error is not detected during transmission scan 1: ECC error is detected during transmission scan	R/(W) *1
b31 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing 1 has no effect. This flag is cleared by writing 0 to it.

This register indicates the detection of global errors.

DEDF Flag (DLC Error Detect Flag)

The DEDF flag indicates the error status of the DLC.
This flag cannot be written in GL_SLEEP or GL_RESET mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.
This flag is automatically set to 1 when a DLC error is detected in a received frame.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
The flag is cleared by writing 0 to it. This flag is automatically set to 0 in GL_RESET mode.

MLDF Flag (Message Lost Detect Flag)

The MLDF flag indicates the status of the message lost error.
This flag is automatically set to 1 when a FIFO message lost error is detected.
This flag is automatically set to 0 when:

- All FIFO message lost flags (RFSRn.LOST, CFSR0.LOST) are cleared
- The CANFD module is in GL_RESET mode.

THLDF Flag (Transmission History Entry Lost Detect Flag)

The THLDF flag indicates the status of the transmission history entry lost error.
This flag is automatically set to 1 when a transmission history entry lost error is detected.
This flag is automatically set to 0 when:

- The transmission history lost flag (THSR.LOST) is cleared
- The CANFD module is in GL_RESET mode.

PODF Flag (Payload Overflow Detect Flag)

The PODF flag is automatically set to 1 when a message payload overflow is detected on at least one channel.
This flag cannot be written in GL_SLEEP or GL_RESET mode.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.
This flag is automatically set to 0 in GL_RESET mode.

EEDF0 Flag (Channel 0 ECC Error Detect Flag)

The EEDF0 flag specifies whether an ECC error has occurred.
This flag cannot be written in GL_SLEEP or GL_RESET mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
This bit is automatically set to 0 in GL_RESET mode.

36.2.14 Transmit Interrupt Status Register (TISR)

Address(es): CANFD.TISR 000A 80A4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	THIF0	CFTIF0	TQIF0	TAIF0	TSIF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TSIF0	Channel 0 Transmission Successful Interrupt Flag	0: Channel 0 transmission successful interrupt is not generated. 1: Channel 0 transmission successful interrupt is generated.	R
b1	TAIF0	Channel 0 Transmission Abort Interrupt Flag	0: Channel 0 transmission abort interrupt is not generated. 1: Channel 0 transmission abort interrupt is generated.	R
b2	TQIF0	Channel 0 Transmit Queue Interrupt Flag	0: Channel 0 transmit queue interrupt is not generated. 1: Channel 0 transmit queue interrupt is generated.	R
b3	CFTIF0	Channel 0 Common FIFO Transmission Interrupt Flag	0: Channel 0 common FIFO transmission interrupt is not generated. 1: Channel 0 common FIFO transmission interrupt is generated.	R
b4	THIF0	Channel 0 Transmission History Interrupt Flag	0: Channel 0 transmission history interrupt is not generated. 1: Channel 0 transmission history interrupt is generated.	R
b31 to b5	—	Reserved	These bits are read as 0.	R

This register indicates the detection of transmit specific interrupts.

TSIF0 Flag (Channel 0 Transmission Successful Interrupt Flag)

The TSIF0 flag becomes 1 if transmission from the transmit message buffer n on channel 0 is successful when the transmit message buffer n interrupt is enabled.
This flag is automatically set to 0:

- When the interrupt is disabled (TMIER0.TMIEn bits = 0)
- When the Transmission Result flags (TMSRn.TXRF[1:0]) are cleared
- When in GL_RESET or CH_RESET mode.

TAIF0 Flag (Channel 0 Transmission Abort Interrupt Flag)

The TAIF0 flag becomes 1 if transmission from the transmit message buffer n on channel 0 is aborted when the transmission abort interrupt is enabled.
This flag is automatically set to 0:

- When the interrupt is disabled (CHCR.TAIE bit = 0)
- When the Transmission Result flags (TMSRn.TXRF[1:0]) are cleared
- When in GL_RESET or CH_RESET mode.

TQIF0 Flag (Channel 0 Transmit Queue Interrupt Flag)

The TQIF0 flag becomes 1 if the Transmit Queue Interrupt flag of channel 0 is set to 1 when the transmit queue interrupt is enabled.

This flag is automatically set to 0:

- When the interrupt is disabled (TQCR0.TQIE bit = 0)
- When the Transmission Result flags (TQSR0.TQIF) is cleared
- When in GL_RESET or CH_RESET mode.

CFTIF0 Flag (Channel 0 Common FIFO Transmission Interrupt Flag)

The CFTIF0 flag becomes 1 if the Common FIFO Transmit Interrupt flag (CFSR0.CFTIF) of channel 0 is set to 1 when the common FIFO transmit interrupt is enabled.

This flag is automatically set to 0:

- When the interrupt is disabled (CFCR0.CFTIE bit = 0)
- When the Common Transmit FIFO Interrupt flag (CFSR0.CFTIF) is cleared
- When in GL_RESET or CH_RESET mode.

THIF0 Flag (Channel 0 Transmission History Interrupt Flag)

The THIF0 flag becomes 1 if the Transmission History Interrupt flag (THSR.THIF) of channel 0 is set to 1 when the transmission history interrupt is enabled.

This flag is automatically set to 0:

- When the interrupt is disabled (THCR.THIE bit = 0)
- When the Transmission History Interrupt flag (THSR.THIF) is cleared
- When in GL_RESET or CH_RESET mode.

36.2.15 Timestamp Counter Register (TSCR)

Address(es): CANFD.TSCR 000A 8024h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register stores the timestamp based on the selected configuration.
The timestamp value is stored in the TSCR register based on the configuration of the GCFG.TSCS bit and GCFG.TSP[3:0] bits. The accuracy of the timestamp counter cannot be guaranteed when transitioning to GL_HALT state.
This register is automatically set to 00000000h in GL_RESET mode.

36.2.16 Acceptance Filter List Control Register (AFCR)

Address(es): CANFD.AFCR 000A 8028h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	AFLWE	—	—	—	—	—	—	—	PAGE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PAGE	Access Page Setting	Select the page number of the Acceptance Filter List	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	AFLWE	AFL Write Enable	0: Acceptance Filter List data access is disabled 1: Acceptance Filter List data access is enabled	R/W
b31 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to select the Acceptance Filter List page for reading or writing entries into the Acceptance Filter List.

PAGE Bit (Access Page Setting)

The PAGE bit is used to select the page number to access the desired RAM area of the Acceptance Filter List. One Acceptance Filter List page consists of 16 Acceptance Filter List entries.
Read/write accesses to the Acceptance Filter List can only be performed through a fixed window.
This bit cannot be written in GL_SLEEP mode.

AFLWE Bit (AFL Write Enable)

The AFLWE bit prevents write access to the Acceptance Filter List by setting this bit to 0 after configuration of the Acceptance Filter List.
Data can be read from the Acceptance Filter List independent of the status of this bit.
This bit cannot be written in GL_SLEEP mode.
Setting this bit to 1 enables write access for the Acceptance Filter List.

36.2.17 Acceptance Filter List Configuration Register (AFCFG)

Address(es): CANFD.AFCFG 000A 802Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	RN0[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b21 to b16	RN0[5:0]	Channel 0 Number of Rules Setting	Number of rules in the Acceptance Filter List	R/W
b31 to b22	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to define the number of rules for entry in the Acceptance Filter List.
The maximum number of available entries in the Acceptance Filter List is 32.

RN0[5:0] Bits (Channel 0 Number of Rules Setting)

The RN0[5:0] bits define the number of rules in the Acceptance Filter List. This bit can set 32 rules or less.
These bits can only be written in GL_RESET mode.

36.2.18 Acceptance Filter List n ID Register (AFLn.IDR) (n = 0 to 15)

Address(es): CANFD.AFL0.IDR 000A 8120h, CANFD.AFL1.IDR 000A 8130h, CANFD.AFL2.IDR 000A 8140h, CANFD.AFL3.IDR 000A 8150h, CANFD.AFL4.IDR 000A 8160h, CANFD.AFL5.IDR 000A 8170h, CANFD.AFL6.IDR 000A 8180h, CANFD.AFL7.IDR 000A 8190h, CANFD.AFL8.IDR 000A 81A0h, CANFD.AFL9.IDR 000A 81B0h, CANFD.AFL10.IDR 000A 81C0h, CANFD.AFL11.IDR 000A 81D0h, CANFD.AFL12.IDR 000A 81E0h, CANFD.AFL13.IDR 000A 81F0h, CANFD.AFL14.IDR 000A 8200h, CANFD.AFL15.IDR 000A 8210h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IDE	RTR	LPC	ID[28:16]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ID[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b28 to b0	ID[28:0]	ID Field	ID part of the Acceptance Filter List entry	R/W
b29	LPC	Loopback Configuration	0: Message with reception attribute 1: Message with transmission attribute	R/W
b30	RTR	RTR	0: Data frame 1: Remote frame	R/W
b31	IDE	IDE	0: Standard ID 1: Extended ID	R/W

This register is used to configure the ID field in the rule entry of the Acceptance Filter List.
When rewriting this register, set the AFCR.AFLWE bit to 1. When the AFLWE bit is 0, it cannot be rewritten.

ID[28:0] Bits (ID Field)

The ID[28:0] bits indicate the CAN identifier (ID) field of each entry in the Acceptance Filter List.
The acceptance filter process compares this field against the ID of a received message. For alignment of these bits in base and extended formats, refer to section 36.2.60, Identifier Bits Alignment.
Write to these bits only when the CAN channel is in CH_RESET or CH_HALT mode.

LPC Bit (Loopback Configuration)

The LPC bit is used to select whether entry in the Acceptance Filter List gets the reception or transmission attribute. This attribute determines the validity of the entry in Mirror mode, Loopback mode, and during standard (non-loopback) reception. Refer to section 36.5.8, Loopback Modes for detailed description of the validity of the Acceptance Filter List entry depending on transmitter/receiver case, the type of loopback mode, and reception/transmission attribute.
Write to this bit only when the CAN channel is in CH_RESET or CH_HALT mode.

RTR Bit (RTR)

The RTR bit allows the configuration of the specified frame format (data frame or remote frame) for each entry of the Acceptance Filter List. For each rule entry in a CAN channel, the acceptance filter process compares this bit against the RTR bit of the received message.
Write to this bit only when the CAN channel is in CH_RESET or CH_HALT mode.

IDE Bit (IDE)

The IDE bit allows the configuration of the ID format (standard ID or extended ID) for each entry in the Acceptance Filter List. For each rule entry in a CAN channel, the acceptance filter process compares this bit against the IDE bit of the

received message.
Write to this bit only when the CAN channel is in CH_RESET or CH_HALT mode.

36.2.19 Acceptance Filter List n Mask Register (AFLn.MASK) (n = 0 to 15)

Address(es): CANFD.AFL0.MASK 000A 8124h, CANFD.AFL1.MASK 000A 8134h, CANFD.AFL2.MASK 000A 8144h, CANFD.AFL3.MASK 000A 8154h, CANFD.AFL4.MASK 000A 8164h, CANFD.AFL5.MASK 000A 8174h, CANFD.AFL6.MASK 000A 8184h, CANFD.AFL7.MASK 000A 8194h, CANFD.AFL8.MASK 000A 81A4h, CANFD.AFL9.MASK 000A 81B4h, CANFD.AFL10.MASK 000A 81C4h, CANFD.AFL11.MASK 000A 81D4h, CANFD.AFL12.MASK 000A 81E4h, CANFD.AFL13.MASK 000A 81F4h, CANFD.AFL14.MASK 000A 8204h, CANFD.AFL15.MASK 000A 8214h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IDEM	RTRM	IFL1	IDM[28:16]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	IDM[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b28 to b0	IDM[28:0]	ID Mask Field	0: Corresponding ID bit is not used for ID matching 1: Corresponding ID bit is used for ID matching	R/W
b29	IFL1	Information Label 1	Configure an information label 1 to be attached to a received message	R/W
b30	RTRM	RTR Mask	0: RTR bit is not used for ID matching 1: RTR bit is used for ID matching	R/W
b31	IDEM	IDE Mask	0: IDE bit is not used for ID matching 1: IDE bit is used for ID matching	R/W

This register is used to configure the mask field of each rule for entries in the Acceptance Filter List.
When rewriting this register, set the AFCR.AFLWE bit to 1. When the AFLWE bit is 0, it cannot be rewritten.

IDM[28:0] Bits (ID Mask Field)

The IDM[28:0] bits are the filter mask bits for the related bits in the CAN Identifier field of each Acceptance Filter List entry.
Write to these bits only when the CAN channel is in CH_RESET or CH_HALT mode.

IFL1 Bit (Information Label 1)

The IFL1 bit is the upper bit of a 2-bit information label to be attached to a received message accepted by the associated entry in the Acceptance Filter List.
Write to this bit only when the CAN channel is in CH_RESET or CH_HALT mode.
This bit is stored in the upper bit of the Information Label Field (RMBn.HF2.IFL[1], RFBn.HF2.IFL[1], CFB0.HF2.IFL[1]) of the storage location of a received message.

RTRM Bit (RTR Mask)

The RTRM bit is the RTR mask bit for each entry in the Acceptance Filter List.
Write to this bit only when the CAN channel is in CH_RESET or CH_HALT mode.

IDEM Bit (IDE Mask)

The IDEM bit is the IDE mask bit for each entry in the Acceptance Filter List.
When the IDE mask bit is 0, the ID comparison depends on the IDE bit of the received message.

- If the IDE bit of the received message is 0, the standard ID comparison takes place.
- If the IDE bit of the received message is 1, the extended ID comparison takes place.

Write to this bit only when the CAN channel is in CH_RESET or CH_HALT mode.

36.2.20 Acceptance Filter List n Pointer 0 Register (AFLn.PTR0) (n = 0 to 15)

Address(es): CANFD.AFL0.PTR0 000A 8128h, CANFD.AFL1.PTR0 000A 8138h, CANFD.AFL2.PTR0 000A 8148h, CANFD.AFL3.PTR0 000A 8158h, CANFD.AFL4.PTR0 000A 8168h, CANFD.AFL5.PTR0 000A 8178h, CANFD.AFL6.PTR0 000A 8188h, CANFD.AFL7.PTR0 000A 8198h, CANFD.AFL8.PTR0 000A 81A8h, CANFD.AFL9.PTR0 000A 81B8h, CANFD.AFL10.PTR0 000A 81C8h, CANFD.AFL11.PTR0 000A 81D8h, CANFD.AFL12.PTR0 000A 81E8h, CANFD.AFL13.PTR0 000A 81F8h, CANFD.AFL14.PTR0 000A 8208h, CANFD.AFL15.PTR0 000A 8218h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PTR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DMBE	—	—	DMB[4:0]					IFL0	—	—	—	DLC[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	DLC[3:0]	DLC Field	Configure a minimum DLC value of received message	R/W
b6 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	IFL0	Information Label 0	Configure an information label 0 to be attached to a received message	R/W
b12 to b8	DMB[4:0]	Destination Message Buffer Setting	Configure the receive message buffer number for storage of received messages	R/W
b14 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	DMBE	Destination Message Buffer Setting Enable	0: The Destination Message Buffer Setting bit is disabled 1: The Destination Message Buffer Setting bit is enabled	R/W
b31 to b16	PTR[15:0]	Pointer	Configure a 16-bit pointer to be attached to a received message	R/W

This register is used to configure the data length code (DLC), software pointer, and destination message buffer for each rule entry in the Acceptance Filter List.
When rewriting this register, set the AFCR.AFLWE bit to 1. When the AFLWE bit is 0, it cannot be rewritten.

DLC[3:0] Bits (DLC Field)

The DLC[3:0] bits allow the configuration of a minimum data length code (DLC) value for a message to be accepted by the associated entry in the Acceptance Filter List (automatic DLC filter function).
DLC filter process is only passed if the DLC value of the message accepted by an entry in the Acceptance Filter List is equal to or higher than the DLC value configured for this associated Acceptance Filter List entry. Automatic DLC filter function is disabled for the corresponding rule entry when this field is set to 0000b.
Write to these bits only when the CAN channel is in CH_RESET or CH_HALT mode.

IFL0 Bit (Information Label 0)

The IFL0 bit is the lower bit of a 2-bit information label that can be attached to a received message accepted by the Acceptance Filter List entry.
Write to this bit only when the CAN channel is in CH_RESET or CH_HALT mode.
This bit is stored in the lower bit of the Information Label Field (RMBn.HF2.IFL[0], RFBn.HF2.IFL[0], CFB0.HF2.IFL[0]) of the storage location of a received message.

DMB[4:0] Bits (Destination Message Buffer Setting)

The DMB[4:0] bits allow the configuration of a message buffer as the destination for a received message that passes the acceptance check of the Acceptance Filter List entry. Configure the destination message buffer number.
Write to these bits only when the CAN channel is in CH_RESET or CH_HALT mode.
Write the RMCR.NMB[5:0] bits to configure the number of receive message buffers. The value to be entered in the DMB[4:0] bits should only be between 00000b and 'NMB[5:0] – 1'.
If RMCR.NMB[5:0] is 000000b, set the DMBE bit to 0.

DMBE Bit (Destination Message Buffer Setting Enable)

The DMBE bit allows the enabling or disabling of the receive message buffer as the destination message buffer for a received message that passes the acceptance check of the Acceptance Filter List entry.
Write to this bit only when the CAN channel is in CH_RESET or CH_HALT mode.

PTR[15:0] Bits (Pointer)

The PTR[15:0] bits allow the configuration of a 16-bit pointer to be attached to a received message accepted by the Acceptance Filter List entry. The pointer is added during message storage in the message buffer area and can be used by the application as a support function. The pointer information can be used for example, to support PDU Identifier allocation for the received message in AUTOSAR systems.
Write to these bits only when the CAN channel is in CH_RESET or CH_HALT mode.

36.2.21 Acceptance Filter List n Pointer 1 Register (AFLn.PTR1) (n = 0 to 15)

Address(es): CANFD.AFL0.PTR1 000A 812Ch, CANFD.AFL1.PTR1 000A 813Ch, CANFD.AFL2.PTR1 000A 814Ch, CANFD.AFL3.PTR1 000A 815Ch, CANFD.AFL4.PTR1 000A 816Ch, CANFD.AFL5.PTR1 000A 817Ch, CANFD.AFL6.PTR1 000A 818Ch, CANFD.AFL7.PTR1 000A 819Ch, CANFD.AFL8.PTR1 000A 81ACh, CANFD.AFL9.PTR1 000A 81BCh, CANFD.AFL10.PTR1 000A 81CCh, CANFD.AFL11.PTR1 000A 81DCh, CANFD.AFL12.PTR1 000A 81ECh, CANFD.AFL13.PTR1 000A 81FCh, CANFD.AFL14.PTR1 000A 820Ch, CANFD.AFL15.PTR1 000A 821Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CF0E	—	—	—	—	—	—	RF1E	RF0E
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RF0E	Receive FIFO 0 Destination Enable	0: Do not select receive FIFO 0 as the message storage destination 1: Selects receive FIFO 0 as the message storage destination	R/W
b1	RF1E	Receive FIFO 1 Destination Enable	0: Do not select receive FIFO 1 as the message storage destination 1: Selects receive FIFO 1 as the message storage destination	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	CF0E	Common FIFO 0 Destination Enable	0: Do not select common FIFO 0 as the message storage destination 1: Selects common FIFO 0 as the message storage destination	R/W
b31 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to configure the destination FIFO buffer for each rule entry in the Acceptance Filter List.
Up to two storage destinations can be specified for received messages. Two FIFO buffers, or one FIFO buffer and one receive message buffer are valid.
When rewriting this register, set the AFCR.AFLWE bit to 1. When the AFLWE bit is 0, it cannot be rewritten.
Write to this register only when the CAN channel is in CH_RESET or CH_HALT mode.

RF0E Bit (Receive FIFO 0 Destination Enable)

The RF0E bit allows the configuration of the receive FIFO 0 as a storage destination of a received message that passes the acceptance check of the Acceptance Filter List entry.

RF1E Bit (Receive FIFO 1 Destination Enable)

The RF1E bit allows the configuration of the receive FIFO 1 as a storage destination of a received message that passes the acceptance check of the Acceptance Filter List entry.

CF0E Bit (Common FIFO 0 Destination Enable)

The CF0E bit allows the configuration of the common FIFO 0 as a storage destination of a received message that passes the acceptance check of the Acceptance Filter List entry.
The common FIFO 0 must be configured as the receive FIFO.

36.2.22 Receive Message Buffer Configuration Register (RMCR)

Address(es): CANFD.RMCR 000A 8030h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	PLS[2:0]			—	—	NMB[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	NMB[5:0]	Number of Message Buffer Setting	Configure the number of receive message buffers	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b10 to b8	PLS[2:0]	Payload Size Setting	b10 b8 0 0 0: 8 bytes 0 0 1: 12 bytes 0 1 0: 16 bytes 0 1 1: 20 bytes 1 0 0: 24 bytes 1 0 1: 32 bytes 1 1 0: 48 bytes 1 1 1: 64 bytes	R/W
b31 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to configure the total number of receive message buffers allocated to a channel.

NMB[5:0] Bits (Number of Message Buffer Setting)

The NMB[5:0] bits are used to configure the number of receive message buffers.
These bits can only be written in GL_RESET mode.
Set a value between 0 and 32. 0 indicates that no receive message buffer is allocated.

PLS[2:0] Bits (Payload Size Setting)

The PLS[2:0] bits are used to configure the message buffer payload size.
These bits can only be written in GL_RESET mode.

36.2.23 Receive Message Buffer New Data Register (RMNDR)

Address(es): CANFD.RMNDR 000A 8034h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	NDF[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	NDF[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	NDF[31:0]	New Data Flag	0: New data is not stored in corresponding receive message buffer 1: New data is stored in corresponding receive message buffer	R/(W) *1

Note 1. Writing 1 has no effect. This flag is cleared by writing 0 to it.

This register specifies the new data storage status of the receive message buffers.
The bit number of this register corresponds to the buffer number of the received message buffer.

NDF[31:0] Flags (New Data Flag)

The NDF[31:0] flags indicate that the new data is stored in the corresponding receive message buffer. The NDF[0] flag corresponds to receive message buffer 0.
These flags are automatically set to 1 when storage of new messages are in the corresponding receive message buffer.
When RMCR.PLS[2:0] = 000b (up to 8 bytes payload), the duration of message storage is 6 PCLKB cycles.
When RMCR.PLS[2:0] > 000b, the duration of message storage is 6 PCLKB cycles + 1 for each 4 bytes (up to 20 PCLKB cycles for 64 bytes).
Do not write to these flags when the CANFD module is in GL_RESET or GL_SLEEP mode.
These flags are automatically set to 0 when the CANFD module is in GL_RESET mode.
These flags cannot be cleared when message storage in the corresponding receive message buffer is in progress.
Do not use the bit clear instruction to clear the flags. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.

36.2.24 Receive FIFO n Configuration Register (RFCRn) (n = 0, 1)

Address(es): CANFD.RFCR0 000A 803Ch, CANFD.RFCR1 000A 8040h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RFITH[2:0]			RFIM	—	FDS[2:0]			—	PLS[2:0]			—	—	RFIE	RFE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RFE	Receive FIFO Enable	0: FIFO is disabled 1: FIFO is enabled	R/W
b1	RFIE	Receive FIFO Interrupt Enable	0: FIFO interrupt generation is disabled 1: FIFO interrupt generation is enabled	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6 to b4	PLS[2:0]	Payload Size Setting	b6 b4 0 0 0: 8 bytes 0 0 1: 12 bytes 0 1 0: 16 bytes 0 1 1: 20 bytes 1 0 0: 24 bytes 1 0 1: 32 bytes 1 1 0: 48 bytes 1 1 1: 64 bytes	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10 to b8	FDS[2:0]	FIFO Depth Setting	b10 b8 0 0 0: 0 messages 0 0 1: 4 messages 0 1 0: 8 messages 0 1 1: 16 messages 1 0 0: 32 messages 1 0 1: 48 messages Settings other than above are prohibited.	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b12	RFIM	Receive FIFO Interrupt Mode Setting	0: Interrupt is generated when the number of messages stored in the receive FIFO reaches the value of the RFITH[2:0] bits from a value smaller than the RFITH[2:0] bits 1: Interrupt is generated at the end of every received message storage	R/W
b15 to b13	RFITH[2:0]	Receive FIFO Interrupt Threshold Setting	b15 b13 0 0 0: Interrupt generated when FIFO is 1/8 full 0 0 1: Interrupt generated when FIFO is 1/4 full 0 1 0: Interrupt generated when FIFO is 3/8 full 0 1 1: Interrupt generated when FIFO is 1/2 full 1 0 0: Interrupt generated when FIFO is 5/8 full 1 0 1: Interrupt generated when FIFO is 3/4 full 1 1 0: Interrupt generated when FIFO is 7/8 full 1 1 1: Interrupt generated when FIFO is full	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

These registers are used to configure and control the two receive FIFOs.

RFE Bit (Receive FIFO Enable)

The RFE bit enables the FIFO. When this bit is set to 0, the receive FIFO is cleared to empty.
This bit can only be written in GL_HALT or GL_OPERATION mode.
This bit can be set to 1 only when the FIFO depth is 4 to 48 (001b ≤ FDS[2:0] ≤ 101b).
Set the RFE bit to 1 with a separate write access to the RFCRn register, after all the other flags in the RFCRn register are set.
This bit is automatically set to 0 when the CANFD module is in GL_RESET mode.

RFIE Bit (Receive FIFO Interrupt Enable)

The RFIE bit enables generation of the FIFO interrupt.
This bit cannot be written in GL_SLEEP mode.

PLS[2:0] Bits (Payload Size Setting)

The PLS[2:0] bits define the message data payload size in the RAM.
This is the maximum number of bytes which can be received by this FIFO.
These bits can only be written in GL_RESET mode.

FDS[2:0] Bits (FIFO Depth Setting)

The FDS[2:0] bits select the depth of the FIFO in units of the number of messages. If the FIFO depth is set to 0, the FIFO cannot be used.
These bits can only be written in GL_RESET mode.

RFIM Bit (Receive FIFO Interrupt Mode Setting)

The RFIM bit selects the interrupt generation condition for the FIFO.
This bit cannot be written in GL_SLEEP mode.
Write to this bit only in GL_RESET mode.

RFITH[2:0] Bits (Receive FIFO Interrupt Threshold Setting)

The RFITH[2:0] bits select the counter value of the FIFO for generation of receive FIFO interrupts. These values represent fractions of the FIFO depth for which an interrupt is generated.
These bits cannot be written in GL_SLEEP mode.
The setting of the RFITH[2:0] bits is restricted by the value of the FDS[2:0] bits. For details, refer to section 36.6.2.1, FIFO Buffers Configuration.
Write to these bits only in GL_RESET mode.

36.2.25 Receive FIFO n Status Register (RFSRn) (n = 0, 1)

Address(es): CANFD.RFSR0 000A 8044h, CANFD.RFSR1 000A 8048h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	FLVL[5:0]						—	—	—	—	RFIF	LOST	FULL	EMPTY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	EMPTY	Receive FIFO Empty Flag	0: Message in receive FIFO 1: No message in receive FIFO (empty)	R
b1	FULL	Receive FIFO Full Flag	0: Receive FIFO not full 1: Receive FIFO full	R
b2	LOST	Message Lost Flag	0: Receive FIFO message lost has not occurred 1: Receive FIFO message lost has occurred	R/(W) *1
b3	RFIF	Receive FIFO Interrupt Flag	0: Receive FIFO interrupt condition is not satisfied 1: Receive FIFO interrupt condition is satisfied	R/(W) *1
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13 to b8	FLVL[5:0]	Receive FIFO Fill Level	Indicate the number of messages stored in receive FIFO	R
b31 to b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing 1 has no effect. This flag is cleared by writing 0 to it.

These registers show the status of messages stored in the corresponding FIFO buffers.

EMPTY Flag (Receive FIFO Empty Flag)

The EMPTY flag is automatically set to 1 when:

- The FLVL[5:0] flags are 00000b
- The RFCRn.RFE bit is set to 0 (receive FIFO is disabled)
- The CANFD module is in GL_RESET mode.

The EMPTY flag is automatically set to 0 when the first message is stored in the receive FIFO.

FULL Flag (Receive FIFO Full Flag)

The FULL flag is automatically set to 1 when the number of messages stored in the FIFO buffer matches the configured FIFO depth.

The FULL flag is automatically set to 0 when:

- The number of messages stored in the FIFO buffer is less than the configured FIFO depth
- The RFCRn.RFE bit is set to 0 (receive FIFO is disabled)
- The CANFD module is in GL_RESET mode.

LOST Flag (Message Lost Flag)

The LOST flag is automatically set to 1 whenever a message is lost due to attempted storage when the FIFO buffer is already full. This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.

The flag is set to 0:

- By writing 0 to it

- When the CANFD module is in GL_RESET mode.

Write to the LOST flag only when CANFD module is in GL_HALT or GL_OPERATION mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.

RFIF Flag (Receive FIFO Interrupt Flag)

The RFIF flag is automatically set to 1 when the configured interrupt condition is satisfied. This flag is not automatically cleared when the receive FIFO is disabled.
The flag is set to 0:

- By writing 0 to it
- When the CANFD module is in GL_RESET mode.

Write to this flag only when the CANFD module is in GL_HALT or GL_OPERATION mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.

FLVL[5:0] Bits (Receive FIFO Fill Level)

The FLVL[5:0] bits indicate the number of messages stored in the receive FIFO that can be read by the CPU. These bits are automatically set to 000000b when the FIFO is disabled and when the CANFD module is in GL_RESET mode.

36.2.26 Receive FIFO n Pointer Control Register (RFPCRn) (n = 0, 1)

Address(es): CANFD.RFPCR0 000A 804Ch, CANFD.RFPCR1 000A 8050h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register can be used to increment the read pointer of the corresponding receive FIFOs.
When the value 000000FFh is written to this register, the pointer of the corresponding receive FIFO is moved to the next FIFO entry. Write to this register only when the corresponding receive FIFO is enabled and not empty.
The read value from this register is always 00000000h.
This register can only be written in GL_HALT or GL_OPERATION mode.
Do not write to this register when DTC/DMA transfer is enabled (DTCR.RFDTEn bit = 1).

36.2.27 Common FIFO 0 Configuration Register (CFCR0)

Address(es): CANFD.CFCR0 000A 8054h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TINT[7:0]								FDS[2:0]			—	—	—	LTM[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CFITH[2:0]			CFIM	ITR	ITCS	—	MODE	—	PLS[2:0]			—	CFTIE	CFRIE	CFE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CFE	Common FIFO Enable	0: FIFO is disabled 1: FIFO is enabled	R/W
b1	CFRIE	Common FIFO Receive Interrupt Enable	0: FIFO receive interrupt generation is disabled 1: FIFO receive interrupt generation is enabled	R/W
b2	CFTIE	Common FIFO Transmit Interrupt Enable	0: FIFO transmit interrupt generation is disabled 1: FIFO transmit interrupt generation is enabled	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6 to b4	PLS[2:0]	Payload Size Setting	b6 b4 0 0 0: 8 bytes 0 0 1: 12 bytes 0 1 0: 16 bytes 0 1 1: 20 bytes 1 0 0: 24 bytes 1 0 1: 32 bytes 1 1 0: 48 bytes 1 1 1: 64 bytes	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b8	MODE	Operation Mode Setting	0: Receive FIFO mode 1: Transmit FIFO mode	R/W
b9	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10	ITCS	Interval Timer Count Source Select	0: Reference clock (× 1 / × 10 period) 1: Bit time clock	R/W
b11	ITR	Interval Timer Resolution Select	0: Reference clock period × 1 1: Reference clock period × 10	R/W
b12	CFIM	Common FIFO Interrupt Mode Setting	Receive FIFO mode: 0: Reception interrupt generated when the number of messages stored in the common FIFO reaches CFITH[2:0] value from a lower value 1: Reception interrupt generated at the end of every received message storage Transmit FIFO mode: 0: Transmission interrupt generated when common FIFO transmits the last message successfully 1: Transmission interrupt generated for every successfully transmitted message	R/W
b15 to b13	CFITH[2:0]	Common FIFO Receive Interrupt Threshold Setting	b15 b13 0 0 0: Interrupt is generated when FIFO is 1/8 full 0 0 1: Interrupt is generated when FIFO is 1/4 full 0 1 0: Interrupt is generated when FIFO is 3/8 full 0 1 1: Interrupt is generated when FIFO is 1/2 full 1 0 0: Interrupt is generated when FIFO is 5/8 full 1 0 1: Interrupt is generated when FIFO is 3/4 full 1 1 0: Interrupt is generated when FIFO is 7/8 full 1 1 1: Interrupt is generated when FIFO is full	R/W

Bit	Symbol	Bit Name	Description	R/W
b17 to b16	LTM[1:0]	Linked Transmit Message Buffer Select	Transmission scan link position of the corresponding channel	R/W
b20 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23 to b21	FDS[2:0]	FIFO Depth Setting	b23 b21 0 0 0: 0 messages 0 0 1: 4 messages 0 1 0: 8 messages 0 1 1: 16 messages 1 0 0: 32 messages 1 0 1: 48 messages Settings other than the above are prohibited.	R/W
b31 to b24	TINT[7:0]	Transmission Interval Setting	Delay the start of transmission from the FIFO if configured in transmit mode, delay is a multiple of basic interval timer clock source unit	R/W

CFE Bit (Common FIFO Enable)

Setting the CFE bit to 1 enables the FIFO. The FIFO is disabled when this bit is set to 0.
This bit can also be used, by clearing it, to abort transmission from common FIFO when configured in transmit FIFO mode, or to stop reception into the common FIFO in receive FIFO mode.
This bit can only be written in GL_HALT or GL_OPERATION mode. Also, if the common FIFO is configured as transmit FIFO mode, this bit can only be written when the CAN channel is not in CH_RESET mode.
This bit can only be set to 1 when the FIFO depth is between 4 and 48 (001b ≤ FDS[2:0] ≤ 101b).
Set the CFE bit to 1 with a separate write access to the CFCR0 register, after setting all the other bits in this register.
This bit is automatically set to 0 when the CANFD module is in GL_RESET mode. Also, if the common FIFO is configured as transmit FIFO mode, this bit is automatically set to 0 when the CAN channel is in CH_RESET mode.

CFRIE Bit (Common FIFO Receive Interrupt Enable)

The CFRIE bit enables or disables generation of common FIFO receive interrupt. If this flag is 1, common FIFO receive interrupt is generated when the common FIFO receive interrupt flag is set to 1 after reception of a frame in the corresponding FIFO buffer.
Do not write to this bit when the CANFD module is in GL_SLEEP mode.

CFTIE Bit (Common FIFO Transmit Interrupt Enable)

The CFTIE bit enables or disables generation of common FIFO transmit interrupt. If this flag is 1, common FIFO transmit interrupt is generated when the common FIFO transmit interrupt flag is set to 1 after transmission of a frame from the corresponding FIFO buffer.
Do not write to this bit when the CANFD module is in GL_SLEEP mode.

PLS[2:0] Bits (Payload Size Setting)

The PLS[2:0] bits define the message data payload size in the RAM. This is the maximum number of bytes which can be received or transmitted by the FIFO buffer.
For details, refer to section 36.6, FIFO Buffers and Message Buffer Configuration.
These bits can only be written in GL_RESET mode.

MODE Bit (Operation Mode Setting)

The MODE bit is used to select the operation mode of the common FIFO. When the MCU reset is applied, all the common FIFO are configured in receive FIFO mode.
This bit cannot be written in GL_OPERATION or GL_SLEEP mode.
Write to this bit only when the CANFD module is in GL_RESET mode.

ITCS Bit (Interval Timer Count Source Select)

The ITCS bit is used to select the count source for the transmission interval timer.
Do not write to this bit when the CANFD module is in GL_SLEEP mode. Also, do not write to this bit when the CFE bit is set to 1.
Do not write 1 to this bit when CAN FD communication is used.

Note: The bit time clock can vary depending on the nominal bit rate and data bit rate configuration.

ITR Bit (Interval Timer Resolution Select)

The ITR bit is used to select the resolution of the reference clock for the transmission interval timer.
This bit cannot be written in GL_SLEEP mode. Do not write to this bit when the CFE bit is set to 1.

CFIM Bit (Common FIFO Interrupt Mode Setting)

The CFIM bit is used to select the condition for the generation of common FIFO interrupts.
This bit cannot be written in GL_SLEEP mode.
Write to this bit only when the CANFD module is in GL_RESET mode.

CFITH[2:0] Bits (Common FIFO Receive Interrupt Threshold Setting)

The CFITH[2:0] bits are used to select the message counter value for the generation of common FIFO interrupts. These values represent fractions of the FIFO depth at which the interrupt is to be generated.
These bits cannot be written in GL_SLEEP mode.
The setting of the CFITH[2:0] bits is restricted by the value of the FDS[2:0] bits. For details, refer to section 36.6.2.1, FIFO Buffers Configuration.
Write to these bits only when the CANFD module is in GL_RESET mode.

LTM[1:0] Bits (Linked Transmit Message Buffer Select)

The LTM[1:0] bits are used to select the number of the transmit message buffer that links common FIFO configured in transmit FIFO mode, for transmission scanning.
These bits cannot be written in GL_OPERATION or GL_SLEEP mode.
Write to these bits only when the CANFD module is in GL_RESET mode.

FDS[2:0] Bits (FIFO Depth Setting)

The FDS[2:0] bits are used to select the FIFO depth in units of the number of messages. If the FIFO depth is configured to 0, the FIFO cannot be used.
These bits can only be written in GL_RESET mode.

TINT[7:0] Bits (Transmission Interval Setting)

The TINT[7:0] bits are used to select the delay in the start of transmission for all messages transmitted from the common FIFO configured in transmit FIFO mode. The delay is an integral multiple of the count source cycle of the interval timer (reference clock period × 1, reference clock period × 10, or bit-time clock period).
These bits cannot be written in GL_SLEEP mode.
Do not write to these bits when the CFE bit is set to 1.
When the GCFG.ITP[15:0] bits are set to 0000h, set the TINT[7:0] bits to 00h.

36.2.28 Common FIFO 0 Status Register (CFSR0)

Address(es): CANFD.CFSR0 000A 8058h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	FLVL[5:0]						—	—	—	CFTIF	CFRIF	LOST	FULL	EMPTY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	EMPTY	Common FIFO Empty Flag	0: Message in common FIFO 1: No message in common FIFO (empty)	R
b1	FULL	Common FIFO Full Flag	0: Common FIFO is not full 1: Common FIFO is full	R
b2	LOST	Message Lost Flag	0: No message lost in common FIFO 1: Common FIFO message lost	R/(W) *1
b3	CFRIF	Common FIFO Receive Interrupt Flag	0: Common FIFO receive interrupt condition is not satisfied 1: Common FIFO receive interrupt condition is satisfied	R/(W) *1
b4	CFTIF	Common FIFO Transmit Interrupt Flag	0: Common FIFO transmit interrupt condition is not satisfied 1: Common FIFO transmit interrupt condition is satisfied	R/(W) *1
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13 to b8	FLVL[5:0]	Common FIFO Fill Level	Indicate the number of messages stored in common FIFO	R
b31 to b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing 1 has no effect. This flag is cleared by writing 0 to it.

This register shows the status of messages stored in the corresponding FIFO buffers.

EMPTY Flag (Common FIFO Empty Flag)

The EMPTY flag is automatically set to 1 when:

- The CPU has read all messages from the FIFO buffer in receive FIFO mode
- All messages have been transmitted from the FIFO buffer in transmit FIFO mode
- The common FIFO is disabled by setting the CFCR0.CFE bit to 0
- The CANFD module is in GL_RESET mode
- The CAN channel transits to CH_RESET mode in transmit FIFO mode.

The EMPTY flag is automatically set to 0 when:

- The first reception message is stored in the FIFO buffer in receive FIFO mode
- The first message to be transmitted is stored in the FIFO buffer in transmit FIFO mode.

FULL Flag (Common FIFO Full Flag)

The FULL flag is automatically set to 1 when the number of messages stored in the FIFO matches the configured FIFO depth.

The FULL flag is automatically set to 0 when:

- The number of messages stored in the FIFO is less than the configured FIFO depth
- The common FIFO is disabled by setting the CFCR0.CFE bit to 0
- The CANFD module is in GL_RESET mode
- The CAN channel transits to CH_RESET mode in transmit FIFO mode.

LOST Flag (Message Lost Flag)

The LOST flag is automatically set to 1 whenever a message is lost due to attempted storage of a new message when FIFO buffer is already full in receive FIFO mode.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
This flag can only be written in GL_HALT or GL_OPERATION mode. Also, if the common FIFO is configured as transmit FIFO mode, this bit can only be written when the CAN channel is not in CH_RESET mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.
The LOST flag is set to 0:

- By writing 0 to it
- When the CANFD module is in GL_RESET mode
- When the CAN channel transits to CH_RESET mode in transmit FIFO mode.

CFRIF Flag (Common FIFO Receive Interrupt Flag)

The CFRIF flag is set to 1 when the configured interrupt condition is satisfied in receive FIFO mode.
The CFRIF flag is not cleared automatically if the common FIFO is disabled.
This flag can only be written in GL_HALT or GL_OPERATION mode. Also, if the common FIFO is configured as transmit FIFO mode, this bit can only be written when the CAN channel is not in CH_RESET mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
The CFRIF flag is set to 0:

- By writing 0 to it
- When the CANFD module is in GL_RESET mode.

CFTIF Flag (Common FIFO Transmit Interrupt Flag)

The CFTIF flag is set to 1 when the configured interrupt condition is satisfied in transmit FIFO mode.
The CFTIF flag is not cleared automatically if the common FIFO is disabled.
This flag can only be written in GL_HALT or GL_OPERATION mode. Also, if the common FIFO is configured as transmit FIFO mode, this bit can only be written when the CAN channel is not in CH_RESET mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
The CFTIF flag is set to 0:

- By writing 0 to it
- When the CANFD module is in GL_RESET mode
- When the CAN channel is in CH_RESET mode.

FLVL[5:0] Bits (Common FIFO Fill Level)

The FLVL[5:0] bits indicate the following:

- The number of messages stored by the CPU and waiting to be transmitted in the transmit FIFO mode

- The number of CPU-readable messages stored by CANFD in the receive FIFO mode.

The FLVL[5:0] bits are automatically set to 000000b when:

- The FIFO is disabled
- The CANFD module is in GL_RESET mode
- The CAN channel transits to CH_RESET mode in the transmit FIFO mode.

36.2.29 Common FIFO 0 Pointer Control Register (CFPCR0)

Address(es): CANFD.CFPCR0 000A 805Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register can be used to increment the read or write pointer of the corresponding common FIFO.
When the value 000000FFh is written into this register, the read pointer of the corresponding common FIFO in receive FIFO mode, or the write pointer of the corresponding common FIFO in transmit FIFO mode moves to the next FIFO entry.
The read value from this register is always 00000000h.
This register can only be written in GL_HALT or GL_OPERATION mode.
Write to this register only when:

- The common FIFO is enabled and is not empty in receive FIFO mode
- The common FIFO is enabled and is not full in transmit FIFO mode.

Do not write to the CFPCR0 register when DTC/DMA transfer is enabled (DTCR.CFDTE0 bit = 1).

36.2.30 FIFO Empty Status Register (FESR)

Address(es): CANFD.FESR 000A 8060h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CFEMP0	—	—	—	—	—	—	RFEMP1	RFEMP0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	RFEMP0	Receive FIFO 0 Empty Flag	0: Message in corresponding receive FIFO 0 1: No message in corresponding receive FIFO 0 (empty)	R
b1	RFEMP1	Receive FIFO 1 Empty Flag	0: Message in corresponding receive FIFO 1 1: No message in corresponding receive FIFO 1 (empty)	R
b7 to b2	—	Reserved	These bits are read as 0	R
b8	CFEMP0	Common FIFO 0 Empty Flag	0: Message in corresponding common FIFO 0 1: No message in corresponding common FIFO 0 (empty)	R
b31 to b9	—	Reserved	These bits are read as 0	R

This register shows the status of the empty flags of the FIFO buffers.

RFEMP0 Flag (Receive FIFO 0 Empty Flag)

The RFEMP0 flag is set to 1 when the RFSR0.EMPTY flag is set to 1, and is set to 0 when the RFSR0.EMPTY flag is set to 0.

The RFEMP0 flag is set to 1 when the CANFD module is in GL_RESET mode.

RFEMP1 Flag (Receive FIFO 1 Empty Flag)

The RFEMP1 flag is set to 1 when the RFSR1.EMPTY flag is set to 1, and is set to 0 when the RFSR1.EMPTY flag is set to 0.

The RFEMP1 flag is set to 1 when the CANFD module is in GL_RESET mode.

CFEMP0 Flag (Common FIFO 0 Empty Flag)

The CFEMP0 flag is set to 1 when the CFSR0.EMPTY flag is set to 1, and is set to 0 when the CFSR0.EMPTY flag is set to 0.

The CFEMP0 flag is set to 1 when the CANFD module is in GL_RESET mode.

36.2.31 FIFO Full Status Register (FFSR)

Address(es): CANFD.FFSR 000A 8064h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CFFUL0	—	—	—	—	—	—	RFFUL1	RFFUL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RFFUL0	Receive FIFO 0 Full Flag	0: Corresponding receive FIFO 0 is not full 1: Corresponding receive FIFO 0 is full	R
b1	RFFUL1	Receive FIFO 1 Full Flag	0: Corresponding receive FIFO 1 is not full 1: Corresponding receive FIFO 1 is full	R
b7 to b2	—	Reserved	These bits are read as 0	R
b8	CFFUL0	Common FIFO 0 Full Flag	0: Common FIFO 0 is not full 1: Common FIFO 0 is full	R
b31 to b9	—	Reserved	These bits are read as 0	R

This register shows the status of the full flags of the FIFO buffers.

RFFUL0 Flag (Receive FIFO 0 Full Flag)

The RFFUL0 flag is set to 1 when the RFSR0.FULL flag is set to 1, and is set to 0 when the RFSR0.FULL flag is set to 0.

The RFFUL0 flag is set to 0 when CANFD module is in GL_RESET mode.

RFFUL1 Flag (Receive FIFO 1 Full Flag)

The RFFUL1 flag is set to 1 when the RFSR1.FULL flag is set to 1, and is set to 0 when the RFSR1.FULL flag is set to 0.

The RFFUL1 flag is set to 0 when CANFD module is in GL_RESET mode.

CFFUL0 Flag (Common FIFO 0 Full Flag)

The CFFUL0 flag is set to 1 when the CFSR0.FULL flag is set to 1 is set to 0 when the CFSR0.FULL flag is set to 0.

The CFFUL0 flag is set to 0 when the CANFD module is in GL_RESET mode.

36.2.32 FIFO Message Lost Status Register (FMLSR)

Address(es): CANFD.FMLSR 000A 8068h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CFML0	—	—	—	—	—	—	RFML1	RFML0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RFML0	Receive FIFO 0 Message Lost Flag	0: Message lost has not occurred in the receive FIFO 0 1: Message lost has occurred in the receive FIFO 0	R
b1	RFML1	Receive FIFO 1 Message Lost Flag	0: Message lost has not occurred in the receive FIFO 1 1: Message lost has occurred in the receive FIFO 1	R
b7 to b2	—	Reserved	These bits are read as 0	R
b8	CFML0	Common FIFO 0 Message Lost Flag	0: Message lost has not occurred in the common FIFO 0 1: Message lost has occurred in the common FIFO 0	R
b31 to b9	—	Reserved	These bits are read as 0	R

This register shows the status of the message lost flags of the FIFO buffers.

RFML0 Flag (Receive FIFO 0 Message Lost Flag)

The RFML0 flag is set to 1 when the RFSR0.LOST flag is set to 1, and is set to 0 when the RFSR0.LOST flag is set to 0. The RFML0 flag is cleared when the CANFD module is in GL_RESET mode.

RFML1 Flag (Receive FIFO 1 Message Lost Flag)

The RFML1 flag is set to 1 when the RFSR1.LOST flag is set to 1, and is set to 0 when the RFSR1.LOST flag is set to 0. The RFML1 flag is cleared when the CANFD module is in GL_RESET mode.

CFML0 Flag (Common FIFO 0 Message Lost Flag)

The CFML0 flag is set to 1 when the CFSR0.LOST flag is set to 1, and is set to 0 when the CFSR0.LOST flag is set to 0. The CFML0 flag is cleared when the CANFD module is in GL_RESET mode.

36.2.33 Receive FIFO Interrupt Status Register (RFISR)

Address(es): CANFD.RFISR 000A 806Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RFIF1	RFIF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RFIF0	Receive FIFO0 Interrupt Flag	0: Receive FIFO 0 interrupt condition is not satisfied 1: Receive FIFO 0 interrupt condition is satisfied	R
b1	RFIF1	Receive FIFO1 Interrupt Flag	0: Receive FIFO 1 interrupt condition is not satisfied 1: Receive FIFO 1 interrupt condition is satisfied	R
b31 to b2	—	Reserved	These bits are read as 0	R

This register shows the status of the interrupt flags of the receive FIFOs.

RFIF0 Flag (Receive FIFO0 Interrupt Flag)

The RFIF0 flag is set to 1 when the RFSR0.RFIF flag is set to 1, and is set to 0 when the RFSR0.RFIF flag is set to 0. The RFIF0 flag is cleared when the CANFD module is in GL_RESET mode.

RFIF1 Flag (Receive FIFO1 Interrupt Flag)

The RFIF1 flag is set to 1 when the RFSR1.RFIF flag is set to 1, and is set to 0 when the RFSR1.RFIF flag is set to 0. The RFIF1 flag is cleared when the CANFD module is in GL_RESET mode.

36.2.34 DMA Transfer Control Register (DTCR)

Address(es): CANFD.DTCR 000A 80C8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CFDTE0	—	—	—	—	—	—	RFDTE1	RFDTE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RFDTE0	Receive FIFO 0 DMA Transfer Enable	0: DTC/DMA transfer request is disabled 1: DTC/DMA transfer request is enabled	R/W
b1	RFDTE1	Receive FIFO 1 DMA Transfer Enable	0: DTC/DMA transfer request is disabled 1: DTC/DMA transfer request is enabled	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	CFDTE0	Common FIFO 0 DMA Transfer Enable	0: DTC/DMA transfer request is disabled 1: DTC/DMA transfer request is enabled	R/W
b31 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register controls the start and stop of DTC/DMA transfer operation.

RFDTE0 Bit (Receive FIFO 0 DMA Transfer Enable)

The RFDTE0 bit enables or disables DTC/DMA transfer request for receive FIFO 0.
This bit cannot be set to 1 in GL_SLEEP or GL_RESET mode.
This bit is set to 0 when the CANFD module is in GL_RESET mode.

RFDTE1 Bit (Receive FIFO 1 DMA Transfer Enable)

The RFDTE1 bit enables or disables DTC/DMA transfer request for receive FIFO 1.
This bit cannot be set to 1 in GL_SLEEP or GL_RESET mode.
This bit is set to 0 when the CANFD module is in GL_RESET mode.

CFDTE0 Bit (Common FIFO 0 DMA Transfer Enable)

The CFDTE0 bit enables or disables DTC/DMA transfer request for common FIFO.
This bit cannot be set to 1 in GL_SLEEP or GL_RESET mode.
Do not enable DTC/DMA transfer for the common FIFO that is configured as transmit FIFO.
This bit is set to 0 when the CANFD module is in GL_RESET mode.

36.2.35 DMA Transfer Status Register (DTSR)

Address(es): CANFD.DTSR 000A 80CCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CFDTS0	—	—	—	—	—	—	RFDTS1	RFDTS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RFDTS0	Receive FIFO 0 DMA Transfer Status Flag	0: DTC/DMA transfer is stopped state 1: DTC/DMA transfer is in progress	R
b1	RFDTS1	Receive FIFO 1 DMA Transfer Status Flag	0: DTC/DMA transfer is stopped state 1: DTC/DMA transfer is in progress	R
b7 to b2	—	Reserved	These bits are read as 0.	R
b8	CFDTS0	Common FIFO 0 DMA Transfer Status Flag	0: DTC/DMA transfer is stopped state 1: DTC/DMA transfer is in progress	R
b31 to b9	—	Reserved	These bits are read as 0.	R

This register shows the status of the DTC/DMA transfer.

RFDTS0 Flag (Receive FIFO 0 DMA Transfer Status Flag)

The RFDTS0 flag is automatically set to 1 when the DTCR.RFDTE0 bit is set to 1 and the receive FIFO 0 is not empty. This flag is automatically set to 0 when the DTC/DMA transfer stops either because the DTCR.RFDTE0 bit is set to 0 or the receive FIFO 0 is empty.

When the DTCR.RFDTE0 bit is set to 0 while DTC/DMA transfer for the receive FIFO 0 is in progress, the RFDTS0 flag becomes 0 when the DTC/DMA transfer is completed.

This flag is set to 0 when the CANFD module is in GL_RESET mode.

RFDTS1 Flag (Receive FIFO 1 DMA Transfer Status Flag)

The RFDTS1 flag is automatically set to 1 when the DTCR.RFDTE1 bit is set to 1 and the receive FIFO 1 is not empty. This flag is automatically set to 0 when the DTC/DMA transfer stops either because the DTCR.RFDTE1 bit is set to 0 or the receive FIFO 1 is empty.

When the DTCR.RFDTE1 bit is set to 0 while DTC/DMA transfer for the receive FIFO 1 is in progress, the RFDTS1 flag becomes 0 when the DTC/DMA transfer is completed.

This flag is set to 0 when the CANFD module is in GL_RESET mode.

CFDTS0 Flag (Common FIFO 0 DMA Transfer Status Flag)

The CFDTS0 flag is automatically set to 1 when the DTCR.CFDTE0 bit is set to 1 and the common FIFO 0 is not empty. This flag is automatically set to 0 when the DTC/DMA transfer stops either because the DTCR.CFDTE0 bit is set to 0 or the common FIFO 0 is empty.

When the DTCR.CFDTE0 bit is set to 0 while DTC/DMA transfer for the common FIFO 0 is in progress, the CFDTS0 flag becomes 0 when the DTC/DMA transfer is complete.

This flag is set to 0 when the CANFD module is in GL_RESET mode.

36.2.36 Transmit Message Buffer n Control Register (TMCRn) (n = 0 to 3)

Address(es): CANFD.TMCR0 000A 8070h, CANFD.TMCR1 000A 8071h, CANFD.TMCR2 000A 8072h, CANFD.TMCR3 000A 8073h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	ONESHOT	TARQ	TXRQ
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TXRQ	Transmission Request	0: Message transmission is not requested 1: Message transmission is requested	R/W
b1	TARQ	Transmission Abort Request	0: Message transmission abort is not requested 1: Message transmission abort is requested	R/W
b2	ONESHOT	One-shot Transmission Enable	0: Transmission in one-shot mode is disabled 1: Transmission in one-shot mode is enabled	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register configures the transmit message buffer functions.

TXRQ Bit (Transmission Request)

When the TXRQ bit is set to 1, the CANFD module logic requests transmitting the message stored in the corresponding message buffer.
This bit can only be written in CH_HALT or CH_OPERATION mode.
This bit cannot be set to 1 if the corresponding transmit message buffer is linked to the common FIFO configured in transmit FIFO or is a part of the transmit queue.
This bit can be set to 1 only when the TMSRn.TXRF[1:0] flags are set to 00b.
This bit cannot be directly cleared by a CPU write access. The TXRQ bit is automatically set to 0:

- At the end of a successful transmission
- At the end of a transmission abort, requested by the TARQ bit
- When a CAN bus error or arbitration lost is detected if the ONESHOT bit is set to 1
- When the CANFD module is in GL_RESET or the CAN channel is in CH_RESET mode.

TARQ Bit (Transmission Abort Request)

When the TARQ bit is set to 1, the CANFD module logic requests aborting the transmission of the frame stored in the corresponding message buffer.
In most cases, transmission cannot be aborted if the internal scan for transmission is complete and the message buffer has already been selected for transmission. In this case, frame will be transmitted successfully from the message buffer. The message buffer selection is released by entering CH_HALT mode.
However, message buffer selected for transmission can be aborted by an abort request when the CAN node detects a new message on the bus (pin for reception) before it starts transmission from the selected message buffer.
Write to the TARQ bit only when the CAN channel is in CH_HALT or CH_OPERATION mode. This bit can be set to 1 only when the TXRQ bit is set to 1.
This bit cannot be set to 0 by a CPU write access. This bit is set to 0 if writing 1 by CPU and clearing by the CAN channel occur at the same time.
The TARQ bit is automatically set to 0:

- At the end of a successful transmission
- At the end of a transmission abort
- When a CAN bus error or arbitration lost is detected

- When the CANFD module is in GL_RESET or the CAN channel enters CH_RESET mode.

ONESHOT Bit (One-shot Transmission Enable)

When the ONESHOT bit is set to 1, the CANFD module logic request transmitting the message only once.
If the transmission is successful, the TMSRn.TXRF[1:0] flags are set to 10b or 11b. Otherwise, the transmission is automatically aborted and TMSRn.TXRF[1:0] flags are set to 01b due to a bus error or an arbitration lost detection.
The ONESHOT bit remains 1 if the transmission has completed successfully or aborted due to a CAN bus error or an arbitration lost detection.
Write to this bit only when the CAN channel is in CH_HALT or CH_OPERATION mode.
Set this bit at the same time as the TXRQ bit. Clear this bit with a write access.
If a message has already been requested for transmission, do not write to this bit until the message has been successfully transmitted or transmission has been aborted.
The ONESHOT bit is automatically set to 0 when the CANFD module is in GL_RESET or the CAN channel is in CH_RESET mode.

36.2.37 Transmit Message Buffer n Status Register (TMSRn) (n = 0 to 3)

Address(es): CANFD.TMSR0 000A 8074h, CANFD.TMSR1 000A 8075h, CANFD.TMSR2 000A 8076h, CANFD.TMSR3 000A 8077h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	TARQS	TXRQS	TXRF[1:0]		TXSF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TXSF	Transmission Status Flag	0: No transmission is in progress 1: Transmission is in progress	R
b2, b1	TXRF[1:0]	Transmission Result Flag	b2 b1 0 0: No result (transmission is not requested or in progress) 0 1: Transmission was aborted. 1 0: Transmission was successful and transmission abort is not requested. 1 1: Transmission was successful and transmission abort is requested.	R/W
b3	TXRQS	Transmission Request Status Flag	0: Transmission is not requested 1: Transmission is requested	R
b4	TARQS	Transmission Abort Request Status Flag	0: Transmission abort is not requested 1: Transmission abort is requested	R
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register shows the status of the transmission and transmission abort for the transmit message buffers.

TXSF Flag (Transmission Status Flag)

The TXSF flag is automatically set to 1 at the start of the transmission from the corresponding transmit message buffer. This flag is automatically set to 0 when:

- Transmission stops
- The CANFD module is in GL_RESET mode
- The CAN channel is in CH_RESET mode.

TXRF[1:0] Flags (Transmission Result Flag)

The TXRF[1:0] flags show the transmission result for the transmit message buffer.
Write to these flags only when the CAN channel is in CH_HALT or CH_OPERATION mode.
Each bit of these flags is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
The TXRF[1:0] flags are automatically set to 00b when the CANFD module is in GL_RESET or the CAN channel is in CH_RESET mode.

TXRQS Flag (Transmission Request Status Flag)

The TXRQS flag reflects the value of the TMCRn.TXRQ bit.
The TXRQS flag is set to 1 when the TMCRn.TXRQ bit is set to 1. This flag is set to 0 when the TMCRn.TXRQ bit is set to 0.

TARQS Flag (Transmission Abort Request Status Flag)

The TARQS flag reflects the value of the TMCRn.TARQ bit.
The TARQS flag is set to 1 when the TMCRn.TARQ bit is set to 1. This flag is set to 0 when the TMCRn.TARQ bit is set to 0.

36.2.38 Transmit Message Buffer Transmission Request Status Register 0 (TMTRSR0)

Address(es): CANFD.TMTRSR0 000A 8078h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	TXRQS3	TXRQS2	TXRQS1	TXRQS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TXRQS0	Transmit Message Buffer 0 Transmission Request Status Flag	0: Transmission is not requested for the transmit message buffer 0. 1: Transmission is requested for the transmit message buffer 0	R
b1	TXRQS1	Transmit Message Buffer 1 Transmission Request Status Flag	0: Transmission is not requested for the transmit message buffer 1. 1: Transmission is requested for the transmit message buffer 1.	R
b2	TXRQS2	Transmit Message Buffer 2 Transmission Request Status Flag	0: Transmission is not requested for the transmit message buffer 2. 1: Transmission is requested for the transmit message buffer 2.	R
b3	TXRQS3	Transmit Message Buffer 3 Transmission Request Status Flag	0: Transmission is not requested for the transmit message buffer 3. 1: Transmission is requested for the transmit message buffer 3.	R
b31 to b4	—	Reserved	These bits are read as 0	R

This register shows the status of the transmission request in each transmit message buffer.

TXRQSn Flag (Transmit Message Buffer n Transmission Request Status Flag) (n = 0 to 3)

The TXRQSn flag indicates status of the TMCRn.TXRQ bit.

Each flag is set to 1 only when the TMCRn.TXRQ bit is set to 1 and the message buffer does not belong to a transmit queue.

Each flag is automatically set to 0 when:

- The TMCRn.TXRQ bit is set to 0
- The CANFD module is in GL_RESET mode
- The CAN channel is in CH_RESET mode.

36.2.39 Transmit Message Buffer Transmission Abort Request Status Register 0 (TMARSR0)

Address(es): CANFD.TMARSR0 000A 807Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	TARQS3	TARQS2	TARQS1	TARQS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TARQS0	Transmit Message Buffer 0 Transmission Abort Request Status Flag	0: Transmission abort is not requested for the transmit message buffer 0 1: Transmission abort is requested for the transmit message buffer 0	R
b1	TARQS1	Transmit Message Buffer 1 Transmission Abort Request Status Flag	0: Transmission abort is not requested for the transmit message buffer 1 1: Transmission abort is requested for the transmit message buffer 1	R
b2	TARQS2	Transmit Message Buffer 2 Transmission Abort Request Status Flag	0: Transmission abort is not requested for the transmit message buffer 2 1: Transmission abort is requested for the transmit message buffer 2	R
b3	TARQS3	Transmit Message Buffer 3 Transmission Abort Request Status Flag	0: Transmission abort is not requested for the transmit message buffer 3 1: Transmission abort is requested for the transmit message buffer 3	R
b31 to b4	—	Reserved	These bits are read as 0	R

This register shows the status of the transmission abort request in each transmit message buffer.

TARQSn Flag (Transmit Message Buffer n Transmission Abort Request Status Flag) (n = 0 to 3)

The TARQSn flag indicates status of the TMCRn.TARQ bit.

Each flag is set to 1 when the TMCRn.TARQ bit is set to 1 or the message buffer belongs to a transmit queue.

Each flag is automatically set to 0 when:

- The TMCRn.TARQ bit is set to 0
- The CANFD module is in GL_RESET mode
- The CAN channel is in CH_RESET mode.

36.2.40 Transmit Message Buffer Transmission Completion Status Register 0 (TMTCSR0)

Address(es): CANFD.TMTCSR0 000A 8080h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	TXCF3	TXCF2	TXCF1	TXCF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TXCF0	Transmit Message Buffer 0 Transmission Complete Flag	0: Transmission is not completed for the transmit message buffer 0 1: Transmission is completed for the transmit message buffer 0	R
b1	TXCF1	Transmit Message Buffer 1 Transmission Complete Flag	0: Transmission is not completed for the transmit message buffer 1 1: Transmission is completed for the transmit message buffer 1	R
b2	TXCF2	Transmit Message Buffer 2 Transmission Complete Flag	0: Transmission is not completed for the transmit message buffer 2 1: Transmission is completed for the transmit message buffer 2	R
b3	TXCF3	Transmit Message Buffer 3 Transmission Complete Flag	0: Transmission is not completed for the transmit message buffer 3 1: Transmission is completed for the transmit message buffer 3	R
b31 to b4	—	Reserved	These bits are read as 0	R

This register shows the transmission completion status for each transmit message buffer.

TXCFn Flag (Transmit Message Buffer n Transmission Completion Status) (n = 0 to 3)

The TXCFn flag indicates the transmission completion status for the transmit message buffer n.

Each flag is automatically set to 1 when the TMSRn.TXRF[1] bit is set to 1.

Each flag is automatically set to 0 when:

- The TMSRn.TXRF[1] bit is set to 0
- The CANFD module is in GL_RESET mode
- The CAN channel is in CH_RESET mode.

36.2.41 Transmit Message Buffer Transmission Abort Status Register 0 (TMTASR0)

Address(es): CANFD.TMTASR0 000A 8084h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	TAF3	TAF2	TAF1	TAF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TAF0	Transmit Message Buffer 0 Transmission Abort Status Flag	0: Transmission is not aborted for the transmit message buffer 0. 1: Transmission is aborted for the transmit message buffer 0.	R
b1	TAF1	Transmit Message Buffer 1 Transmission Abort Status Flag	0: Transmission is not aborted for the transmit message buffer 1. 1: Transmission is aborted for the transmit message buffer 1.	R
b2	TAF2	Transmit Message Buffer 2 Transmission Abort Status Flag	0: Transmission is not aborted for the transmit message buffer 2. 1: Transmission is aborted for the transmit message buffer 2.	R
b3	TAF3	Transmit Message Buffer 3 Transmission Abort Status Flag	0: Transmission is not aborted for the transmit message buffer 3. 1: Transmission is aborted for the transmit message buffer 3.	R
b31 to b4	—	Reserved	These bits are read as 0	R

This register shows the transmission abort status for each transmit message buffer.

TAFn Flag (Transmit Message Buffer n Transmission Abort Status Flag) (n = 0 to 3)

The TAFn flag indicates the transmission abort status for the transmit message buffer n.

Each flag is automatically set to 1 when the TMSRn.TXRF[1:0] flags are set to 01b.

Each flag is automatically set to 0 when:

- The TMSRn.TXRF[1:0] flags are set to 00b
- The CANFD module is in GL_RESET mode
- The CAN channel is in CH_RESET mode.

36.2.42 Transmit Message Buffer Interrupt Enable Register (TMIER0)

Address(es): CANFD.TMIER0 000A 8088h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	TMIE3	TMIE2	TMIE1	TMIE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TMIE0	Transmit Message Buffer 0 Interrupt Enable	0: Interrupt for transmit message buffer 0 is disabled. 1: Interrupt for transmit message buffer 0 is enabled.	R/W
b1	TMIE1	Transmit Message Buffer 1 Interrupt Enable	0: Interrupt for transmit message buffer 1 is disabled. 1: Interrupt for transmit message buffer 1 is enabled.	R/W
b2	TMIE2	Transmit Message Buffer 2 Interrupt Enable	0: Interrupt for transmit message buffer 2 is disabled. 1: Interrupt for transmit message buffer 2 is enabled.	R/W
b3	TMIE3	Transmit Message Buffer 3 Interrupt Enable	0: Interrupt for transmit message buffer 3 is disabled. 1: Interrupt for transmit message buffer 3 is enabled.	R/W
b31 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register enables or disables the interrupt for each transmit message buffer.

TMIEn Bit (Transmit Message Buffer n Interrupt Enable) (n = 0 to 3)

If the TMIEn bit is set to 1, an interrupt is generated when transmission from the transmit message buffer n is completed successfully.

Refer to section 36.10, Interrupts and DTC/DMA Requests for the specification of the interrupt for transmit message buffer.

This bit cannot be written when the CANFD module is in GL_SLEEP mode.

Do not write to the TMIEn bit when:

- The CAN channel is in CH_SLEEP mode
- The transmit message buffer n is part of a transmit queue
- The transmit message buffer n is linked to a common FIFO by the CFCR0.LTM[1:0] bits.

36.2.43 Transmit Queue 0 Configuration Register (TQCR0)

Address(es): CANFD0.TQCR0 000A 808Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	QDS[1:0]		TQIM	—	TQIE	—	—	—	—	TQE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TQE	Transmit Queue Enable	0: Transmit queue is disabled. 1: Transmit queue is enabled.	R/W
b4 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	TQIE	Transmit Queue Interrupt Enable	0: Transmit queue transmission interrupt is disabled. 1: Transmit queue transmission interrupt is enabled.	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	TQIM	Transmit Queue Interrupt Mode Setting	0: When the last message is successfully transmitted 1: At every successful transmission	R/W
b9, b8	QDS[1:0]	Queue Depth Setting	b9 b8 0 0: 0 messages (disabled) 0 1: Setting prohibited 1 0: 3 messages (transmit message buffer 0 to 2) 1 1: 4 messages (transmit message buffer 0 to 3)	R/W
b31 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to configure the transmit queue 0.

TQE Bit (Transmit Queue Enable)

The TQE bit enables the transmit queue. When the TQE bit is set to 1, the transmit message buffer is used to configure the transmit queue.

The TQE bit cannot be set to 1 if the configured transmit queue depth is 0 (QDS[1:0] = 00b).

This bit cannot be written in GL_SLEEP mode.

Also, this bit cannot be written in CH_RESET or CH_SLEEP mode.

The TQE bit is automatically set to 0 when the CAN channel is in CH_RESET mode.

TQIE Bit (Transmit Queue Interrupt Enable)

When the TQIE bit is set to 1, an interrupt is generated based on the setting of the TQIM bit.

This bit cannot be written in GL_SLEEP mode.

Do not write to this bit when the CAN channel is in CH_SLEEP mode.

TQIM Bit (Transmit Queue Interrupt Mode Setting)

The TQIM bit selects the interrupt generation condition for the transmit queue.

This bit cannot be written in GL_SLEEP mode.

Do not write to this bit when the CAN channel is in CH_SLEEP, CH_HALT, or CH_OPERATION mode.

QDS[1:0] Bits (Queue Depth Setting)

The QDS[1:0] bits select the depth of the transmission queue.

When these bits are set to 10b, the message buffers 0 to 2 are used, and when these bits are set to 11b, the message buffers 0 to 3 are used.
These bits cannot be written in GL_SLEEP mode.
Also, these bits cannot be written in CH_HALT or CH_OPERATION mode.
Do not write to these bits when the CAN channel is in CH_SLEEP mode.

36.2.44 Transmit Queue 0 Status Register (TQSR0)

Address(es): CANFD0.TQSR0 000A 8090h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	FLVL[2:0]			—	—	—	—	—	TQIF	FULL	EMPTY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	EMPTY	Transmit Queue Empty Flag	0: There is at least one message in transmit queue 1: No message in transmit queue (empty)	R
b1	FULL	Transmit Queue Full Flag	0: Transmit queue is not full 1: Transmit queue is full	R
b2	TQIF	Transmit Queue Interrupt Flag	0: Transmit queue interrupt condition is not satisfied 1: Transmit queue interrupt condition is satisfied	R/(W) *1
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b10 to b8	FLVL[2:0]	Transmit Queue Fill Level	Indicates the number of messages in the transmit queue	R
b31 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing 1 has no effect. This flag is cleared by writing 0 to it.

This register shows the status of the transmit queue 0.

EMPTY Flag (Transmit Queue Empty Flag)

This flag is automatically set to 1 when:

- The TQCR0.TQE bit is set to 0 (transmit queue is disabled)
- No messages are stored in the transmit queue
- The last message is transmitted from the transmit queue
- The CAN channel is in CH_RESET mode.

This flag is automatically set to 0 when the first message to be transmitted is stored in the transmit queue.

FULL Flag (Transmit Queue Full Flag)

The FULL flag is automatically set to 1 when the number of messages stored in the transmit queue matches the configured transmit queue depth.

This flag is automatically set to 0 when:

- The number of messages stored in the transmit queue is less than the configured transmit queue depth
- The CAN channel is in CH_RESET mode.

TQIF Flag (Transmit Queue Interrupt Flag)

The TQIF flag is not cleared automatically if the transmit queue is disabled.

When stopping the transmit queue, this flag should be cleared, after setting the TQCR0.TQE bit to 0 and checking an empty state of transmit queue.

Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.

This flag is automatically set to 1 when the configured interrupt condition is satisfied for the transmit queue.

This flag is set to 0:

- By writing 0 to it
- When the CAN channel is in CH_RESET mode.

This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
This flag cannot be written when the CAN channel is in CH_SLEEP or CH_RESET mode.

FLVL[2:0] Bits (Transmit Queue Fill Level)

The FLVL[2:0] bits indicates the number of messages in the transmit queue.
These bits are automatically set to 000b when the CAN channel is in CH_RESET mode.

36.2.45 Transmit Queue 0 Pointer Control Register (TQPCR0)

Address(es): CANFD0.TQPCR0 000A 8094h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is used to increment the write pointer to the transmit queue.
When the value 000000FFh is written to this register, the write pointer to the transmit queue is updated and transmission of the existing message is requested.
The read value from this register is always 00000000h.
This register cannot be written when the CAN channel is in CH_SLEEP or CH_RESET mode.
Write to this register only when:

- The transmit queue is enabled and not full.

36.2.46 Transmission History Configuration Register (THCR)

Address(es): CANFD0.THCR 000A 8098h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	THRC	THIM	THIE	—	—	—	—	—	—	—	THE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	THE	Transmission History Enable	0: Transmission history buffer is disabled 1: Transmission history buffer is enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	THIE	Transmission History Interrupt Enable	0: Transmission history interrupt is disabled 1: Transmission history interrupt is enabled	R/W
b9	THIM	Transmission History Interrupt Mode Setting	0: Interrupt is generated if transmission history level reaches 3/4 of the transmission history depth 1: Interrupt is generated for every successfully stored entry	R/W
b10	THRC	Transmission History Recording Condition Setting	0: Transmit FIFO + Transmit queue 1: Transmit message buffer + Transmit FIFO + Transmit queue	R/W
b31 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register configures the transmission history functions.

THE Bit (Transmission History Enable)

The THE bit enables the transmission history buffer when it is set to 1.
This bit cannot be written when the CAN channel is in CH_RESET or CH_SLEEP mode.
This bit is automatically set to 0 when the CAN channel is in CH_RESET mode.

THIE Bit (Transmission History Interrupt Enable)

The THIE bit enables the generation of the transmission history interrupt when it is set to 1.
This bit cannot be written when the CANFD module is in GL_SLEEP mode.

THIM Bit (Transmission History Interrupt Mode Setting)

The THIM bit selects the transmission history interrupt generation condition.
This bit cannot be written when the CANFD module is in GL_SLEEP mode.
Do not write to this bit when the CANFD module is in GL_HALT or GL_OPERATION mode.

THRC Bit (Transmission History Recording Condition Setting)

The THRC bit selects the condition for storing an entry in the transmission history buffer after successful transmission.
This bit cannot be written when the CANFD module is in GL_SLEEP mode.
Do not write to this bit when the CANFD module is in GL_HALT or GL_OPERATION mode.

36.2.47 Transmission History Status Register (THSR)

Address(es): CANFD0.THSR 000A 809Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	FLVL[3:0]				—	—	—	—	THIF	LOST	FULL	EMPTY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	EMPTY	Transmission History Empty Flag	0: There is a transmission history in the transmission history buffer 1: There is no transmission history in the transmission history buffer (empty)	R
b1	FULL	Transmission History Full Flag	0: Transmission history buffer is not full 1: Transmission history buffer is full	R
b2	LOST	Transmission History Lost Flag	0: Transmission history has not been lost 1: Transmission history has been lost	R/(W) *1
b3	THIF	Transmission History Interrupt Flag	0: Transmission history interrupt condition is not satisfied 1: Transmission history interrupt condition is satisfied	R/(W) *1
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11 to b8	FLVL[3:0]	Transmission History Fill Level	Number of transmission histories stored in transmission history buffer	R
b31 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing 1 has no effect. This flag is cleared by writing 0 to it.

This register shows the status of data stored in the transmission history buffer.

EMPTY Flag (Transmission History Empty Flag)

This flag is automatically set to 0 when the first transmission history is stored to the transmission history buffer.
This flag is automatically set to 1 when:

- The CPU has read all the transmission histories from the transmission history buffer
- The THCR.THE bit is set to 0 (transmission history buffer is disabled)
- The CAN channel is in CH_RESET mode.

FULL Flag (Transmission History Full Flag)

The FULL flag is automatically set to 1 when the number of transmission histories in the transmission history buffer is eight.
This flag is automatically set to 0 when:

- The number of transmission histories in the transmission history buffer is less than eight
- The THCR.THE bit is set to 0 (transmission history buffer is disabled)
- The CAN channel is in CH_RESET mode.

LOST Flag (Transmission History Lost Flag)

The LOST flag is set to 1 when a new transmission history cannot be stored because the related transmission history buffer is already full.

Write to this flag only when the CAN channel is in CH_HALT or CH_OPERATION mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
This flag is set to 0:

- By writing 0 to it
- When the CAN channel is in CH_RESET mode.

THIF Flag (Transmission History Interrupt Flag)

The THIF flag is set to 1 when the configured interrupt condition is satisfied.
Write to this flag only when the CAN channel is in CH_HALT or CH_OPERATION mode.
Do not use the bit clear instruction to clear the flag. Set only the flags to be cleared to 0 and the other flags to 1, and write the data in 32-bit units using MOV instruction.
This flag is set to 1 if clearing by write access and setting by the CAN channel occur at the same time.
This flag is set to 0:

- By writing 0 to it
- When the CAN channel is in CH_RESET mode.

FLVL[3:0] Bits (Transmission History Fill Level)

The FLVL[3:0] bits show the number of transmission histories stored in the transmission history buffer.
These bits are automatically set to 0000b when the CAN channel is in CH_RESET mode.

36.2.48 Transmission History Access Register 0 (THACR0)

Address(es): CANFD0.THACR0 000A 8740h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TS[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	BN[1:0]		BT[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	BT[2:0]	Transmitted Buffer Type	b2 b0 0 0 1: Transmit message buffer 0 1 0: Common FIFO 1 0 0: Transmit Queue	R
b4, b3	BN[1:0]	Transmitted Buffer Number	Number of the message buffer	R
b15 to b5	—	Reserved	These bits are read as 0	R
b31 to b16	TS[15:0]	Transmitted Timestamp	Transmit timestamp value	R

This register is used to access to the histories in the transmission history buffer based on the read pointer value.

BT[2:0] Bits (Transmitted Buffer Type)

The BT[2:0] bits indicate which type of buffer the read history came from.

BN[1:0] Bits (Transmitted Buffer Number)

The BN[1:0] bits indicate which buffer number the read history came from.
For a common FIFO, these bits indicate the number of the linked transmit message buffer.

TS[15:0] Bits (Transmitted Timestamp)

The TS[15:0] bits indicate the timestamp used by the software driver.

36.2.49 Transmission History Access Register 1 (THACR1)

Address(es): CANFD0.THACR1 000A 8744h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IFL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PTR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	PTR[15:0]	Transmitted Pointer	Stores the value of the PTR[15:0] field attached to the transmit message	R
b17, b16	IFL[1:0]	Transmitted Information Label	Stores the value of the IFL[1:0] field attached to the transmit message	R
b31 to b18	—	Reserved	These bits are read as 0	R

This register is used to access to the histories in the transmission history buffer based on the read pointer value.

PTR[15:0] Bits (Transmitted Pointer)

The PTR[15:0] bits store the value of the pointer field (TMBn.HF2.PTR[15:0] bits or CFB0.HF2.PTR[15:0] bits) attached to the transmit message.

IFL[1:0] Bits (Transmitted Information Label)

The IFL[1:0] bits store the value of the information label field (TMBn.HF2.IFL[1:0] bits or CFB0.HF2.IFL[1:0] bits) attached to the transmit message.

36.2.50 Transmission History Pointer Control Register (THPCR)

Address(es): CANFD0.THPCR 000A 80A0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is used to increment the read pointer of the transmission history buffer.
When 000000FFh is written to this register, the read pointer of the transmission history buffer is moved to the next history.
The read value from this register is 00000000h.
This register can only be written in CH_HALT or CH_OPERATION mode.
Write 000000FFh to this register only when the transmission history buffer is enabled and not empty.

36.2.51 Global Reset Control Register (GRCR)

Address(es): CANFD.GRCR 000A 80D8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								—	—	—	—	—	—	—	SRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SRST	Software Reset	0: Releases the software reset 1: Initiates the software reset	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	Key Code	These bits control rewriting of the SRST bit. These bits are read as 00h.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SRST Bit (Software Reset)

When the SRST bit is set to 1, the CANFD module is in the same state as the MCU reset. When a reset is required, write 1 to this bit and then 0.

After releasing the software reset, the CANFD module is in GL_SLEEP mode.

After a software reset, the RAM initialization sequence is not performed. Initialize the RAM with software.

Similarly, if a software reset is performed during RAM initialization, the RAM is not initialized. Initialize the RAM with software.

KEY[7:0] Bits (Key Code)

When rewriting the value of the SRST bit, set these bits to C4h at the same time (write this register in 32 bits).

36.2.52 Global Test Mode Configuration Register (GTMCR)

Address(es): CANFD.GTMCR 000A 80A8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	RTPS[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b19 to b16	RTPS[3:0]	RAM Test Page Select	Select the page of RAM to be tested	R/W
b31 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to specify the page number of the RAM to be tested in RAM test mode.

RTPS[3:0] Bits (RAM Test Page Select)

The RTPS[3:0] bits select the RAM page number that the CPU reads and writes when the CANFD module is in RAM test mode. Specify the page number in the range of 0 to 9.

Refer to section 36.9.2.1, RAM Test Mode for the specifications of the RAM test mode.

These bits cannot be written in GL_RESET or GL_SLEEP mode. Write to these bits only in GL_HALT mode.

These bits are automatically set to 0000b when the CAN channel is in GL_RESET mode.

36.2.53 Global Test Mode Enable Register (GTMER)

Address(es): CANFD.GTMER 000A 80ACh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	RTME	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b2	RTME	RAM Test Mode Enable	0: RAM Test mode is disabled. 1: RAM Test mode is enabled.	R/W
b31 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to control the global test modes of the CANFD module.

RTME Bit (RAM Test Mode Enable)

When the RTME bit is set to 1, the CANFD module is in RAM test mode. Refer to section 36.9.2.1, RAM Test Mode for the specifications of the RAM test mode.

This bit can only be set to 1 when the CANFD module is in GL_HALT mode. To exit the RAM test mode, set this bit to 0 in GL_HALT mode.

This bit is automatically set to 0 when the CANFD module is in GL_RESET mode.

36.2.54 Global CAN FD Configuration Register (GFDCFG)

Address(es): CANFD.GFDCFG 000A 80B0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	TSCPS[1:0]		—	—	—	—	—	—	—	PXEDIS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PXEDIS	Protocol Exception Event Detection Disable	0: Protocol exception event detection is enabled. 1: Protocol exception event detection is disabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	TSCPS[1:0]	Timestamp Capturing Point Select	b9 b8 0 0: Sample point of SOF (start of frame) 0 1: EOF (end of frame) when frame is taken valid 1 0: Sample point of SOF (Classical CAN frame), or sample point of res bit following the FDF bit (CAN FD frame) 1 1: Setting prohibited	R/W
b31 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

PXEDIS Bit (Protocol Exception Event Detection Disable)

The PXEDIS bit configures the protocol exception event handling according to ISO 11898-1.

When this bit is set to 1, protocol exception event detection is disabled, and an error frame is transmitted if a protocol exception (a recessive reserved bit following the FDF bit) is detected.

This bit can only be written in GL_RESET mode.

TSCPS[1:0] Bits (Timestamp Capturing Point Select)

The TSCPS[1:0] bits selects the capture point of the timestamps for transmitted and received messages.

When the TSCPS[1:0] bits are 10b, the timestamp is captured at the reserved bit following the FDF bit in CAN FD frames and at the SOF in Classical CAN frames.

These bits can only be written in GL_RESET mode.

36.2.55 Global Test Mode Lock Key Register (GTMLKR)

Address(es): CANFD.GTMLKR 000A 80B8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is used to unlock the protection for RAM test mode. Refer to section 36.9.2, Global Test Modes for the specification of the lock key.
To put the CANFD module into RAM test mode, two unlock keys must be written to this register in consecutive bus cycles.
The read value from this register is always 00000000h.
This register cannot be written when the CANFD module is in GL_SLEEP or GL_RESET mode.
Do not write to this register when the CANFD module is in GL_OPERATION mode.

36.2.56 RAM Test Page Access Register k (RTPARk) (k = 0 to 63)

Address(es): CANFD.RTPAR0 000A 8280h to CANFD.RTPAR63 000A 837Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register can be read or written when the CANFD module is in RAM test mode.
This register can only be written in GL_HALT mode when RAM test mode is enabled.

36.2.57 Acceptance Filter List Ignore Entry Setting Register (AFIGSR)

Address(es): CANFD.AFIGSR 000A 80C0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	IGES[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	IGES[4:0]	Ignore Entry Select	Set the rule number to be ignored during acceptance filtering.	R/W
b31 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

IGES[4:0] Bits (Ignore Entry Select)

The IGES[4:0] bits are used to specify the rule number to update when updating the acceptance filter.

Write to these bits only when the AFIGER.IGEE bit is 0.
These bits cannot be written in GL_SLEEP mode.

36.2.58 Acceptance Filter List Ignore Entry Enable Register (AFIGER)

Address(es): CANFD.AFIGER 000A 80C4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								—	—	—	—	—	—	—	IGEE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IGEE	Ignore Entry Enable	0: The value of the AFIGSR.IGES[4:0] bits is invalid. 1: The value of the AFIGSR.IGES[4:0] bits is valid.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	Key Code	Controls the validity of rewriting the IGEE bit	W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

IGEE Bit (Ignore Entry Enable)

When the IGEE bit is set to 1, the entry selected by the AFIGSR.IGES[4:0] bits is ignored.
This bit is automatically set to 0 when the CANFD module is in GL_RESET mode.

KEY[7:0] Bits (Key Code)

Writing to the IGEE bit is enabled when C4h is written to the KEY[7:0] bits. The value read from these bits is always 00h.
Write the IGEE bit and the KEY [7:0] bits at the same time.

36.2.59 Receive Message Buffer Interrupt Enable Register (RMIER)

Address(es): CANFD.RMIER 000A 8038h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RMIE31	RMIE30	RMIE29	RMIE28	RMIE27	RMIE26	RMIE25	RMIE24	RMIE23	RMIE22	RMIE21	RMIE20	RMIE19	RMIE18	RMIE17	RMIE16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RMIE15	RMIE14	RMIE13	RMIE12	RMIE11	RMIE10	RMIE9	RMIE8	RMIE7	RMIE6	RMIE5	RMIE4	RMIE3	RMIE2	RMIE1	RMIE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RMIE0	Receive Message Buffer 0 Interrupt Enable	0: Interrupt for receive message buffer 0 is disabled. 1: Interrupt for receive message buffer 0 is enabled.	R/W
b1	RMIE1	Receive Message Buffer 1 Interrupt Enable	0: Interrupt for receive message buffer 1 is disabled. 1: Interrupt for receive message buffer 1 is enabled.	R/W
b2	RMIE2	Receive Message Buffer 2 Interrupt Enable	0: Interrupt for receive message buffer 2 is disabled. 1: Interrupt for receive message buffer 2 is enabled.	R/W
b3	RMIE3	Receive Message Buffer 3 Interrupt Enable	0: Interrupt for receive message buffer 3 is disabled. 1: Interrupt for receive message buffer 3 is enabled.	R/W
b4	RMIE4	Receive Message Buffer 4 Interrupt Enable	0: Interrupt for receive message buffer 4 is disabled. 1: Interrupt for receive message buffer 4 is enabled.	R/W
b5	RMIE5	Receive Message Buffer 5 Interrupt Enable	0: Interrupt for receive message buffer 5 is disabled. 1: Interrupt for receive message buffer 5 is enabled.	R/W
b6	RMIE6	Receive Message Buffer 6 Interrupt Enable	0: Interrupt for receive message buffer 6 is disabled. 1: Interrupt for receive message buffer 6 is enabled.	R/W
b7	RMIE7	Receive Message Buffer 7 Interrupt Enable	0: Interrupt for receive message buffer 7 is disabled. 1: Interrupt for receive message buffer 7 is enabled.	R/W
b8	RMIE8	Receive Message Buffer 8 Interrupt Enable	0: Interrupt for receive message buffer 8 is disabled. 1: Interrupt for receive message buffer 8 is enabled.	R/W
b9	RMIE9	Receive Message Buffer 9 Interrupt Enable	0: Interrupt for receive message buffer 9 is disabled. 1: Interrupt for receive message buffer 9 is enabled.	R/W
b10	RMIE10	Receive Message Buffer 10 Interrupt Enable	0: Interrupt for receive message buffer 10 is disabled. 1: Interrupt for receive message buffer 10 is enabled.	R/W
b11	RMIE11	Receive Message Buffer 11 Interrupt Enable	0: Interrupt for receive message buffer 11 is disabled. 1: Interrupt for receive message buffer 11 is enabled.	R/W
b12	RMIE12	Receive Message Buffer 12 Interrupt Enable	0: Interrupt for receive message buffer 12 is disabled. 1: Interrupt for receive message buffer 12 is enabled.	R/W
b13	RMIE13	Receive Message Buffer 13 Interrupt Enable	0: Interrupt for receive message buffer 13 is disabled. 1: Interrupt for receive message buffer 13 is enabled.	R/W
b14	RMIE14	Receive Message Buffer 14 Interrupt Enable	0: Interrupt for receive message buffer 14 is disabled. 1: Interrupt for receive message buffer 14 is enabled.	R/W
b15	RMIE15	Receive Message Buffer 15 Interrupt Enable	0: Interrupt for receive message buffer 15 is disabled. 1: Interrupt for receive message buffer 15 is enabled.	R/W
b16	RMIE16	Receive Message Buffer 16 Interrupt Enable	0: Interrupt for receive message buffer 16 is disabled. 1: Interrupt for receive message buffer 16 is enabled.	R/W
b17	RMIE17	Receive Message Buffer 17 Interrupt Enable	0: Interrupt for receive message buffer 17 is disabled. 1: Interrupt for receive message buffer 17 is enabled.	R/W
b18	RMIE18	Receive Message Buffer 18 Interrupt Enable	0: Interrupt for receive message buffer 18 is disabled. 1: Interrupt for receive message buffer 18 is enabled.	R/W
b19	RMIE19	Receive Message Buffer 19 Interrupt Enable	0: Interrupt for receive message buffer 19 is disabled. 1: Interrupt for receive message buffer 19 is enabled.	R/W

Bit	Symbol	Bit Name	Description	R/W
b20	RMIE20	Receive Message Buffer 20 Interrupt Enable	0: Interrupt for receive message buffer 20 is disabled. 1: Interrupt for receive message buffer 20 is enabled.	R/W
b21	RMIE21	Receive Message Buffer 21 Interrupt Enable	0: Interrupt for receive message buffer 21 is disabled. 1: Interrupt for receive message buffer 21 is enabled.	R/W
b22	RMIE22	Receive Message Buffer 22 Interrupt Enable	0: Interrupt for receive message buffer 22 is disabled. 1: Interrupt for receive message buffer 22 is enabled.	R/W
b23	RMIE23	Receive Message Buffer 23 Interrupt Enable	0: Interrupt for receive message buffer 23 is disabled. 1: Interrupt for receive message buffer 23 is enabled.	R/W
b24	RMIE24	Receive Message Buffer 24 Interrupt Enable	0: Interrupt for receive message buffer 24 is disabled. 1: Interrupt for receive message buffer 24 is enabled.	R/W
b25	RMIE25	Receive Message Buffer 25 Interrupt Enable	0: Interrupt for receive message buffer 25 is disabled. 1: Interrupt for receive message buffer 25 is enabled.	R/W
b26	RMIE26	Receive Message Buffer 26 Interrupt Enable	0: Interrupt for receive message buffer 26 is disabled. 1: Interrupt for receive message buffer 26 is enabled.	R/W
b27	RMIE27	Receive Message Buffer 27 Interrupt Enable	0: Interrupt for receive message buffer 27 is disabled. 1: Interrupt for receive message buffer 27 is enabled.	R/W
b28	RMIE28	Receive Message Buffer 28 Interrupt Enable	0: Interrupt for receive message buffer 28 is disabled. 1: Interrupt for receive message buffer 28 is enabled.	R/W
b29	RMIE29	Receive Message Buffer 29 Interrupt Enable	0: Interrupt for receive message buffer 29 is disabled. 1: Interrupt for receive message buffer 29 is enabled.	R/W
b30	RMIE30	Receive Message Buffer 30 Interrupt Enable	0: Interrupt for receive message buffer 30 is disabled. 1: Interrupt for receive message buffer 30 is enabled.	R/W
b31	RMIE31	Receive Message Buffer 31 Interrupt Enable	0: Interrupt for receive message buffer 31 is disabled. 1: Interrupt for receive message buffer 31 is enabled.	R/W

This register enables or disables the interrupt for each receive message buffer.

RMIEn Bit (Receive Message Buffer n Interrupt Enable) (n = 0 to 31)

If the RMIEn bit is set to 1, an interrupt is generated when reception to the receive message buffer n is completed successfully.

Refer to section 36.10, Interrupts and DTC/DMA Requests for the specification of the receive message buffer interrupt.

This bit cannot be written when the CANFD module is in GL_SLEEP mode.

36.2.60 Identifier Bits Alignment

Base format (11-bit length identifier): ID-28 to ID-18 are located in b10 to b0, and b28 to b11 are 0.
Extended format (29-bit length identifier): ID-28 to ID-0 are located in b28 to b0

Table 36.4 Standard Identifier (11-bit format)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
IDE = 0	RTR	—	0	0	0	0	0	0	0	0	0	0	0	0	0

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21	ID-20	ID-19	ID-18

Table 36.5 Extended Identifier (29-bit format)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
IDE = 1	RTR	—	ID-28	ID-27	ID-26	ID-25	ID-24	ID-23	ID-22	ID-21	ID-20	ID-19	ID-18	ID-17	ID-16

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ID-15	ID-14	ID-13	ID-12	ID-11	ID-10	ID-9	ID-8	ID-7	ID-6	ID-5	ID-4	ID-3	ID-2	ID-1	ID-0

36.2.61 Message Buffer Structure

The message buffer configuration consists of four types of message buffer:

- Receive message buffer (RMBn)
- Receive FIFO (RFBn)
- Common FIFO (CFB0)
- Transmit message buffer (TMBn)

n is a message buffer number whose range changes depending on the type of message buffer.
Refer to Figure 36.33 for an overview of this configuration. For more information on the number and types of message buffers, refer to section 36.6, FIFO Buffers and Message Buffer Configuration.

36.2.61.1 Start Addresses

The start address of each message buffer is calculated using the number of related message buffer components.
The start addresses for each register in the message buffer are listed in Table 36.6.

Table 36.6 Start Address for Each Register of the Message Buffer Component

Message Buffer	Symbol	n	Register	p	Start Address
Receive Message Buffer	RMBn	0 to 7	HF0	—	000A 8920h + n × 4Ch
			HF1	—	000A 8924h + n × 4Ch
			HF2	—	000A 8928h + n × 4Ch
			DFp	0 to 15	000A 892Ch + n × 4Ch + p × 4
	RMBn	8 to 15	HF0	—	000A 8D20h + (n – 8) × 4Ch
			HF1	—	000A 8D24h + (n – 8) × 4Ch
			HF2	—	000A 8D28h + (n – 8) × 4Ch
			DFp	0 to 15	000A 8D2Ch + (n – 8) × 4Ch + p × 4
	RMBn	16 to 23	HF0	—	000A 9120h + (n – 16) × 4Ch
			HF1	—	000A 9124h + (n – 16) × 4Ch
			HF2	—	000A 9128h + (n – 16) × 4Ch
			DFp	0 to 15	000A 912Ch + (n – 16) × 4Ch + p × 4
	RMBn	24 to 31	HF0	—	000A 9520h + (n – 24) × 4Ch
			HF1	—	000A 9524h + (n – 24) × 4Ch
			HF2	—	000A 9528h + (n – 24) × 4Ch
			DFp	0 to 15	000A 952Ch + (n – 24) × 4Ch + p × 4
Receive FIFO	RFBn	0, 1	HF0	—	000A 8520h + n × 4Ch
			HF1	—	000A 8524h + n × 4Ch
			HF2	—	000A 8528h + n × 4Ch
			DFp	0 to 15	000A 852Ch + n × 4Ch + p × 4
Common FIFO	CFB0	0	HF0	—	000A 85B8h
			HF1	—	000A 85BCh
			HF2	—	000A 85C0h
			DFp	0 to 15	000A 85C4h + p × 4
Transmit FIFO Buffer	TMBn	0 to 3	HF0	—	000A 8604h + n × 4Ch
			HF1	—	000A 8608h + n × 4Ch
			HF2	—	000A 860Ch + n × 4Ch
			DFp	0 to 15	000A 8610h + n × 4Ch + p × 4

36.2.61.2 Receive Message Buffer n (RMBn) (n = 0 to 31)

The total number of the receive message buffer (RMB) is 32, as shown in Figure 36.33.
The receive message buffer consists of the following registers:

- RMBn.HF0
- RMBn.HF1
- RMBn.HF2
- RMBn.DF0 to RMBn.DF15

Table 36.7 shows the structure of this buffer.

Table 36.7 Structure of Receive Message Buffer

Address Offset	Symbol	Register Name	Contents
+00h	RMBn.HF0	Receive Message Buffer n Header Field 0	IDE, RTR, ID
+04h	RMBn.HF1	Receive Message Buffer n Header Field 1	DLC, timestamp
+08h	RMBn.HF2	Receive Message Buffer n Header Field 2	Pointer, information label, FDF, BRS, ESI
+0Ch	RMBn.DF0	Receive Message Buffer n Data Field 0	DATA0 to DATA3
+10h	RMBn.DF1	Receive Message Buffer n Data Field 1	DATA4 to DATA7
+14h	RMBn.DF2	Receive Message Buffer n Data Field 2	DATA8 to DATA11
+18h	RMBn.DF3	Receive Message Buffer n Data Field 3	DATA12 to DATA15
+1Ch	RMBn.DF4	Receive Message Buffer n Data Field 4	DATA16 to DATA19
+20h	RMBn.DF5	Receive Message Buffer n Data Field 5	DATA20 to DATA23
+24h	RMBn.DF6	Receive Message Buffer n Data Field 6	DATA24 to DATA27
+28h	RMBn.DF7	Receive Message Buffer n Data Field 7	DATA28 to DATA31
+2Ch	RMBn.DF8	Receive Message Buffer n Data Field 8	DATA32 to DATA35
+30h	RMBn.DF9	Receive Message Buffer n Data Field 9	DATA36 to DATA39
+34h	RMBn.DF10	Receive Message Buffer n Data Field 10	DATA40 to DATA43
+38h	RMBn.DF11	Receive Message Buffer n Data Field 11	DATA44 to DATA47
+3Ch	RMBn.DF12	Receive Message Buffer n Data Field 12	DATA48 to DATA51
+40h	RMBn.DF13	Receive Message Buffer n Data Field 13	DATA52 to DATA55
+44h	RMBn.DF14	Receive Message Buffer n Data Field 14	DATA56 to DATA59
+48h	RMBn.DF15	Receive Message Buffer n Data Field 15	DATA60 to DATA63

36.2.61.3 Receive Message Buffer n Header Field 0 (RMBn.HF0) (n = 0 to 31)

Address(es): CANFD.RMB0.HF0 000A 8920h, CANFD.RMB1.HF0 000A 896Ch, CANFD.RMB2.HF0 000A 89B8h, CANFD.RMB3.HF0 000A 8A04h, CANFD.RMB4.HF0 000A 8A50h, CANFD.RMB5.HF0 000A 8A9Ch, CANFD.RMB6.HF0 000A 8AE8h, CANFD.RMB7.HF0 000A 8B34h, CANFD.RMB8.HF0 000A 8D20h, CANFD.RMB9.HF0 000A 8D6Ch, CANFD.RMB10.HF0 000A 8DB8h, CANFD.RMB11.HF0 000A 8E04h, CANFD.RMB12.HF0 000A 8E50h, CANFD.RMB13.HF0 000A 8E9Ch, CANFD.RMB14.HF0 000A 8EE8h, CANFD.RMB15.HF0 000A 8F34h, CANFD.RMB16.HF0 000A 9120h, CANFD.RMB17.HF0 000A 916Ch, CANFD.RMB18.HF0 000A 91B8h, CANFD.RMB19.HF0 000A 9204h, CANFD.RMB20.HF0 000A 9250h, CANFD.RMB21.HF0 000A 929Ch, CANFD.RMB22.HF0 000A 92E8h, CANFD.RMB23.HF0 000A 9334h, CANFD.RMB24.HF0 000A 9520h, CANFD.RMB25.HF0 000A 956Ch, CANFD.RMB26.HF0 000A 95B8h, CANFD.RMB27.HF0 000A 9604h, CANFD.RMB28.HF0 000A 9650h, CANFD.RMB29.HF0 000A 969Ch, CANFD.RMB30.HF0 000A 96E8h, CANFD.RMB31.HF0 000A 9734h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IDE	RTR	—	ID[28:16]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ID[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b28 to b0	ID[28:0]	Identifier	Standard ID/extended ID field	R
b29	—	Reserved	This bit is read as 0	R
b30	RTR	Remote Transmission Request	0: Data frame 1: Remote frame	R
b31	IDE	Identifier Extension	0: Standard ID 1: Extended ID	R

This register stores the ID field, IDE bit, and RTR bit of the received message.

ID[28:0] Bits (Identifier)

The ID[28:0] bits store the standard or extended identifier field of message stored in the receive message buffer.
For alignment of these bits in base and extended format, refer to section 36.2.60, Identifier Bits Alignment.

RTR Bit (Remote Transmission Request)

The RTR bit stores the value of the RTR bit of the received message.
The RTR bit indicates whether the receive message buffer stores a data frame or a remote frame.

Note: There are no remote frames in CAN FD format. When a CAN FD frame is received, this bit reflects the value of the RRS bit.

IDE Bit (Identifier Extension)

The IDE bit stores the value of the IDE bit of the received message.
The IDE bit indicates whether the message stored in the receive message buffer has a standard identifier or extended identifier.

36.2.61.4 Receive Message Buffer n Header Field 1 (RMBn.HF1) (n = 0 to 31)

Address(es): CANFD.RMB0.HF1 000A 8924h, CANFD.RMB1.HF1 000A 8970h, CANFD.RMB2.HF1 000A 89BCh, CANFD.RMB3.HF1 000A 8A08h, CANFD.RMB4.HF1 000A 8A54h, CANFD.RMB5.HF1 000A 8AA0h, CANFD.RMB6.HF1 000A 8AECh, CANFD.RMB7.HF1 000A 8B38h, CANFD.RMB8.HF1 000A 8D24h, CANFD.RMB9.HF1 000A 8D70h, CANFD.RMB10.HF1 000A 8DBCh, CANFD.RMB11.HF1 000A 8E08h, CANFD.RMB12.HF1 000A 8E54h, CANFD.RMB13.HF1 000A 8EA0h, CANFD.RMB14.HF1 000A 8EECh, CANFD.RMB15.HF1 000A 8F38h, CANFD.RMB16.HF1 000A 9124h, CANFD.RMB17.HF1 000A 9170h, CANFD.RMB18.HF1 000A 91BCh, CANFD.RMB19.HF1 000A 9208h, CANFD.RMB20.HF1 000A 9254h, CANFD.RMB21.HF1 000A 92A0h, CANFD.RMB22.HF1 000A 92ECh, CANFD.RMB23.HF1 000A 9338h, CANFD.RMB24.HF1 000A 9524h, CANFD.RMB25.HF1 000A 9570h, CANFD.RMB26.HF1 000A 95BCh, CANFD.RMB27.HF1 000A 9608h, CANFD.RMB28.HF1 000A 9654h, CANFD.RMB29.HF1 000A 96A0h, CANFD.RMB30.HF1 000A 96ECh, CANFD.RMB31.HF1 000A 9738h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DLC[3:0]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TS[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	TS[15:0]	Timestamp	Timestamp value stored for the message in the receive message buffer	R
b27 to b16	—	Reserved	These bits are read as 0	R
b31 to b28	DLC[3:0]	Data Length Code	Number of data bytes received in a CAN frame.	R

This register stores the data length code (DLC) and timestamp of the received message.

TS[15:0] Bits (Timestamp)

The TS[15:0] bits store the timestamp of the received message captured at the point specified by the GFDCFG.TSCPS[1:0] bits.

DLC[3:0] Bits (Data Length Code)

The DLC[3:0] bits store the number of data bytes of the received message.
Refer to Table 5 in ISO 11898-1:2015 Standard for details on defining the number of data bytes.

Note: The maximum number of data bytes in the buffer is specified by the RMCR.PLS[2:0] bits.

36.2.61.5 Receive Message Buffer n Header Field 2 (RMBn.HF2) (n = 0 to 31)

Address(es): CANFD.RMB0.HF2 000A 8928h, CANFD.RMB1.HF2 000A 8974h, CANFD.RMB2.HF2 000A 89C0h, CANFD.RMB3.HF2 000A 8A0Ch, CANFD.RMB4.HF2 000A 8A58h, CANFD.RMB5.HF2 000A 8AA4h, CANFD.RMB6.HF2 000A 8AF0h, CANFD.RMB7.HF2 000A 8B3Ch, CANFD.RMB8.HF2 000A 8D28h, CANFD.RMB9.HF2 000A 8D74h, CANFD.RMB10.HF2 000A 8DC0h, CANFD.RMB11.HF2 000A 8E0Ch, CANFD.RMB12.HF2 000A 8E58h, CANFD.RMB13.HF2 000A 8EA4h, CANFD.RMB14.HF2 000A 8EF0h, CANFD.RMB15.HF2 000A 8F3Ch, CANFD.RMB16.HF2 000A 9128h, CANFD.RMB17.HF2 000A 9174h, CANFD.RMB18.HF2 000A 91C0h, CANFD.RMB19.HF2 000A 920Ch, CANFD.RMB20.HF2 000A 9258h, CANFD.RMB21.HF2 000A 92A4h, CANFD.RMB22.HF2 000A 92F0h, CANFD.RMB23.HF2 000A 933Ch, CANFD.RMB24.HF2 000A 9528h, CANFD.RMB25.HF2 000A 9574h, CANFD.RMB26.HF2 000A 95C0h, CANFD.RMB27.HF2 000A 960Ch, CANFD.RMB28.HF2 000A 9658h, CANFD.RMB29.HF2 000A 96A4h, CANFD.RMB30.HF2 000A 96F0h, CANFD.RMB31.HF2 000A 973Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PTR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	IFL[1:0]		—	—	—	—	—	FDF	BRS	ESI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ESI	Error State Indicator Flag	0: CAN FD frame received from error active node 1: CAN FD frame received from error passive node	R
b1	BRS	Bit Rate Switch Flag	0: CAN FD frame received with no bit rate switch 1: CAN FD frame received with bit rate switch	R
b2	FDF	FD Format Indicator Flag	0: Non CAN FD frame received 1: CAN FD frame received	R
b7 to b3	—	Reserved	These bits are read as 0	R
b9, b8	IFL[1:0]	Information Label	A field that stores the information label attached by the Acceptance Filter	R
b15 to b10	—	Reserved	These bits are read as 0	R
b31 to b16	PTR[15:0]	Pointer	A field that stores the pointer attached by the Acceptance Filter	R

This register stores the FDF bit, BRS bit, and ESI flag of the received message, and pointer for the received message.

ESI Flag (Error State Indicator Flag)

The ESI flag stores the value of the ESI flag of the received CAN FD frame.
When the received FDF bit is 0, this means a Classical CAN frame is received and 0 is stored to this flag.

BRS Flag (Bit Rate Switch Flag)

The BRS flag stores the value of the BRS bit of the received CAN FD frame.
When the received FDF bit is 0, this means a Classical CAN frame is received and 0 is stored to this flag.

FDF Flag (FD Format Indicator Flag)

The FDF flag stores the value of the FDF bit of the received CAN FD frame.

IFL[1:0] Bits (Information Label)

The IFL[1:0] bits store the information label value set in the corresponding entry in the acceptance filter list.

PTR[15:0] Bits (Pointer)

The PTR[15:0] bits store the pointer value set in the corresponding entry in the acceptance filter list.

36.2.61.6 Receive Message Buffer n Data Field p (RMBn.DFp) (n = 0 to 31; p = 0 to 15)

Address(es): CANFD.RMB0.DF0 000A 892Ch to CANFD.RMB0.DF15 000A 8968h,
CANFD.RMB1.DF0 000A 8978h to CANFD.RMB1.DF15 000A 89B4h,
CANFD.RMB2.DF0 000A 89C4h to CANFD.RMB2.DF15 000A 8A00h,
CANFD.RMB3.DF0 000A 8A10h to CANFD.RMB3.DF15 000A 8A4Ch,
CANFD.RMB4.DF0 000A 8A5Ch to CANFD.RMB4.DF15 000A 8A98h,
CANFD.RMB5.DF0 000A 8AA8h to CANFD.RMB5.DF15 000A 8AE4h,
CANFD.RMB6.DF0 000A 8AF4h to CANFD.RMB6.DF15 000A 8B30h,
CANFD.RMB7.DF0 000A 8B40h to CANFD.RMB7.DF15 000A 8B7Ch,
CANFD.RMB8.DF0 000A 8D2Ch to CANFD.RMB8.DF15 000A 8D68h,
CANFD.RMB9.DF0 000A 8D78h to CANFD.RMB9.DF15 000A 8DB4h,
CANFD.RMB10.DF0 000A 8DC4h to CANFD.RMB10.DF15 000A 8E00h,
CANFD.RMB11.DF0 000A 8E10h to CANFD.RMB11.DF15 000A 8E4Ch,
CANFD.RMB12.DF0 000A 8E5Ch to CANFD.RMB12.DF15 000A 8E98h,
CANFD.RMB13.DF0 000A 8EA8h to CANFD.RMB13.DF15 000A 8EE4h,
CANFD.RMB14.DF0 000A 8EF4h to CANFD.RMB14.DF15 000A 8F30h,
CANFD.RMB15.DF0 000A 8F40h to CANFD.RMB15.DF15 000A 8F7Ch,
CANFD.RMB16.DF0 000A 912Ch to CANFD.RMB16.DF15 000A 9168h,
CANFD.RMB17.DF0 000A 9178h to CANFD.RMB17.DF15 000A 91B4h,
CANFD.RMB18.DF0 000A 91C4h to CANFD.RMB18.DF15 000A 9200h,
CANFD.RMB19.DF0 000A 9210h to CANFD.RMB19.DF15 000A 924Ch,
CANFD.RMB20.DF0 000A 925Ch to CANFD.RMB20.DF15 000A 9298h,
CANFD.RMB21.DF0 000A 92A8h to CANFD.RMB21.DF15 000A 92E4h,
CANFD.RMB22.DF0 000A 92F4h to CANFD.RMB22.DF15 000A 9330h,
CANFD.RMB23.DF0 000A 9340h to CANFD.RMB23.DF15 000A 937Ch,
CANFD.RMB24.DF0 000A 952Ch to CANFD.RMB24.DF15 000A 9568h,
CANFD.RMB25.DF0 000A 9578h to CANFD.RMB25.DF15 000A 95B4h,
CANFD.RMB26.DF0 000A 95C4h to CANFD.RMB26.DF15 000A 9600h,
CANFD.RMB27.DF0 000A 9610h to CANFD.RMB27.DF15 000A 964Ch,
CANFD.RMB28.DF0 000A 965Ch to CANFD.RMB28.DF15 000A 9698h,
CANFD.RMB29.DF0 000A 96A8h to CANFD.RMB29.DF15 000A 96E4h,
CANFD.RMB30.DF0 000A 96F4h to CANFD.RMB30.DF15 000A 9730h,
CANFD.RMB31.DF0 000A 9740h to CANFD.RMB31.DF15 000A 977Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

These registers are read-only registers that store the data byte (p × 4 + 3) to data byte (p × 4) of the received message. Unused data bytes are filled with 00h.

36.2.61.7 Receive Message Buffer n Data Register k (RMBn.DATAk) (n = 0 to 31; k = 0 to 63)

Address(es): CANFD.RMB0.DATA0 000A 892Ch to CANFD.RMB0.DATA63 000A 896Bh,
CANFD.RMB1.DATA0 000A 8978h to CANFD.RMB1.DATA63 000A 89B7h,
CANFD.RMB2.DATA0 000A 89C4h to CANFD.RMB2.DATA63 000A 8A03h,
CANFD.RMB3.DATA0 000A 8A10h to CANFD.RMB3.DATA63 000A 8A4Fh,
CANFD.RMB4.DATA0 000A 8A5Ch to CANFD.RMB4.DATA63 000A 8A9Bh,
CANFD.RMB5.DATA0 000A 8AA8h to CANFD.RMB5.DATA63 000A 8AE7h,
CANFD.RMB6.DATA0 000A 8AF4h to CANFD.RMB6.DATA63 000A 8B33h,
CANFD.RMB7.DATA0 000A 8B40h to CANFD.RMB7.DATA63 000A 8B7Fh,
CANFD.RMB8.DATA0 000A 8D2Ch to CANFD.RMB8.DATA63 000A 8D6Bh,
CANFD.RMB9.DATA0 000A 8D78h to CANFD.RMB9.DATA63 000A 8DB7h,
CANFD.RMB10.DATA0 000A 8DC4h to CANFD.RMB10.DATA63 000A 8E03h,
CANFD.RMB11.DATA0 000A 8E10h to CANFD.RMB11.DATA63 000A 8E4Fh,
CANFD.RMB12.DATA0 000A 8E5Ch to CANFD.RMB12.DATA63 000A 8E9Bh,
CANFD.RMB13.DATA0 000A 8EA8h to CANFD.RMB13.DATA63 000A 8EE7h,
CANFD.RMB14.DATA0 000A 8EF4h to CANFD.RMB14.DATA63 000A 8F33h,
CANFD.RMB15.DATA0 000A 8F40h to CANFD.RMB15.DATA63 000A 8F7Fh,
CANFD.RMB16.DATA0 000A 912Ch to CANFD.RMB16.DATA63 000A 916Bh,
CANFD.RMB17.DATA0 000A 9178h to CANFD.RMB17.DATA63 000A 91B7h,
CANFD.RMB18.DATA0 000A 91C4h to CANFD.RMB18.DATA63 000A 9203h,
CANFD.RMB19.DATA0 000A 9210h to CANFD.RMB19.DATA63 000A 924Fh,
CANFD.RMB20.DATA0 000A 925Ch to CANFD.RMB20.DATA63 000A 929Bh,
CANFD.RMB21.DATA0 000A 92A8h to CANFD.RMB21.DATA63 000A 92E7h,
CANFD.RMB22.DATA0 000A 92F4h to CANFD.RMB22.DATA63 000A 9333h,
CANFD.RMB23.DATA0 000A 9340h to CANFD.RMB23.DATA63 000A 937Fh,
CANFD.RMB24.DATA0 000A 952Ch to CANFD.RMB24.DATA63 000A 956Bh,
CANFD.RMB25.DATA0 000A 9578h to CANFD.RMB25.DATA63 000A 95B7h,
CANFD.RMB26.DATA0 000A 95C4h to CANFD.RMB26.DATA63 000A 9603h,
CANFD.RMB27.DATA0 000A 9610h to CANFD.RMB27.DATA63 000A 964Fh,
CANFD.RMB28.DATA0 000A 965Ch to CANFD.RMB28.DATA63 000A 969Bh,
CANFD.RMB29.DATA0 000A 96A8h to CANFD.RMB29.DATA63 000A 96E7h,
CANFD.RMB30.DATA0 000A 96F4h to CANFD.RMB30.DATA63 000A 9733h,
CANFD.RMB31.DATA0 000A 9740h to CANFD.RMB31.DATA63 000A 977Fh

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

These registers are read-only registers that store the data bytes of the received message.
Unused data bytes are filled with 00h.

36.2.61.8 Receive FIFO n (RFBn) (n = 0, 1)

The total number of the receive FIFO (RFB) is two, as shown in Figure 36.33.
The receive FIFO consists of the following registers:

- RFBn.HF0
- RFBn.HF1
- RFBn.HF2
- RFBn.DF0 to RFBn.DF15

Table 36.8 shows the structure of this buffer.

Table 36.8 Structure of Receive FIFO

Address Offset	Symbol	Register Name	Contents
+00h	RFBn.HF0	Receive FIFO n Header Field 0	IDE, RTR, ID
+04h	RFBn.HF1	Receive FIFO n Header Field 1	DLC, timestamp
+08h	RFBn.HF2	Receive FIFO n Header Field 2	Pointer, information label, FDF, BRS, ESI
+0Ch	RFBn.DF0	Receive FIFO n Data Field 0	DATA0 to DATA3
+10h	RFBn.DF1	Receive FIFO n Data Field 1	DATA4 to DATA7
+14h	RFBn.DF2	Receive FIFO n Data Field 2	DATA8 to DATA11
+18h	RFBn.DF3	Receive FIFO n Data Field 3	DATA12 to DATA15
+1Ch	RFBn.DF4	Receive FIFO n Data Field 4	DATA16 to DATA19
+20h	RFBn.DF5	Receive FIFO n Data Field 5	DATA20 to DATA23
+24h	RFBn.DF6	Receive FIFO n Data Field 6	DATA24 to DATA27
+28h	RFBn.DF7	Receive FIFO n Data Field 7	DATA28 to DATA31
+2Ch	RFBn.DF8	Receive FIFO n Data Field 8	DATA32 to DATA35
+30h	RFBn.DF9	Receive FIFO n Data Field 9	DATA36 to DATA39
+34h	RFBn.DF10	Receive FIFO n Data Field 10	DATA40 to DATA43
+38h	RFBn.DF11	Receive FIFO n Data Field 11	DATA44 to DATA47
+3Ch	RFBn.DF12	Receive FIFO n Data Field 12	DATA48 to DATA51
+40h	RFBn.DF13	Receive FIFO n Data Field 13	DATA52 to DATA55
+44h	RFBn.DF14	Receive FIFO n Data Field 14	DATA56 to DATA59
+48h	RFBn.DF15	Receive FIFO n Data Field 15	DATA60 to DATA63

36.2.61.9 Receive FIFO n Header Filed 0 (RFBn.HF0) (n = 0, 1)

Address(es): CANFD.RFB0.HF0 000A 8520h, CANFD.RFB1.HF0 000A 856Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IDE	RTR	—	ID[28:16]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ID[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b28-b0	ID[28:0]	Identifier	Standard ID/extended ID field	R
b29	—	Reserved	This bit is read as 0	R
b30	RTR	Remote Transmission Request	0: Data frame 1: Remote frame	R
b31	IDE	Identifier Extension	0: Standard ID 1: Extended ID	R

This register stores the ID field, IDE bit, and RTR bit of the received message.

ID[28:0] Bits (Identifier)

The ID[28:0] bits store the standard or extended identifier field of message stored in the receive FIFO.
For alignment of these bits in base and extended format, refer to section 36.2.60, Identifier Bits Alignment.

RTR Bit (Remote Transmission Request)

The RTR bit stores the value of the RTR bit of the received message.

Note: There are no remote frames in CAN FD format. When a CAN FD frame is received, this bit reflects the value of the RRS bit.

IDE Bit (Identifier Extension)

The IDE bit stores the value of the IDE bit of the received message.

36.2.61.10 Receive FIFO n Header Filed 1 (RFBn.HF1) (n = 0, 1)

Address(es): CANFD.RFB0.HF1 000A 8524h, CANFD.RFB1.HF1 000A 8570h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DLC[3:0]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TS[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	TS[15:0]	Timestamp	Timestamp value of the received CAN frame	R
b27 to b16	—	Reserved	These bits are read as 0	R
b31 to b28	DLC[3:0]	Data Length Code	Number of data bytes received in a CAN frame	R

This register stores the data length code (DLC) and timestamp of the received message.

TS[15:0] Bits (Timestamp)

The TS[15:0] bits store the timestamp of the received message captured at the point specified by the GFDCFG.TSCPS[1:0] bits.

DLC[3:0] Bits (Data Length Code)

The DLC[3:0] bits store the number of data bytes of the received message.
Refer to Table 5 in ISO 11898-1:2015 Standard for details on defining the number of data bytes.

Note: The maximum number of data bytes in the buffer is specified by the RFCRn.PLS[2:0] bits.

36.2.61.11 Receive FIFO n Header Field 2 (RFBn.HF2) (n = 0, 1)

Address(es): CANFD.RFB0.HF2 000A 8528h, CANFD.RFB1.HF2 000A 8574h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PTR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	IFL[1:0]		—	—	—	—	—	FDF	BRS	ESI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ESI	Error State Indicator Flag	0: CAN FD frame received from error active node 1: CAN FD frame received from error passive node	R
b1	BRS	Bit Rate Switch Flag	0: CAN FD frame received with no bit rate switch 1: CAN FD frame received with bit rate switch	R
b2	FDF	FD Format Indicator Flag	0: Non CAN FD frame received 1: CAN FD frame received	R
b7 to b3	—	Reserved	These bits are read as 0	R
b9, b8	IFL[1:0]	Information Label	A field that stores the information label attached by the Acceptance Filter	R
b15 to b10	—	Reserved	These bits are read as 0	R
b31 to b16	PTR[15:0]	Pointer	A field that stores the pointer attached by the Acceptance Filter	R

This register stores the FDF bit, BRS bit, and ESI flag of the received message, and pointer for the received message.

ESI Flag (Error State Indicator Flag)

The ESI flag stores the value of the ESI flag of the received CAN FD frame.
When the received FDF bit is 0, this means a Classical CAN frame is received and 0 is stored to this flag.

BRS Flag (Bit Rate Switch Flag)

The BRS flag stores the value of the BRS bit of the received CAN FD frame.
When the received FDF bit is 0, this means a Classical CAN frame is received and 0 is stored to this flag.

FDF Flag (FD Format Indicator Flag)

The FDF flag stores the value of the FDF bit of the received CAN FD frame.

IFL[1:0] Bits (Information Label)

The IFL[1:0] bits store the information label value set in the corresponding entry in the acceptance filter list.

PTR[15:0] Bits (Pointer)

The PTR[15:0] bits store the pointer value set in the corresponding entry in the acceptance filter list.

36.2.61.12 Receive FIFO n Data Field p (RFBn.DFp) (n = 0, 1; p = 0 to 15)

Address(es): CANFD.RFB0.DF0 000A 852Ch to CANFD.RFB0.DF15 000A 8568h,
CANFD.RFB1.DF0 000A 8578h to CANFD.RFB1.DF15 000A 85B4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

These registers are read-only registers that store the data byte (p × 4 + 3) to data byte (p × 4) of the received message.
Unused data bytes are filled with 00h.

36.2.61.13 Receive FIFO n Data Register k (RFBn.DATAk) (n = 0, 1; k = 0 to 63)

Address(es): CANFD.RFB0.DATA0 000A 852Ch to CANFD.RFB0.DATA63 000A 856Bh,
CANFD.RFB1.DATA0 000A 8578h to CANFD.RFB1.DATA63 000A 85B7h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

These registers are read-only registers that store the data bytes of the received message.
Unused data bytes are filled with 00h.

36.2.61.14 Common FIFO 0 (CFB0)

The total number of the common FIFO (CFB) is one, as shown in Figure 36.33.
The common FIFO consists of the following registers:

- CFB0.HF0
- CFB0.HF1
- CFB0.HF2
- CFB0.DF0 to CFB0.DF15

Table 36.9 shows the structure of this buffer.

Table 36.9 Structure of Common FIFO

Address Offset	Symbol	Register Name	Contents
+00h	CFB0.HF0	Common FIFO 0 Header Field 0	IDE, RTR, ID
+04h	CFB0.HF1	Common FIFO 0 Header Field 1	DLC, timestamp
+08h	CFB0.HF2	Common FIFO 0 Header Field 2	Pointer, information label, FDF, BRS, ESI
+0Ch	CFB0.DF0	Common FIFO 0 Data Field 0	DATA0 to DATA3
+10h	CFB0.DF1	Common FIFO 0 Data Field 1	DATA4 to DATA7
+14h	CFB0.DF2	Common FIFO 0 Data Field 2	DATA8 to DATA11
+18h	CFB0.DF3	Common FIFO 0 Data Field 3	DATA12 to DATA15
+1Ch	CFB0.DF4	Common FIFO 0 Data Field 4	DATA16 to DATA19
+20h	CFB0.DF5	Common FIFO 0 Data Field 5	DATA20 to DATA23
+24h	CFB0.DF6	Common FIFO 0 Data Field 6	DATA24 to DATA27
+28h	CFB0.DF7	Common FIFO 0 Data Field 7	DATA28 to DATA31
+2Ch	CFB0.DF8	Common FIFO 0 Data Field 8	DATA32 to DATA35
+30h	CFB0.DF9	Common FIFO 0 Data Field 9	DATA36 to DATA39
+34h	CFB0.DF10	Common FIFO 0 Data Field 10	DATA40 to DATA43
+38h	CFB0.DF11	Common FIFO 0 Data Field 11	DATA44 to DATA47
+3Ch	CFB0.DF12	Common FIFO 0 Data Field 12	DATA48 to DATA51
+40h	CFB0.DF13	Common FIFO 0 Data Field 13	DATA52 to DATA55
+44h	CFB0.DF14	Common FIFO 0 Data Field 14	DATA56 to DATA59
+48h	CFB0.DF15	Common FIFO 0 Data Field 15	DATA60 to DATA63

36.2.61.15 Common FIFO 0 Header Field 0 (CFB0.HF0)

Address(es): CANFD.CFB0.HF0 000A 85B8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IDE	RTR	THENT	ID[28:16]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ID[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b28 to b0	ID[28:0]	Identifier	Standard ID/extended ID field	R/W
b29	THENT	Transmission History Entry	Receive FIFO mode: Reserved. This bit is read as 0. Transmit FIFO mode: 0: Entry is not stored to the Transmission History after successful transmission. 1: Entry is stored to the Transmission History after successful transmission.	R/W
b30	RTR	Remote Transmission Request	0: Data frame 1: Remote frame	R/W
b31	IDE	Identifier Extension	0: Standard ID 1: Extended ID	R/W

In receive FIFO mode, this register is a read-only register for reading the ID field, IDE bit, and RTR bit of the received message from the beginning of the FIFO buffer.
In transmit FIFO mode, this register is a read/write register for writing the ID field, IDE bit, and RTR bit of the message to be transmitted at the end of the FIFO buffer.

ID[28:0] Bits (Identifier)

In receive FIFO mode, the ID[28:0] bits store the standard or extended identifier field of the received message.
In transmit FIFO mode, the ID[28:0] bits are used to set the value of the standard or extended identifier field of the message to be transmitted.
For alignment of these bits in base and extended format, refer to section 36.2.60, Identifier Bits Alignment.

THENT Bit (Transmission History Entry)

This bit is valid only in transmit FIFO mode.
The THENT bit controls whether the corresponding entry is stored in the transmission history after a successful transmission of the message.

RTR Bit (Remote Transmission Request)

In receive FIFO mode, the RTR bit stores the value of the RTR bit of the received message.
In transmit FIFO mode, the RTR bit is used to specify the value of the RTR bit of the message to be transmitted.

Note: There are no remote frames in CAN FD format. When a CAN FD frame was received (receive mode), this bit reflects the value of the RRS bit. When transmitting a CAN FD frame (CFB0.HF2.FDF = 1), this bit is transmitted dominant regardless of the value of this bit.

IDE Bit (Identifier Extension)

In receive FIFO mode, the IDE bit stores the value of the IDE bit of the received message.
In transmit FIFO mode, the IDE bit is used to specify the value of the IDE bit of the message to be transmitted.

36.2.61.16 Common FIFO 0 Header Field 1 (CFB0.HF1)

Address(es): CANFD.CFB0.HF1 000A 85BCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DLC[3:0]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TS[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	TS[15:0]	Timestamp	Timestamp value of the received CAN frame (in receive FIFO mode)	R/W
b27 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31 to b28	DLC[3:0]	Data Length Code	Number of data bytes received in a CAN frame, or to be transmitted in a CAN frame	R/W

In receive FIFO mode, this register is a read-only register for reading the data length code (DLC) and timestamp of the received message from the beginning of the FIFO buffer.
In transmit FIFO mode, this register is a read/write register for writing the data length code (DLC) of the message to be transmitted at the end of the FIFO buffer.

TS[15:0] Bits (Timestamp)

These bits are valid only in receive FIFO mode.
The TS[15:0] bits store the timestamp of the received message captured at the point specified by the GFDCFG.TSCPS[1:0] bits.

DLC[3:0] Bits (Data Length Code)

In receive FIFO mode, the DLC[3:0] bits store the number of data bytes of the received message.
In transmit FIFO mode, The DLC[3:0] bits are used to specify the number of data bytes of the message to be transmitted.
Refer to Table 5 in ISO 11898-1:2015 Standard for details on defining the number of data bytes.

Note: The maximum number of data bytes in the buffer is specified by the CFCR0.PLS[2:0] bits.

36.2.61.17 Common FIFO 0 Header Field 2 (CFB0.HF2)

Address(es): CANFD.CFB0.HF2 000A 85C0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PTR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	IFL[1:0]		—	—	—	—	—	FDF	BRS	ESI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ESI	Error State Indicator	0: CAN FD frame received or to transmit by error active node 1: CAN FD frame received or to transmit by error passive node	R/W
b1	BRS	Bit Rate Switch	0: CAN FD frame received or to transmit with no bit rate switch 1: CAN FD frame received or to transmit with bit rate switch	R/W
b2	FDF	FD Format Indicator	0: Non CAN FD frame received or to transmit 1: CAN FD frame received or to transmit	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	IFL[1:0]	Information Label	A field that stores the information label attached by the Acceptance Filter or sets the information label to be stored in the Transmission History	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31 to b16	PTR[15:0]	Pointer	A field that stores the pointer attached by the Acceptance Filter or sets the pointer to be stored in the Transmission History	R/W

In receive FIFO mode, this register is a read-only register for reading the FDF bit, BRS bit, and ESI flag of the received message, and information label and pointer for the received message from the beginning of the FIFO buffer.
In transmit FIFO mode, this register is a read/write register for writing the FDF bit, BRS bit, and ESI flag of the message to be transmitted, and the information label and pointer to be stored in the transmission history at the end of the FIFO buffer.

ESI Bit (Error State Indicator)

In receive FIFO mode, the ESI bit stores the value of the ESI flag of the received CAN FD frame. When the received FDF bit is 0 (Classical CAN frame), 0 is stored to this bit.
In transmit FIFO mode, the ESI bit is used to specify the value of the ESI flag of the CAN FD frame to be transmitted. If the channel is not in error passive, the ESI flag of the transmitted message equals the value of this bit. If the channel is in error passive, this bit is transmitted recessive regardless of the value of this bit.

BRS Bit (Bit Rate Switch)

In receive FIFO mode, the BRS bit stores the value of the BRS bit of the received CAN FD frame. When the received FDF bit is 0 (Classical CAN frame), 0 is stored to this bit.
In transmit FIFO mode, the BRS bit is used to specify the value of the BRS bit of the CAN FD frame to be transmitted.

FDF Bit (FD Format Indicator)

In receive FIFO mode, the FDF bit stores the value of the FDF bit of the received CAN FD frame.
In transmit FIFO mode, the FDF bit is used to specify the value of the FDF bit of the CAN FD frame to be transmitted.

IFL[1:0] Bits (Information Label)

In receive FIFO mode, the IFL[1:0] bits store the information label value set in the corresponding entry in the acceptance filter list.
In transmit FIFO mode, the IFL[1:0] bits are used to specify the information label value to be stored in the transmission history after a successful transmission of the message.

PTR[15:0] Bits (Pointer)

In receive FIFO mode, the PTR[15:0] bits store the pointer value set in the corresponding entry in the acceptance filter list.
In transmit FIFO mode, the PTR[15:0] bits are used to specify the pointer value to be stored in the transmission history after a successful transmission of the message.

36.2.61.18 Common FIFO 0 Data Field p (CFB0.DFp) (p = 0 to 15)

Address(es): CANFD.CFB0.DF0 000A 85C4h to CANFD.CFB0.DF15 000A 8600h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

In receive FIFO mode, these registers are read-only registers for reading the data byte (p × 4 + 3) to data byte (p × 4) of the received message from the beginning of the FIFO buffer. Unused data bytes are filled with 00h.
In transmit FIFO mode, these registers are read/write registers for storing data byte (p × 4 + 3) to data byte (p × 4) of the message to be transmitted.

36.2.61.19 Common FIFO 0 Data Register k (CFB0.DATAk) (k = 0 to 63)

Address(es): CANFD.CFB0.DATA0 000A 85C4h to CANFD.CFB0.DATA63 000A 8603h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

In receive FIFO mode, these registers are read-only registers for reading the data bytes of the received message from the beginning of the FIFO buffer. Unused data bytes are filled with 00h.
In transmit FIFO mode, these registers are read/write registers for storing the data bytes of the message to be transmitted.

36.2.61.20 Transmit Message Buffer n (TMBn) (n = 0 to 3)

The total number of transmit message buffer (TMB) is four, as shown in Figure 36.33.
The transmit message buffer consists of the following registers:

- TMBn.HF0
- TMBn.HF1
- TMBn.HF2
- TMBn.DF0 to TMBn.DF15

Table 36.10 shows the structure of this buffer.

Table 36.10 Structure of Transmit Message Buffer

Address Offset	Symbol	Register Name	Contents
+00h	TMBn.HF0	Transmit Message Buffer n Header Field 0	IDE, RTR, ID
+04h	TMBn.HF1	Transmit Message Buffer n Header Field 1	DLC
+08h	TMBn.HF2	Transmit Message Buffer n Header Field 2	Pointer, information label, FDF, BRS, ESI
+0Ch	TMBn.DF0	Transmit Message Buffer n Data Field 0	DATA0 to DATA3
+10h	TMBn.DF1	Transmit Message Buffer n Data Field 1	DATA4 to DATA7
+14h	TMBn.DF2	Transmit Message Buffer n Data Field 2	DATA8 to DATA11
+18h	TMBn.DF3	Transmit Message Buffer n Data Field 3	DATA12 to DATA15
+1Ch	TMBn.DF4	Transmit Message Buffer n Data Field 4	DATA16 to DATA19
+20h	TMBn.DF5	Transmit Message Buffer n Data Field 5	DATA20 to DATA23
+24h	TMBn.DF6	Transmit Message Buffer n Data Field 6	DATA24 to DATA27
+28h	TMBn.DF7	Transmit Message Buffer n Data Field 7	DATA28 to DATA31
+2Ch	TMBn.DF8	Transmit Message Buffer n Data Field 8	DATA32 to DATA35
+30h	TMBn.DF9	Transmit Message Buffer n Data Field 9	DATA36 to DATA39
+34h	TMBn.DF10	Transmit Message Buffer n Data Field 10	DATA40 to DATA43
+38h	TMBn.DF11	Transmit Message Buffer n Data Field 11	DATA44 to DATA47
+3Ch	TMBn.DF12	Transmit Message Buffer n Data Field 12	DATA48 to DATA51
+40h	TMBn.DF13	Transmit Message Buffer n Data Field 13	DATA52 to DATA55
+44h	TMBn.DF14	Transmit Message Buffer n Data Field 14	DATA56 to DATA59
+48h	TMBn.DF15	Transmit Message Buffer n Data Field 15	DATA60 to DATA63

36.2.61.21 Transmit Message Buffer n Header Field 0 (TMBn.HF0) (n = 0 to 3)

Address(es): CANFD.TMB0.HF0 000A 8604h, CANFD.TMB1.HF0 000A 8650h, CANFD.TMB2.HF0 000A 869Ch, CANFD.TMB3.HF0 000A 86E8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IDE	RTR	THENT	ID[28:16]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ID[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b28 to b0	ID[28:0]	Identifier	Standard ID/extended ID field	R/W
b29	THENT	Transmission History Entry	0: Entry is not stored in transmission history after successful transmission. 1: Entry is stored in transmission history after successful transmission.	R/W
b30	RTR	Remote Transmission Request	0: Data frame 1: Remote frame	R/W
b31	IDE	Identifier Extension	0: Standard ID 1: Extended ID	R/W

This register is used to store the ID field, IDE bit, and RTR bit of the message to be transmitted, and to specify whether to store it in the transmission history.
Do not write to this register when the CAN channel is in CH_SLEEP mode.

ID[28:0] Bits (Identifier)

The ID[28:0] bits are used to set the value of the standard or extended identifier field of the message stored in the transmit message buffer.
For alignment of these bits in base and extended format, refer to section 36.2.60, Identifier Bits Alignment.

THENT Bit (Transmission History Entry)

The THENT bit controls whether the corresponding entry is stored in the transmission history after a successful transmission of the message stored in the transmit message buffer.

RTR Bit (Remote Transmission Request)

The RTR bit is used to specify the value of the RTR bit of the message to be transmitted.

Note: There are no remote frames in CAN FD format. When transmitting a CAN FD frame (TMBn.HF2.FDF = 1), this bit is transmitted dominant regardless of the value of this bit.

IDE Bit (Identifier Extension)

The IDE bit is used to specify the value of the IDE bit of the message to be transmitted.

36.2.61.22 Transmit Message Buffer n Header Field 1 (TMBn.HF1) (n = 0 to 3)

Address(es): CANFD.TMB0.HF1 000A 8608h, CANFD.TMB1.HF1 000A 8654h, CANFD.TMB2.HF1 000A 86A0h, CANFD.TMB3.HF1 000A 86ECh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DLC[3:0]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b27 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31 to b28	DLC[3:0]	Data Length Code	Number of data bytes to be transmitted in a CAN frame.	R/W

This register is used to store the data length code (DLC) fields of the message to be transmitted.
Do not write to this register when the CAN channel is in CH_SLEEP mode.

DLC[3:0] Bits (Data Length Code)

The DLC[3:0] bits are used to specify the number of data bytes of the message to be transmitted when the corresponding TMBn.HF0.RTR bit is set to 0.
Refer to Table 5 in ISO 11898-1:2015 Standard for details on defining the number of data bytes.

36.2.61.23 Transmit Message Buffer n Header Field 2 (TMBn.HF2) (n = 0 to 3)

Address(es): CANFD.TMB0.HF2 000A 860Ch, CANFD.TMB1.HF2 000A 8658h, CANFD.TMB2.HF2 000A 86A4h, CANFD.TMB3.HF2 000A 86F0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PTR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	IFL[1:0]		—	—	—	—	—	FDF	BRS	ESI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ESI	Error State Indicator	0: CAN FD frame to transmit by error active node 1: CAN FD frame to transmit by error passive node	R/W
b1	BRS	Bit Rate Switch	0: CAN FD frame to transmit with no bit rate switch 1: CAN FD frame to transmit with bit rate switch	R/W
b2	FDF	FD Format Indicator	0: Non CAN FD frame to transmit 1: CAN FD frame to transmit	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	IFL[1:0]	Information Label	A field that sets the information label to be stored in the Transmission History	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31 to b16	PTR[15:0]	Pointer	A field that sets the pointer to be stored in the Transmission History	R/W

This register is used to store the FDF bit, BRS bit, and ESI flag of the message to be transmitted, and the information label and pointer to be stored in the transmission history.
Do not write to this register when the CAN channel is in CH_SLEEP mode.

ESI Bit (Error State Indicator)

The ESI bit is used to specify the value of the ESI flag of the CAN FD frame to be transmitted.
If the channel is not in error passive, the ESI flag of the transmitted message equals the value of this bit. If the channel is in error passive, this bit is transmitted recessive regardless of the value of this bit.

BRS Bit (Bit Rate Switch)

The BRS bit is used to specify the value of the BRS bit of the CAN FD frame to be transmitted.

FDF Bit (FD Format Indicator)

The FDF bit is used to specify the value of the FDF bit of the CAN FD frame to be transmitted.

IFL[1:0] Bits (Information Label)

The IFL[1:0] bits are used to specify the information label value to be stored in the transmission history after a successful transmission of the message.

PTR[15:0] Bits (Pointer)

The PTR[15:0] bits are used to specify the pointer value to be stored in the transmission history after a successful transmission of the message.

36.2.61.24 Transmit Message Buffer n Data Field p (TMBn.DFp) (n = 0 to 3; p = 0 to 15)

Address(es): CANFD.TMB0.DF0 000A 8610h to CANFD.TMB0.DF15 000A 864Ch,
CANFD.TMB1.DF0 000A 865Ch to CANFD.TMB1.DF15 000A 8698h,
CANFD.TMB2.DF0 000A 86A8h to CANFD.TMB2.DF15 000A 86E4h,
CANFD.TMB3.DF0 000A 86F4h to CANFD.TMB3.DF15 000A 8730h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

These registers are read/write registers for storing data byte (p × 4 + 3) to data byte (p × 4) of the message to be transmitted.
Do not write to these registers when the CAN channel is in CH_SLEEP mode.

36.2.61.25 Transmit Message Buffer n Data Register k (TMBn.DATAk) (n = 0 to 3; k = 0 to 63)

Address(es): CANFD.TMB0.DATA0 000A 8610h to CANFD.TMB0.DATA63 000A 864Fh,
CANFD.TMB1.DATA0 000A 865Ch to CANFD.TMB1.DATA63 000A 869Bh,
CANFD.TMB2.DATA0 000A 86A8h to CANFD.TMB2.DATA63 000A 86E7h,
CANFD.TMB3.DATA0 000A 86F4h to CANFD.TMB3.DATA63 000A 8733h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

These registers are read/write registers for storing the data bytes of the message to be transmitted.
Do not write to these registers when the CAN channel is in CH_SLEEP mode.

36.2.62 ECC Control/Status Register (ECCSR)

Address(es): CANFD.ECCSR 000E D000h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	EC2EAS	EC1EAS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ECEDWC[1:0]		—	—	ECOVF	EC2EC	EC1EC	—	—	ECEDE	EC1ECD	EC2EIE	EC1EIE	EC2EF	EC1EF	ECEF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ECEF	ECC Error Flag	0: These is no ECC error in the last read RAM data. 1: There is an ECC error in the last read RAM data.	R
b1	EC1EF	1-Bit ECC Error Detection Flag	0: 1-bit ECC error is not detected. 1: 1-bit ECC error is detected.	R
b2	EC2EF	2-Bit ECC Error Detection Flag	0: 2-bit ECC error is not detected. 1: 2-bit ECC error is detected.	R
b3	EC1EIE	1-Bit ECC Error Detection Interrupt Enable	0: 1-bit ECC error detection interrupt is disabled. 1: 1-bit ECC error detection interrupt is enabled.	R/W
b4	EC2EIE	2-Bit ECC Error Detection Interrupt Enable	0: 2-bit ECC error detection interrupt is disabled. 1: 2-bit ECC error detection interrupt is enabled.	R/W
b5	EC1ECD	1-Bit ECC Error Correction Disable	0: At 1-bit ECC error detection, the error correction is executed. 1: At 1-bit ECC error detection, the error correction is not executed.	R/W
b6	ECEDE	ECC Error Detection Enable	0: ECC error detection is disabled. 1: ECC error detection is enabled.	R/W
b8, b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9	EC1EC	EC1EF Flag Clear	Writing 1 to this bit clears the EC1EF flag. Writing 0 is ignored. This bit is read as 0.	R/W
b10	EC2EC	EC2EF Flag Clear	Writing 1 to this bit clears the EC2EF flag. Writing 0 is ignored. This bit is read as 0.	R/W
b11	ECOVF	ECC Overflow Detection Flag	0: The ECEAR register overflow has not occurred. 1: The ECEAR register overflow has occurred.	R
b13, b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15, b14	ECEDWC[1:0]	ECEDE Bit Write Control	Enables or disables write access to the ECEDE bit.	R/W
b16	EC1EAS	1-Bit ECC Error Detected Address Stored Flag	0: No valid address is stored in the ECEAR register. 1: The address where 1-bit ECC error occurred is stored in the ECEAR register.	R
b17	EC2EAS	2-Bit ECC Error Detected Address Stored Flag	0: No valid address is stored in the ECEAR register. 1: The address where 2-bit ECC error occurred is stored in the ECEAR register.	R
b31 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ECEF Flag (ECC Error Flag)

The ECEF flag indicates whether there is an ECC error in the last read RAM data. This flag is updated each time the RAM is read.

If the ECEDE bit is set to 1 without initializing the RAM, the value of this flag has no meaning.

[Setting condition]

- If there is an ECC error in read RAM data when the ECEDE bit is set to 1 (ECC error detection is enabled)

[Clearing condition]

- Under the condition that there is no ECC error in read RAM data
- When the ECEDE bit is set to 0 (ECC error detection is disabled)

EC1EF Flag (1-Bit ECC Error Detection Flag)

The EC1EF flag indicates that the 1-bit ECC error is detected in the RAM read data.
When the 1-bit ECC error interrupt is enabled and this flag is set to 1, the 1-bit ECC error interrupt (EC1EI) is generated.
When the 1-bit ECC error is detected again under the condition that this flag is set to 1, the interrupt is not generated.

[Setting condition]

- If there is a 1-bit ECC error in read RAM data when the ECEDE bit is set to 1 (ECC error detection is enabled)

[Clearing condition]

- Writing 1 to the EC1EC bit
- When the ECEDE bit is set to 0 (ECC error detection is disabled)

If writing 1 to the EC1EC bit and detecting a 1-bit ECC error occur at the same time, the EC1EF flag becomes 0.

EC2EF Flag (2-Bit ECC Error Detection Flag)

The EC2EF flag indicates that the 2-bit ECC error is detected in the RAM read data.
When the 2-bit ECC error interrupt is enabled and this flag is set to 1, the 2-bit ECC error interrupt (EC2EI) is generated.
When the 2-bit ECC error is detected again under the condition that this flag is set to 1, the interrupt is not generated.

[Setting condition]

- If there is a 2-bit ECC error in read RAM data when the ECEDE bit is set to 1 (ECC error detection is enabled)

[Clearing condition]

- Writing 1 to the EC2EC bit
- When the ECEDE bit is set to 0 (ECC error detection is disabled)

If writing 1 to the EC2EC bit and detecting a 2-bit ECC error occur at the same time, the EC2EF flag becomes 0.

EC1EIE Bit (1-Bit ECC Error Detection Interrupt Enable)

The EC1EIE bit enables or disables a 1-bit ECC error detection interrupt.
If the EC1EF flag becomes 1 when this bit is 1, a 1-bit ECC error detection interrupt (EC1EI) is generated.

EC2EIE Bit (2-Bit ECC Error Detection Interrupt Enable)

The EC2EIE bit enables or disables a 2-bit ECC error detection interrupt.
If the EC2EF flag becomes 1 when this bit is 1, a 2-bit ECC error detection interrupt (EC2EI) is generated.

EC1ECD Bit (1-Bit ECC Error Correction Disable)

The EC1ECD bit enables or disables to correct the 1-bit ECC error when the ECEDE bit is set to 1 (ECC error detection is enabled). When this bit is set to 1, the RAM output data is not corrected even if a 1-bit ECC error is detected.

ECEDE Bit (ECC Error Detection Enable)

Setting the ECEDE bit to 1 enables ECC error detection.
Writing to this bit is valid only when the ECEDWC[1:0] bits are set to 01b.

EC1EC Bit (EC1EF Flag Clear)

The EC1EC bit is used to clear the EC1EF flag.
The EC1EF flag is cleared by writing 1 to this bit while the EC1EF flag is set to 1. Additionally, the ECOVF flag, EC1EAS flag, and EC2EAS flag are also cleared.
If the EC1EF flag is cleared by the EC1EC bit and the setting factor of the EC1EF flag occurs at the same time, the

EC1EF flag becomes 0.

EC2EC Bit (EC2EF Flag Clear)

The EC2EC bit is used to clear the EC2EF flag.
The EC2EF flag is cleared by writing 1 to this bit while the EC2EF flag is set to 1. Additionally, the ECOVF flag, EC1EAS flag, and EC2EAS flag are also cleared.
If the EC2EF flag is cleared by the EC2EC bit and the setting factor of the EC2EF flag occurs at the same time, the EC2EF flag becomes 0.

ECOVF Flag (ECC Overflow Detection Flag)

If a new ECC error is detected and the address is overwritten when the address is already stored in the ECEAR register, the ECOVF flag becomes 1 and an ECC overflow interrupt (ECOVI) is generated.
The ECC overflow interrupt is generated again when this flag is set to 1 and new ECC error is detected.

[Setting condition]

- When new error address is captured under the condition that error address is already captured in the ECEAR register.

[Clearing condition]

- Writing 1 to the EC1EC bit
- Writing 1 to the EC2EC bit
- When the ECEDE bit is set to 0 (ECC error detection is disabled)

If the ECOVF flag is cleared by the EC1EC or EC2EC bit and the setting condition of the ECOVF flag occurs at the same time, the ECOVF flag becomes 0.

ECEDWC[1:0] Bits (ECEDE Bit Write Control)

The ECEDWC[1:0] bits are used to enable or disable write access to the ECEDE bit. The read value is always 00b.
When the value of these bits is 01b, it is possible to have write access to the ECEDE bit. If these bits are other than 01b, write access to the ECEDE bit is ignored and the value does not change.

EC1EAS Flag (1-Bit ECC Error Detected Address Stored Flag)

The EC1EAS flag indicates that the address where the 1-bit ECC error occurred is stored in the ECEAR register when the ECEDE bit is set to 1 (ECC error detection is enabled).
When 1-bit ECC error is detected while the 2-bit ECC error address has already been captured in the ECEAR register, the ECEAR register is not updated and this flag is not updated.

[Setting condition]

- When a 1-bit ECC error is detected under the condition that the ECEDE bit is set to 1 (ECC error detection is enabled), and the address is stored in the ECEAR register

[Clearing condition]

- Writing 1 to the EC1EC bit
- When the ECEDE bit is set to 0 (ECC error detection is disabled)

If the EC1EAS flag is cleared by the EC1EC bit and the setting factor of the EC1EAS flag occurs at the same time, the EC1EAS flag becomes 0.

EC2EAS Flag (2-Bit ECC Error Detected Address Stored Flag)

The EC2EAS flag indicates that the address where the 2-bit ECC error occurred is stored in the ECEAR register when the ECEDE bit is set to 1 (ECC error detection is enabled).
When 2-bit ECC error is detected while the 1-bit ECC error address has already been captured in the ECEAR register, the ECEAR register is updated and this flag becomes 1.

[Setting condition]

- When a 2-bit ECC error is detected under the condition that the ECEDE bit is set to 1 (ECC error detection is enabled), and the address is stored in the ECEAR register

[Clearing condition]

- Writing 1 to the EC2EC bit
- When the ECEDE bit is set to 0 (ECC error detection is disabled)

If the EC2EAS flag is cleared by the EC2EC bit and the setting factor of the EC2EAS flag occurs at the same time, the EC2EAS flag becomes 0.

36.2.63 ECC Test Mode Register (ECTMR)

Address(es): CANFD.ECTMR 000E D004h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ECTMWC[1:0]		—	—	—	—	—	—	ECTME	—	—	—	—	—	ECDIS	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b1	ECDIS	ECC Decoder Input Select	0: Input RAM output data to data input of decode circuit 1: Select the ECTDR register to data input of decode circuit	R/W
b6 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	ECTME	ECC Test Mode Enable	0: Access to the ECDIS bit and the ECTDR register is disabled 1: Access to the ECDIS bit and the ECTDR register is enabled	R/W
b13 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15, b14	ECTMWC[1:0]	ECTME Bit Write Control	Enable or disable write access to the ECTME bit.	R/W

ECDIS Bit (ECC Decoder Input Select)

The ECDIS bit selects which of the data value read from RAM and the value of the ECTDR register is used as input data to the ECC decoder.

The write access to this bit is enabled when the ECTME bit is set to 1. It is also possible to set them at the same time.

This bit is cleared by setting the ECTME bit to 0.

ECTME Bit (ECC Test Mode Enable)

The ECTME bit is used to enable or disable the access to the ECDIS bit and the ECTDR register.

This bit can be written only when the ECTMWC[1:0] bits is set to 10b.

ECTMWC[1:0] Bits (ECTME Bit Write Control)

The ECTMWC[1:0] bits are used to enable or disable the write access to the ECTME bit. The read value is always 00b.

When the value of these bits is 10b, it is possible to have write access to the ECTME bit. If these bits are other than 10b, write access to the ECTME bit is ignored and the value does not change.

36.2.64 ECC Decoder Test Data Register (ECTDR)

Address(es): CANFD.ECTDR 000E D00Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is used to set the data for testing the ECC decode.
When the ECTMR.ECTME bit is set to 1, this register can be read and written.
When the ECTMR.ECTME bit is set to 0, the value of this register becomes 00000000h.
When the ECTMR.ECDIS bit is set to 1, the value set in this register is used as the input data of the ECC decoder instead of the data read from RAM.

36.2.65 ECC Error Address Register (ECEAR)

Address(es): CANFD.ECEAR 000E D010h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register holds the address where the ECC error occurred.

If an ECC error is detected when the ECCSR.ECEDE bit is 1 (ECC error detection is enabled), b12 to b2 of the RAM address are stored in b10 to b0 of this register.
If the same error occurred again, this register is not updated.
If a 2-bit ECC error is detected while the address where the 1-bit ECC error occurred is already stored, the ECEAR register is overwritten with the new address and the ECCSR.EC2EAS flag is set to 1.
If a 1-bit ECC error is detected while the address where the 2-bit ECC error occurred is already stored, the ECEAR register is not updated and the ECCSR.EC1EAS flag is not updated.

36.3 Operating Mode

The operating modes of the CANFD module can be classified into two groups:

- Global modes
- Channel modes.

36.3.1 Global Modes

These modes are applicable for the complete CANFD module and therefore are called Global modes.
The Global modes of the CANFD module are:

- GL_SLEEP
- GL_RESET
- GL_HALT
- GL_OPERATION.

Figure 36.2 shows the possible transitions between the Global modes.

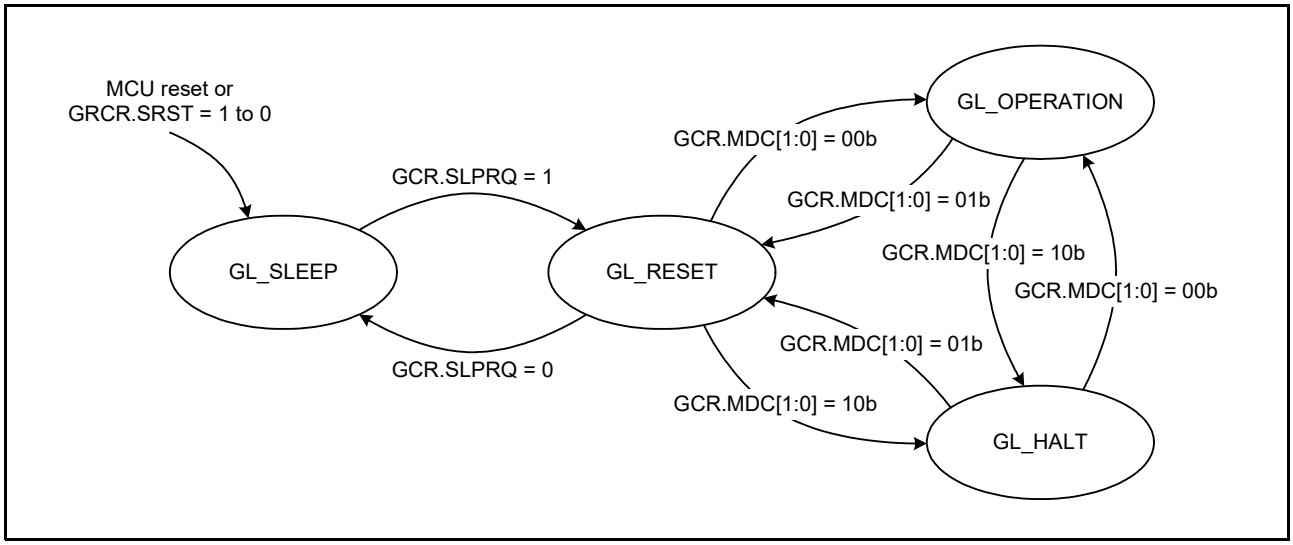


Figure 36.2 Transition between CANFD Global Modes

Change in the Global mode can affect the Channel mode. For details, refer to section 36.3.3, Global Mode and Channel Mode Transition Interactions.

36.3.1.1 GL_SLEEP Mode

The CANFD module automatically enters GL_SLEEP mode when the MCU reset is released or the software reset bit (GRCR.SRST) is changed from 1 to 0.
The CANFD module also enters the GL_SLEEP mode when the GCR.SLPRQ bit is set to 1 while it is in GL_RESET mode. The SLPRQ bit cannot be set to 1 in GL_HALT or GL_OPERATION mode.
Setting the GCR.SLPRQ bit to 1 sets the CHCR.SLPRQ bit to 1 and forces CAN channel into the CH_SLEEP mode.
GL_SLEEP mode is used for power saving purpose. When CANFD module is in GL_SLEEP mode, only the clock for writing to the SLPRQ bit is active. All other clocks are stopped and all other functions of the CANFD module are suspended.
Read access from all registers is still possible and all register values are preserved.
After setting the GCR.SLPRQ bit to 1, it is necessary to confirm the GSR.SLPST flag that the GL_SLEEP status has been updated, indicating successful transition to GL_SLEEP mode before the GCR.SLPRQ bit can be cleared again.

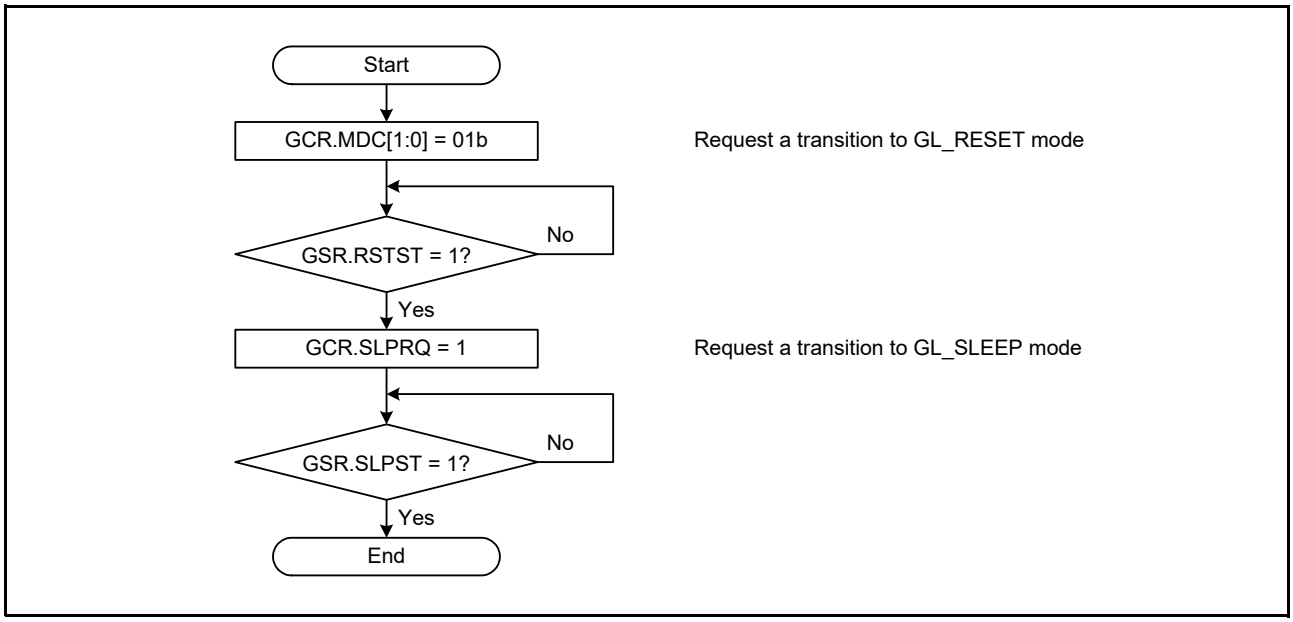


Figure 36.3 Procedure for Entering GL_SLEEP Mode

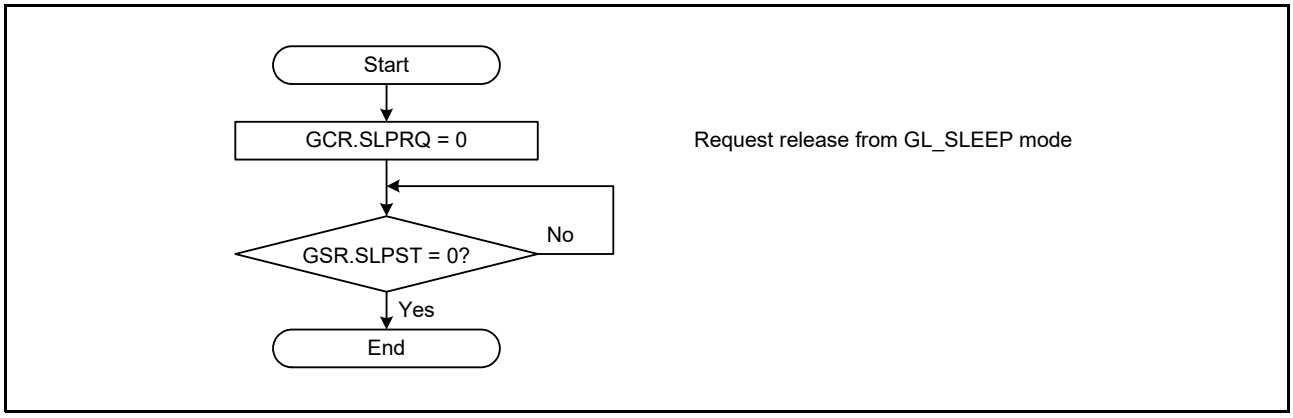


Figure 36.4 Procedure for Exiting GL_SLEEP Mode

36.3.1.2 GL_RESET Mode

The CANFD module enters GL_RESET mode in the following ways:

- The GCR.MDC[1:0] bits are set to 01b while the CANFD module is in GL_HALT or GL_OPERATION mode
- The GCR.SLPRQ bit is set to 0 while the CANFD module is in GL_SLEEP mode.

In GL_RESET mode, all CANFD module functions are suspended and all status and flag registers are initialized. Additionally all FIFOs and all transmit queues are disabled and transmission control bits are cleared.
In this mode, configuration registers other than the GTMCR register and interrupt enable registers are not initialized, so the CANFD module can be configured.
Setting the Global mode to GL_RESET by setting the GCR.MDC[1:0] bits to 01b sets the CHCR.MDC[1:0] bits to 01b and forces the channel into the CH_RESET mode.
When the channel is already in CH_RESET or CH_SLEEP mode, this automatic transition is not performed.
After setting the GCR.MDC[1:0] bits to 01b (GL_RESET mode), it is necessary to confirm that the GSR.RSTST flag has been updated, indicating successful transition to GL_RESET mode before the GCR.MDC[1:0] bits can be changed again.

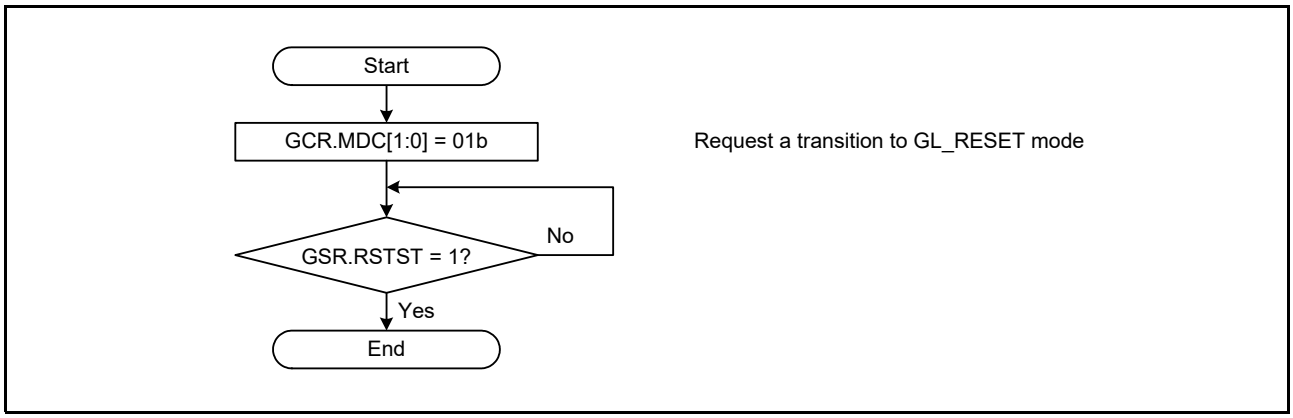


Figure 36.5 Procedure for Entering GL_RESET Mode

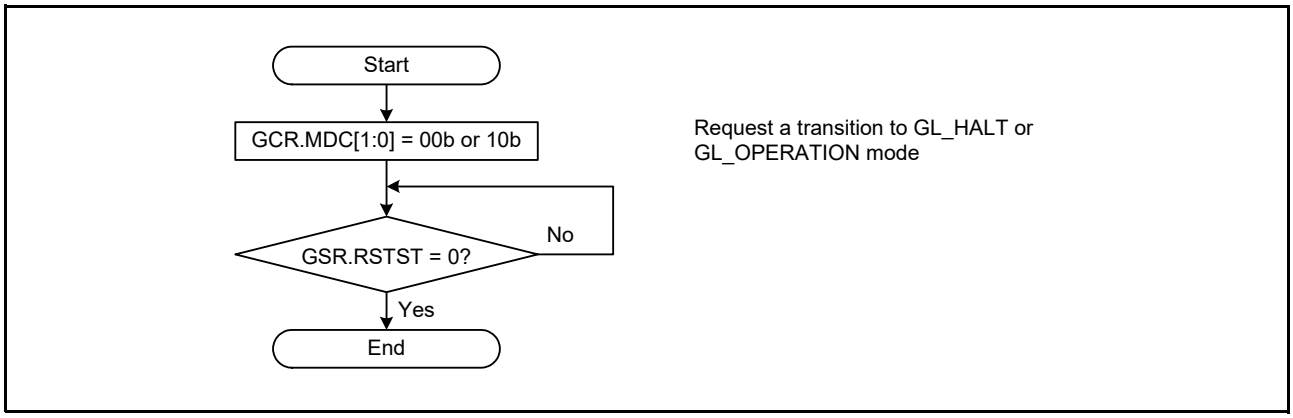


Figure 36.6 Procedure for Exiting GL_RESET Mode

36.3.1.3 GL_HALT Mode

The CANFD module enters GL_HALT mode in the following ways:

- Set the GCR.MDC[1:0] bits to 10b while the CANFD module is in GL_RESET mode:
  – the channels is in either CH_RESET or CH_SLEEP mode and remains in this mode.
- Set the GCR.MDC[1:0] bits to 10b while the CANFD module is in GL_OPERATION mode:
  – CAN channel in CH_RESET, CH_HALT, or CH_SLEEP mode remains in this mode
  – CAN channel in CH_OPERATION mode transits to CH_HALT mode
  – the GSR.HLTST flag is set to 1 when CAN channel exits CH_OPERATION mode.

If a transmission or reception is in progress for a CAN channel, the channel enters CH_HALT mode after waiting for completion of the communication.
Similarly, if a CAN channel is in bus-off, the channel may not enter CH_HALT mode until the bus-off recovery sequence is completed, depending on the channel configuration.
In GL_HALT mode, all communications are suspended and the status and flag registers do not change (only when a channel is in the bus-off, its CHSR.REC[7:0] and TEC[7:0] bits are set to 00h).
Additionally, the GTMCR register and GTMER register are not initialized in this mode. The GL_HALT mode is used to configure Global Test Modes.
Setting the Global mode to GL_HALT by setting the GCR.MDC[1:0] bits to 10b sets the CHCR.MDC[1:0] bits to 10b for the channels that are in CH_OPERATION mode and forces these channels into the CH_HALT mode.
For channels that are already in CH_RESET, CH_HALT, or CH_SLEEP mode, this automatic transition is not performed.
Therefore, the GL_HALT mode request can be used to shut down all CAN channel communications without loss of messages and disruption on the related CAN bus (no interruption of reception/transmission processes on the channels).
After setting the GCR.MDC[1:0] bits to 10b (GL_HALT mode), it is necessary to confirm that the GSR.HLTST flag has been updated to indicate a successful transition to GL_HALT mode. Do not set any other registers until confirming the GSR.HLTST flag is set.

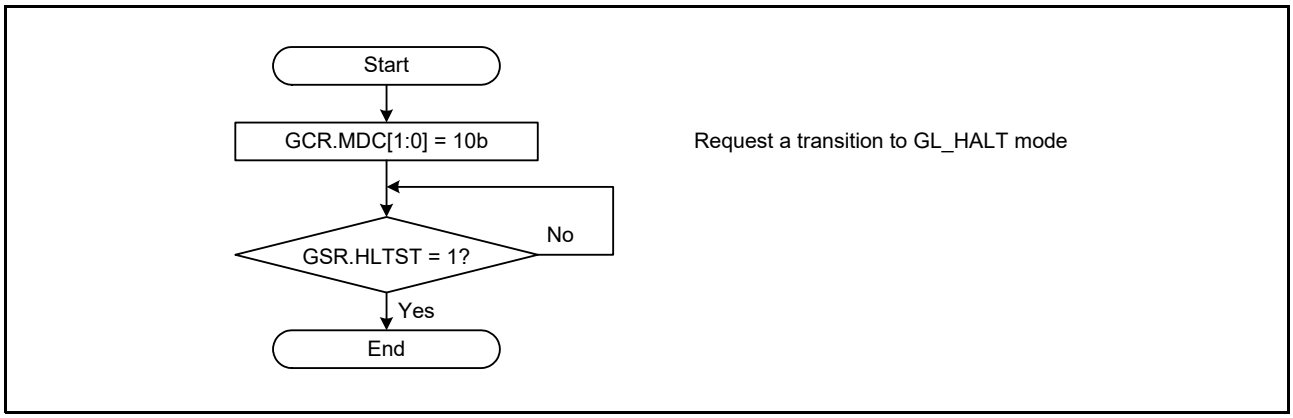


Figure 36.7 Procedure for Entering GL_HALT Mode

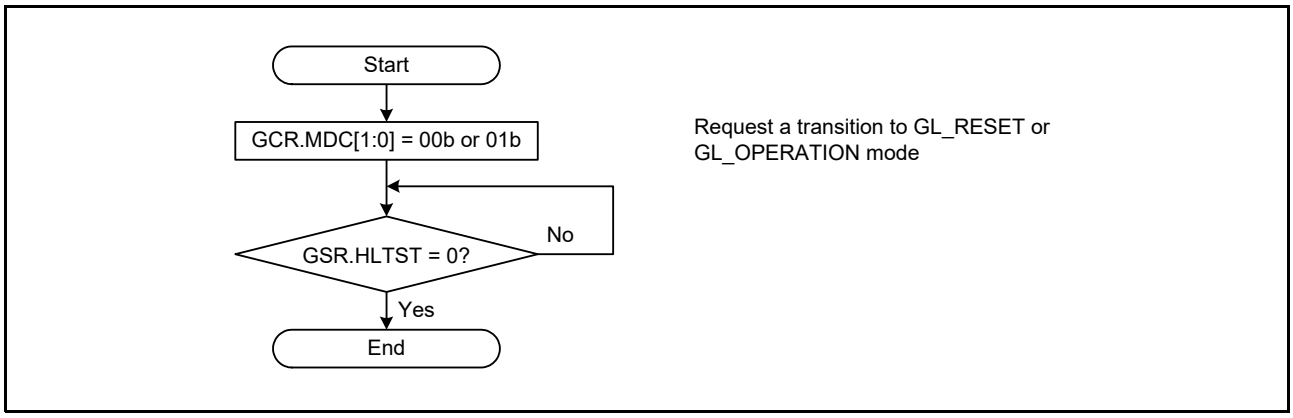


Figure 36.8 Procedure for Exiting GL_HALT Mode

36.3.1.4 GL_OPERATION Mode

The CANFD module enters this mode when the GCR.MDC[1:0] bits are set to 00b.
The CANFD channel can be set to CH_OPERATION mode and start CAN communication only when CANFD is in GL_OPERATION mode.
After setting the GCR.MDC[1:0] bits to 00b (GL_OPERATION mode), it is necessary to confirm that the GSR.RSTST flag and the GSR.HLTST flag have been set to 0 to indicate a successful transition to GL_OPERATION mode before the GCR.MDC[1:0] bits can be modified again.

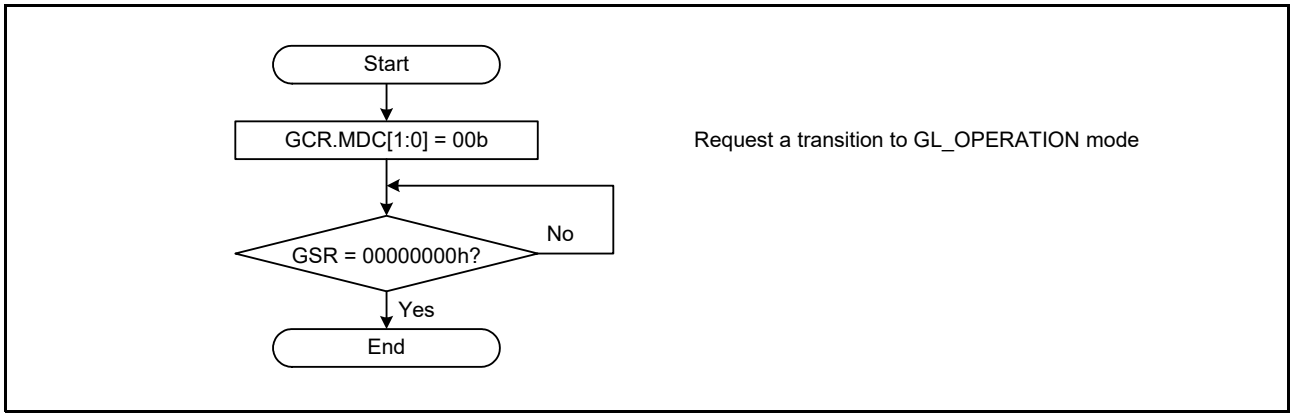


Figure 36.9 Procedure for Entering GL_OPERATION Mode

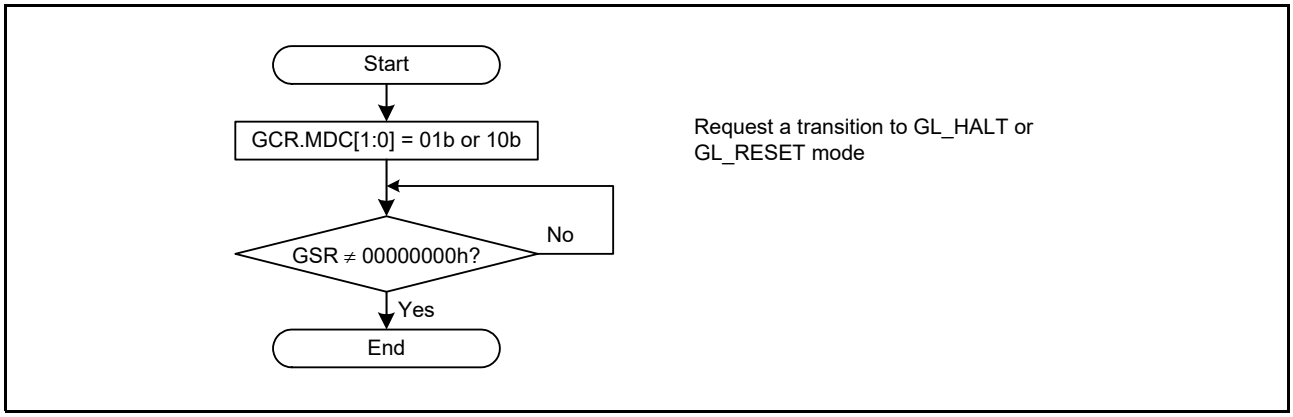


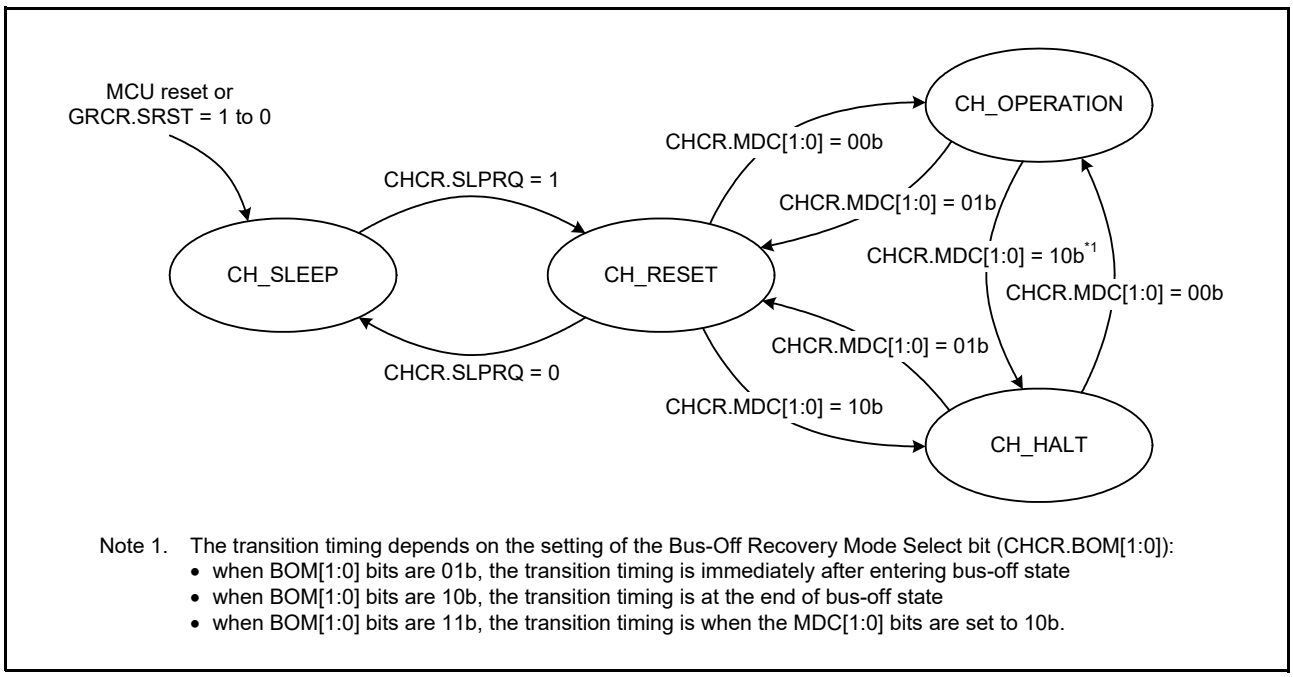
Figure 36.10 Procedure for Exiting GL_OPERATION Mode

36.3.2 Channel Modes

The Channel modes of the CANFD module are:

- CH_RESET
- CH_HALT
- CH_OPERATION
- CH_SLEEP.

Figure 36.11 shows the possible transitions between the channel modes.



Note 1. The transition timing depends on the setting of the Bus-Off Recovery Mode Select bit (CHCR.BOM[1:0]):
- when BOM[1:0] bits are 01b, the transition timing is immediately after entering bus-off state
- when BOM[1:0] bits are 10b, the transition timing is at the end of bus-off state
- when BOM[1:0] bits are 11b, the transition timing is when the MDC[1:0] bits are set to 10b.

Figure 36.11 Transition between CAN Channel Modes

36.3.2.1 CH_SLEEP Mode

The CAN channel of the CANFD module automatically enters CH_SLEEP mode when the MCU reset is released or the software reset bit (GRCR.SRST) is changed from 1 to 0.
The CAN channel also enters CH_SLEEP mode when the CHCR.SLPRQ bit is set to 1 while the CAN channel is in CH_RESET mode. The CHCR.SLPRQ bit cannot be set to 1 in CH_HALT or CH_OPERATION mode.
Entering CH_SLEEP mode instantly stops the clock supplied to the CAN channel unit and therefore reduces power consumption.
After setting the CHCR.SLPRQ bit to 1 and before setting it to 0 again, it is necessary to use the CHSR.SLPST flag to confirm that transition to CH_SLEEP mode was successful.
During CH_SLEEP mode, do not write to channel related registers. Read operation is still possible.

36.3.2.2 CH_RESET Mode

The CAN channel of CANFD module enters this mode in the following ways:

- The CHCR.MDC[1:0] bits are set to 01b while the CAN channel is in CH_HALT or CH_OPERATION mode
- The CHCR.SLPRQ bit is set to 0 while the CAN channel is in CH_SLEEP mode
- The GCR.MDC[1:0] bits are set to 01b while the CAN channel is not in CH_SLEEP or CH_RESET mode.

In CH_RESET mode, all CAN channel status and flags are initialized.
In addition, all transmission-related control bits of the channel are cleared, and the transmit queue of the channel is also disabled.
In this mode, the configuration registers except for the bits related to the channel test mode are not initialized, so the CAN channel can be configured for communication.
After setting the CHCR.MDC[1:0] bits to 01b (CH_RESET mode) and before modifying these bits again, it is necessary to use the CHSR.RSTST flag to confirm that transition to CH_RESET mode was successful.
Refer to Table 36.11 for the behavior of transitioning to CH_RESET mode while CAN communication is in progress.

36.3.2.3 CH_HALT Mode

The CAN channel of CANFD module enters this mode in the following ways:

- The CHCR.MDC[1:0] bits are set to 10b while the CAN channel is in CH_RESET or CH_OPERATION mode
- The GCR.MDC[1:0] bits are set to 10b while the CAN channel is in CH_OPERATION mode.

In CH_HALT mode, all CAN communication of the channel is suspended, but all statuses and flags remain unchanged during CH_HALT mode (except for the bus-off state where the CHSR.REC[7:0] and TEC[7:0] bits for the channel are set to 00h).
In addition, in this mode, the bits related to the channel test mode are not initialized. Use CH_HALT mode to configure the channel test mode.
After setting the CHCR.MDC[1:0] bits to 10b (CH_HALT mode) and before modifying these bits again, it is necessary to use the CHSR.HLTST flag to confirm that transition to CH_HALT mode was successful.
Refer to Table 36.11 for the transition behavior to CH_HALT mode while CAN communication is in progress.

Table 36.11 Transition Behavior in CH_RESET Mode and CH_HALT Mode

Mode	State		
	Receiver	**Transmitter**	**Bus-Off**
CH_RESET mode (CHCR.MDC[1:0] = 01b)	The CAN channel transits to CH_RESET mode without waiting for the completion of the ongoing reception.*1	The CAN channel transits to CH_RESET mode without waiting for the completion of the ongoing transmission.*1	The CAN channel transits to CH_RESET mode without waiting for the completion of the bus-off recovery.
CH_HALT mode (CHCR.MDC[1:0] = 10b)	CAN channel transits to CH_HALT mode at the end of the ongoing reception or when an error occurs.*2	CAN channel transits to CH_HALT mode after completion of the ongoing transmission.	When the CHCR.BOM[1:0] bits are 00b, a CH_HALT mode request is accepted only after the completion of the full bus-off recovery sequence. When the CHCR.BOM[1:0] bits are 10b, the CAN channel transits automatically to CH_HALT mode after waiting for the completion of the bus-off recovery. When the CHCR.BOM[1:0] bits are 01b, the CAN channel transits automatically to CH_HALT mode without waiting for the completion of the bus-off recovery. When the CHCR.BOM[1:0] bits are 11b, the CAN channel transits to CH_HALT mode as soon as CH_HALT mode is requested (without waiting for the completion of the bus-off recovery).

Note 1. If the entry to CH_RESET mode is required only at the end of an ongoing communication, then CH_HALT mode can be requested first to prevent interruption of CAN communication by direct transition to CH_RESET mode. After the CAN channel enters CH_HALT mode, the CH_RESET mode can be requested.

Note 2. If CAN communication is locked at dominant level after an error flag, software can detect this situation by monitoring the channel related BusLock flag and resolve lock condition by setting the CAN channel to CH_RESET mode.

36.3.2.4 CH_OPERATION Mode (in Other than Bus-Off State)

The CH_OPERATION mode is activated by setting the CHCR.MDC[1:0] bits to 00b. If 11 consecutive recessive bits are detected after entering the CH_OPERATION mode, the CHSR.CRDY flag is set to 1 and the CAN channel:

- Enables the functions of the communication by allowing the channel to become an active node on the CAN network
- Releases the internal fault confinement logic including receive and transmit error counters.

At this point, the CAN channel can start transmission and reception of messages.
Within the CH_OPERATION mode, the channel may be in four different sub-modes, depending on which type of communication functions are performed (refer to Figure 36.12):

- Idle mode: The CAN channel is neither receiving nor transmitting
- Receive mode: The channel is receiving a message transmitted by another CAN node
- Transmit mode: The channel is transmitting a message
  (The channel may receive its own message simultaneously when Self Test mode is enabled.)
- Bus-off mode: The CAN channel is cut-off from CAN bus communication.

After setting the CHCR.MDC[1:0] bits to 00b (CH_OPERATION mode) and before modifying these bits again, it is necessary to use the CHSR.RSTST flag and the CHSR.HLTST flag to confirm that transition to CH_OPERATION mode was successful.

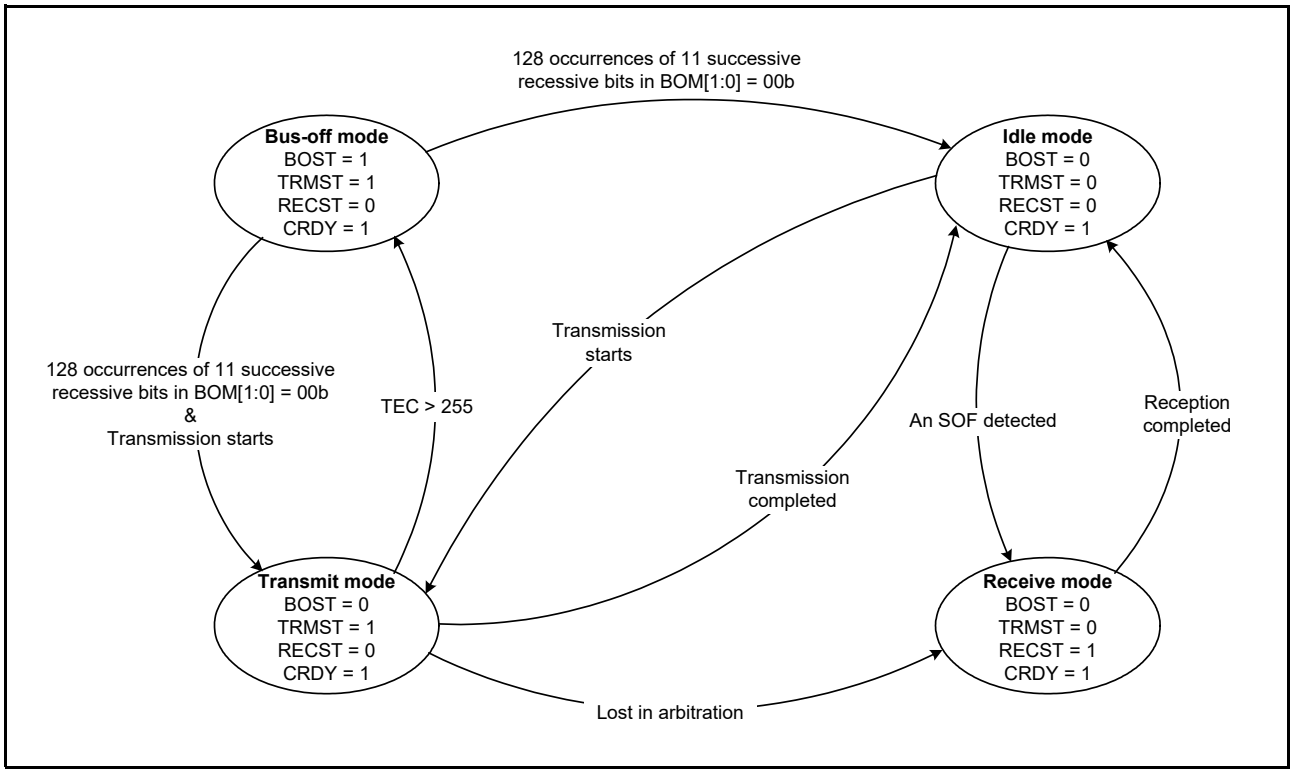


Figure 36.12 Sub-modes of CH_OPERATION Mode (only when CHCR.BOM[1:0] = 00b)

36.3.2.5 CH_OPERATION Mode (in Bus-Off State)

The CAN channel bus-off state is entered according to the fault confinement rules of the CAN specification. The following modes can be configured for returning to the CH_OPERATION mode from the bus-off state:

- CHCR.BOM[1:0] = 00b:
  Bus-off recovery is compliant to ISO 11898-1. The CAN channel re-enters CAN communication (error active state) after 11 consecutive recessive bits are detected 128 times. The CHSR.TEC[7:0] and REC[7:0] bits are initialized to 00h, and the CHESR.BORDF flag is set to 1 in this case.
- CHCR.BOM[1:0] = 01b:
  When entering bus-off state, the CAN channel changes the value of the CHCR.MDC[1:0] bits to 10b and enters CH_HALT mode automatically. The CHSR.TEC[7:0] and REC[7:0] bits are initialized to 00h and the CHESR.BORDF flag is not set to 1 in this case.
- CHCR.BOM[1:0] = 10b:
  When entering bus-off state, the CAN channel changes the value of the CHCR.MDC[1:0] bits to 10b and enters CH_HALT mode automatically after the CAN channel has completed the bus-off recovery sequence (after 11 consecutive recessive bits are detected 128 times). The CHSR.TEC[7:0] and REC[7:0] bits are initialized to 00h and the CHESR.BORDF flag is set to 1 in this case.
- CHCR.BOM[1:0] = 11b:
  Bus-off recovery is initiated but CAN channel can immediately enter CH_HALT mode when still in bus-off state if a request is made to enter CH_HALT mode. The CHSR.TEC[7:0] and REC[7:0] bits are initialized to 00h and the CHESR.BORDF flag is not set to 1.
  Without setting CHCR.MDC[1:0] = 10b and when 11 recessive bits is detected 128 times continuously, transition conditions become the same as CHCR.BOM[1:0] = 00b.

If the recovery from bus-off occurs normally in this mode (after waiting for 128 sequences of 11 consecutive recessive bits), and no CH_HALT request has been generated during this period, then the CHESR.BORDF flag is set to 1.
When software writes to the CHCR.MDC[1:0] bits at the same time as the CAN channel enters CH_HALT mode (at the start of bus-off when CHCR.BOM[1:0] = 01b, or at the end of bus-off when CHCR.BOM[1:0] = 10b), the software request has the highest priority.

Note: In the above case, the automatic setting of the CHCR.MDC[1:0] bits to CH_HALT mode request is performed when the CHCR.MDC[1:0] bits value is previously 00b (CH_OPERATION mode).

Additionally, it is possible to force the CAN channel to recover from the bus-off state by setting the CHCR.RTBO bit to 1. The error state changes from bus-off state to integrating state with a maximum delay of 1 bit time, and the CAN communication becomes possible again after 11 consecutive recessive bits are detected. The CHESR.BORDF flag is not set to 1 in this case, and the CHSR.TEC[7:0] and REC[7:0] bits are initialized to 00h.
Before setting the CHCR.RTBO bit to 1, all pending transmissions from the transmit message buffers, transmit queues and/or common FIFO in transmit mode should be disabled.
The disable of the pending transmit message buffer, transmit queue or common FIFO must be confirmed by the corresponding acknowledge flags (TMSRn.TXRF[1:0] flags, TQSR0.EMPTY flag, and CFSR0.EMPTY flag).
The CHCR.RTBO bit should be used for bus-off recovery only when the CHCR.BOM[1:0] bits are set to 00b.
Setting this bit in any state other than bus-off has no effect and the bit is cleared immediately.
Table 36.12 lists the behavior of the bus-off entry detect flag (CHESR.BOEDF) and the bus-off recovery detect flag (CHESR.BORDF) according to the CHCR.BOM[1:0] bit settings.

Table 36.12 Behavior of Bus-off Entry and Recovery Flags

CHCR.BOM[1:0]	CHESR.BOEDF Flag	CHESR.BORDF Flag
00b	Becomes 1 on entry to bus-off state.	Becomes 1 on exit from bus-off state.
00b Set the CHCR.RTBO bit to 1		Becomes 1 only if normal bus-off recovery occurs before setting the CHCR.RTBO bit to 1.
01b		Does not become 1.
10b		Becomes 1 on exit from bus-off state.
11b		Becomes 1 only if normal bus-off recovery occurs before requesting a transition to CH_HALT mode.

To make efficient software, it is not necessary to wait for the bus-off recovery sequence to end.
It is possible to perform a transmission re-initialization during bus-off recovery. To do this, follow the recommended software flow in Figure 36.13.

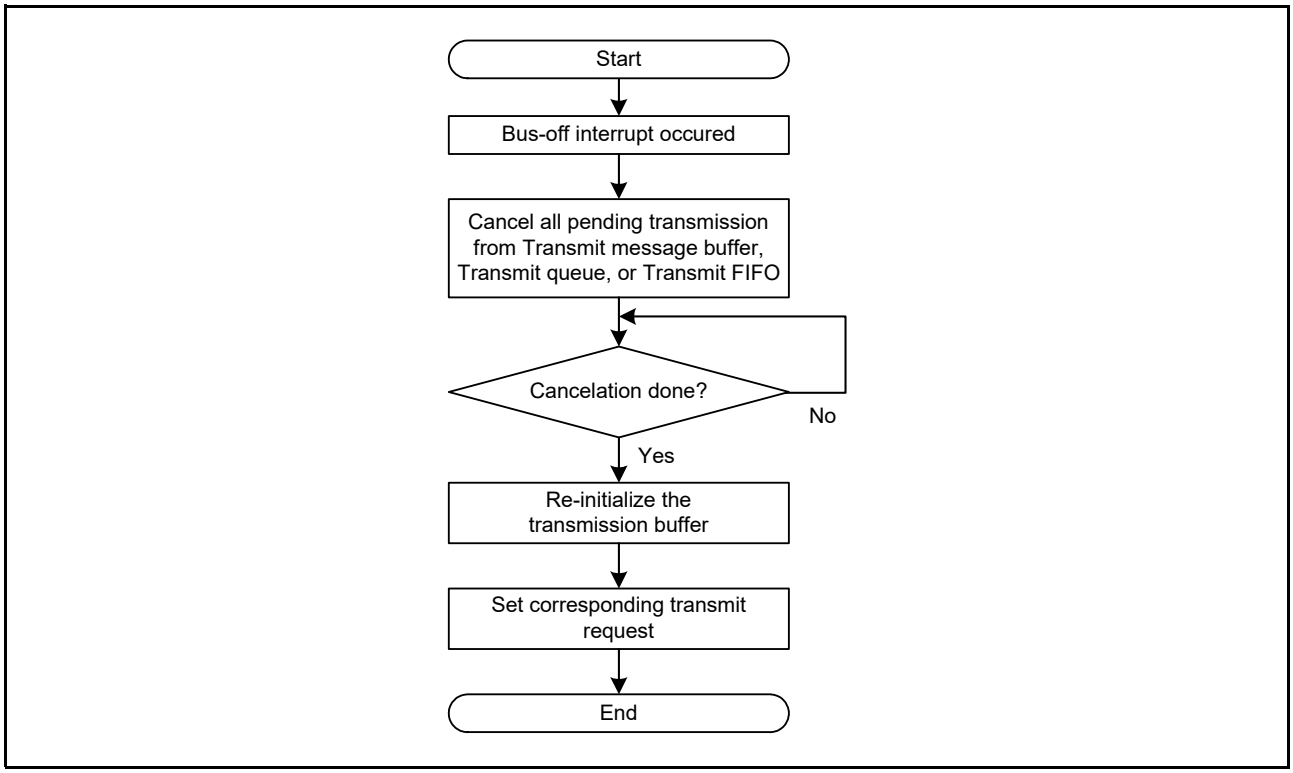


Figure 36.13 Transmission Re-Initialization During Bus-Off

36.3.3 Global Mode and Channel Mode Transition Interactions

The interaction between Global mode setting and Channel mode setting is as follows:

- Changing the CHCR.MDC[1:0] bits does not affect the GCR.MDC[1:0] bits.
- Changing the GCR.MDC[1:0] bits affects the channel mode control as described in Table 36.13.

Table 36.13 Interaction between Global Mode and Channel Mode Transition

Global Mode Change	Channel Mode before Changing Global Mode	Channel Mode after Changing Global Mode
GL_SLEEP → GL_RESET	CH_SLEEP	CH_SLEEP (no change)
GL_RESET → GL_SLEEP	CH_SLEEP	CH_SLEEP (no change)
	CH_RESET	CH_SLEEP
GL_RESET → GL_HALT	CH_SLEEP	CH_SLEEP (no change)
	CH_RESET	CH_RESET (no change)
GL_RESET → GL_OPERATION	CH_SLEEP	CH_SLEEP (no change)
	CH_RESET	CH_RESET (no change)
GL_HALT → GL_RESET	CH_SLEEP	CH_SLEEP (no change)
	CH_RESET	CH_RESET (no change)
	CH_HALT	CH_RESET
GL_HALT → GL_OPERATION	CH_SLEEP	CH_SLEEP (no change)
	CH_RESET	CH_RESET (no change)
	CH_HALT	CH_HALT (no change)
GL_OPERATION → GL_RESET	CH_SLEEP	CH_SLEEP (no change)
	CH_RESET	CH_RESET (no change)
	CH_HALT	CH_RESET
	CH_OPERATION	CH_RESET
GL_OPERATION → GL_HALT	CH_SLEEP	CH_SLEEP (no change)
	CH_RESET	CH_RESET (no change)
	CH_HALT	CH_HALT (no change)
	CH_OPERATION	After communication ends, CH_HALT

36.3.3.1 Timing of Global Mode Change

The transition time for the Global mode changes are shown in the following table.

Table 36.14 Maximum Transition Time for the Global Mode

From	To	Maximum Transition Time
GL_SLEEP	GL_RESET	3 × PCLKB*2
GL_RESET	GL_SLEEP	3 × PCLKB
GL_RESET	GL_HALT	10 × PCLKB
GL_RESET	GL_OPERATION	10 × PCLKB
GL_HALT	GL_RESET	2 bit times (1 Tq + 16 × PCLKB + 2 DLL clock cycles)
GL_HALT	GL_OPERATION	3 × PCLKB
GL_OPERATION	GL_RESET	2 bit times (1 Tq + 16 × PCLKB + 2 DLL clock cycles)
GL_OPERATION	GL_HALT	3 CAN frames (1 CAN frame + 3424 × PCLKB)*1 *3

Note 1. The given transition time is the time without any errors on the bus. In case of an error condition, the transition time can lengthen to an uncalculated result. The transition time can also come to a stuck condition for locked receive lines or continued error conditions.
Note 2. Exit GL_SLEEP mode only when the GSR.RAMST flag is cleared.
Note 3. Tq, CAN frame and bit times are related to the individual channels. For the maximum transition time, the channel with the lowest bit rate must be used.

36.3.3.2 Timing of Channel Mode Change

Table 36.15 shows the transition time for the Channel mode changes.

Table 36.15 Maximum Transition Time for the Channel Mode

From	To	Maximum Transition Time
CH_SLEEP	CH_RESET	3 × PCLKB
CH_RESET	CH_SLEEP	3 × PCLKB
CH_RESET	CH_HALT	3 bit times (1 CAN bit + 2 Tq + 8 × PCLKB + 2 DLL clock cycles)
CH_RESET	CH_OPERATION	4 bit times (2 CAN bits + 1 TSEG1 + 12 × PCLKB + 2 DLL clock cycles)
CH_HALT	CH_RESET	2 bit times (1 Tq + 10 × PCLKB + 2 DLL clock cycles)
CH_HALT	CH_OPERATION	4 bit times (< 4 CAN bits)*3
CH_OPERATION	CH_RESET	2 bit times (1 Tq + 10 × PCLKB + 2 DLL clock cycles)
CH_OPERATION	CH_HALT	2 CAN frames (1 CAN frame + 13 CAN bits)*1 *2

Note 1. The time specified for this transition does not include the case where channel enters bus-off state. For bus-off, the timing depends on the configuration of the CHCR.BOM[1:0] bits.

Note 2. The given transition time is the time without any errors on the bus. In case of an error condition, the transition time can lengthen to an uncalculated result. The transition time can also come to a stuck condition for locked receive lines or continued error conditions.

Note 3. In general, if the bit rate prescaler value (NBCR.BRP[9:0]) is changed in CH_HALT mode, the transition time can deviate. The internal prescaler is a free running down counter that creates the Tq clock, and new BRP value is captured when the counter reaches the value 0.

36.4 Initialization of CANFD Module

Before starting CAN communications, configure the following settings:

- Clock setting
- Bit timing setting (nominal and data bit rate)
- Bit Rate setting (nominal and data bit rate)
- CANFD setting
- Acceptance Filter setting (configuration of Acceptance Filter List)
- Receive FIFO and Transmit FIFO setting
- CAN operating mode setting.

36.4.1 Initialization of CAN Clock, Bit Timing and Bit Rate

36.4.1.1 Bit Timing Conditions

The following lines describe the composition of each segment and the limitations that apply to the segment setting.

1. Each segment setting

- SS = Fixed to 1 Tq
  TSEG1 = 2 Tq to 256 Tq (NBCR), 2 Tq to 32 Tq (DBCR)
  TSEG2 = 2 Tq to 128 Tq (NBCR), 2 Tq to 16 Tq (DBCR)
  SJW = 1 Tq to 128 Tq (NBCR), 1 Tq to 16 Tq (DBCR)
  SS + TSEG1 + TSEG2 = 8 Tq to 385 Tq (NBCR), 5 Tq to 49 Tq (DBCR)

2. Limitations on TSEG1, TSEG2 and SJW
   NBCR register: TSEG1 > TSEG2 ≥ SJW
   DBCR register: TSEG1 ≥ TSEG2 ≥ SJW

Table 36.16 shows an example of how to set the bit timing to achieve the required sample point settings.

Table 36.16 Bit Timing Examples

1 Bit	Set Value (Tq)				Sample Point (%)
	SS	TSEG1	TSEG2	SJW	
5 Tq	1	2	2	1	60.00
8 Tq	1	4	3	1	62.50
	1	5	2	1	75.00
10 Tq	1	6	3	1	70.00
	1	7	2	1	80.00
12 Tq	1	8	3	1	75.00
	1	9	2	1	83.33
15 Tq	1	10	4	1	73.33
	1	11	3	1	80.00
16 Tq	1	10	5	1	68.75
	1	11	4	1	75.00
20 Tq	1	12	7	1	65.00
	1	13	6	1	70.00
24 Tq	1	15	8	1	66.66
	1	16	7	1	70.83
50 Tq	1	39	10	4	80.00

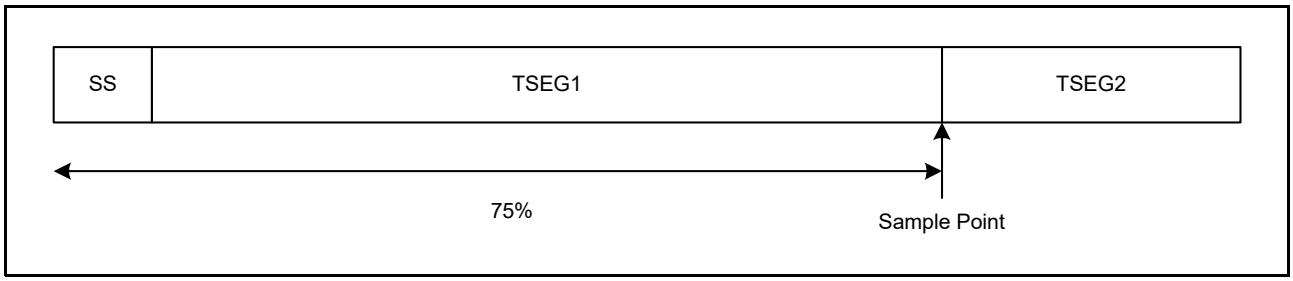


Figure 36.14 The Sample Point (in Case of 75%)

36.4.1.2 Bit Timing

In the CAN protocol, each bit in a communication frame is composed of three segments that can be configured using the NBCR and DBCR registers.

Figure 36.15 shows the segment composition of a bit and the sample point in it.

Of these segments, the Time Segment 1 (TSEG1) and Time Segment 2 (TSEG2) are used to specify the position of the sample point, so that the timing at which each bit on the CAN bus is sampled can be altered by changing the values of these segments.

The minimum resolution for this timing is referred to as Time Quantum (Tq), which is determined by the clock frequency supplied to the CAN channel and the divide-by-N value of the bit rate prescaler (nominal and data bit rate).

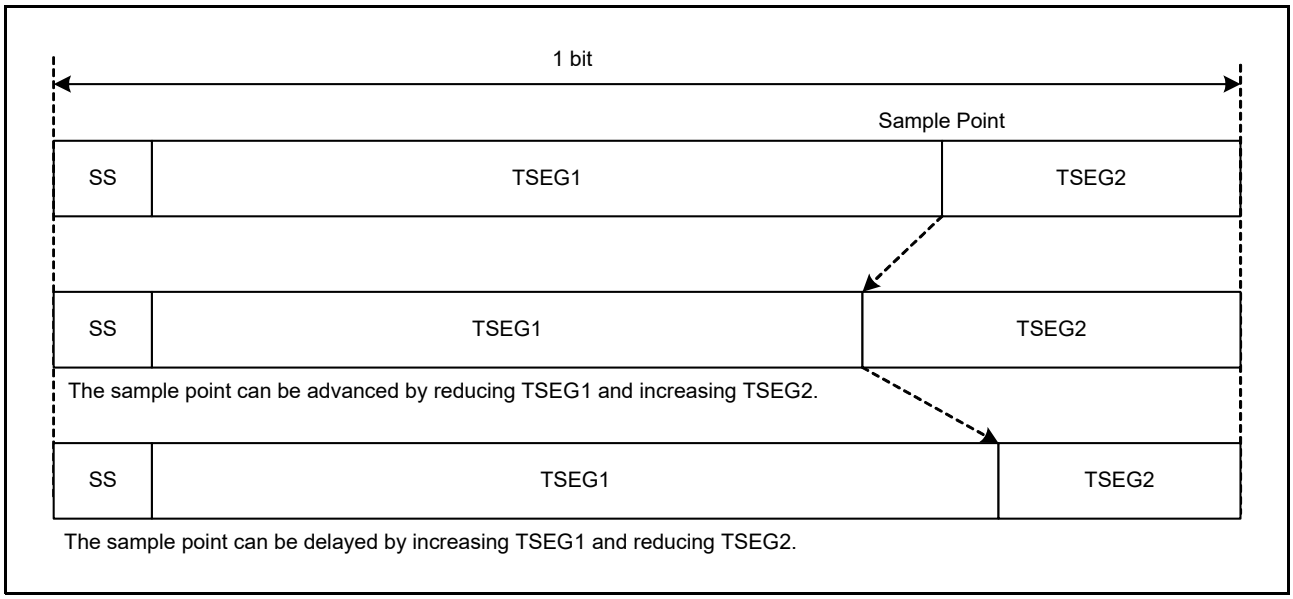


Figure 36.15 Segment Composition of A Bit and The Sample Point

1. SS: Synchronization Segment
   This segment is used to synchronize bits by monitoring a recessive-to-dominant edge during the interframe space. This comprises of intermission, suspend transmission, bus idle, during bus idle, and all nodes that can start transmission.
2. TSEG1: Time Segment 1
   This segment absorbs physical delays on the CAN network. A physical delay on the network is two times the total sum of a bus delay, input comparator delay, and output driver delay. It can be lengthened by SJW.
3. TSEG2: Time Segment 2
   This segment is used to correct a phase error by performing resynchronization. It can be shortened by SJW. While sending or receiving a message, communication frames between some nodes may get out of sync due to a drift in

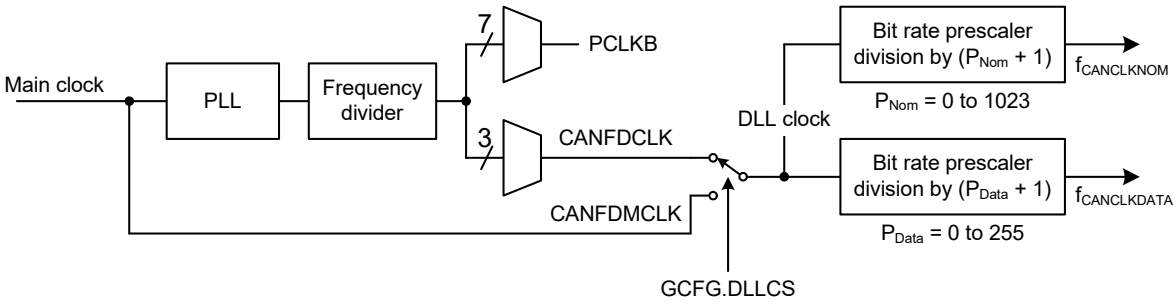
the oscillator frequency or a delay in the transmission path. This is referred to as a phase error.

4. SJW: Resynchronization Jump Width
   This is the maximum width by which bits that have become out of sync due to a phase error may be corrected.

Figure 36.15 shows an example of a typical sample point.

36.4.1.3 Bit Rate

The CAN communication clocks are generated by dividing the operating clock for data link layer (DLL clock). The DLL clock can be selected from either the internal clock (CANFDCLK) or the external clock (CANFDMCLK).
Figure 36.16 shows a block diagram of the circuit that generates the CAN communication clock.



Main clock:	External oscillator clock
CANFDCLK:	Clock equivalent to peripheral module clock
DLL clock:	Data Link Layer clock DLL clock ≤ PCLKB
PCLKB:	CANFD module system clock
P_{Nom}, P_{Data}	Value selected by the NBCR.BRP[9:0] bits and DBCR.BRP[7:0] bits
$f_{CANCLKNOM}$:	CAN nominal communication clock, f(CANCLKNOM) = (DLL clock) / (P_{Nom} + 1)
$f_{CANCLKDATA}$:	CAN data communication clock, f(CANCLKDATA) = (DLL clock) / (P_{Data} + 1)

Figure 36.16 Block Diagram of the Circuit that Generates the CAN Communication Clock

The bit rate is determined by the DLL clock frequency, the division value of the bit rate prescaler (P + 1), and the number of Tq per bit.

$$\text{Bit rate} = \frac{\text{DLL clock frequency}}{\text{Number of Tq per bit} \times (\text{P} + 1)} = \frac{\text{CAN communication clock frequency}}{\text{Number of Tq per bit}}$$

Table 36.17 lists examples of setting the nominal bit rate for Classical CAN frame.

Table 36.17 Nominal Bit Rate Setting Examples for Classical CAN Frame

Bit Rate	DLL Clock Frequency															
	60 MHz		40 MHz		32 MHz		30 MHz		24 MHz		20 MHz		16 MHz		10 MHz		8 MHz*1	
	No. of Tq	P+1	No. of Tq	P+1	No. of Tq	P+1	No. of Tq	P+1	No. of Tq	P+1	No. of Tq	P+1	No. of Tq	P+1	No. of Tq	P+1	No. of Tq	P+1
1 Mbps	10 Tq 15 Tq	6 4	8 Tq 20 Tq	5 2	8 Tq 16 Tq	4 2	10 Tq 15 Tq	3 2	8 Tq 12 Tq 24 Tq	3 2 1	10 Tq 20 Tq	2 1	8 Tq 16 Tq	2 1	10 Tq	1	8 Tq	1
500 kbps	10 Tq 15 Tq 20 Tq	12 8 6	8 Tq 20 Tq	10 4	8 Tq 16 Tq	8 4	10 Tq 15 Tq 20 Tq	6 4 3	8 Tq 12 Tq 24 Tq	6 4 2	10 Tq 20 Tq	4 2	8 Tq 16 Tq	4 2	10 Tq 20 Tq	2 1	8 Tq 16 Tq	2 1
250 kbps	10 Tq 15 Tq 20 Tq	24 16 12	8 Tq 20 Tq	20 8	8 Tq 16 Tq	16 8	10 Tq 15 Tq 20 Tq	12 8 6	8 Tq 12 Tq 24 Tq	12 8 4	10 Tq 20 Tq	8 4	8 Tq 16 Tq	8 4	10 Tq 20 Tq	4 2	8 Tq 16 Tq	4 2
125 kbps	10 Tq 15 Tq 20 Tq	48 32 24	8 Tq 20 Tq	40 16	8 Tq 16 Tq	32 16	10 Tq 15 Tq 20 Tq	24 16 12	8 Tq 12 Tq 24 Tq	24 16 8	10 Tq 20 Tq	16 8	8 Tq 16 Tq	16 8	10 Tq 20 Tq	8 4	8 Tq 16 Tq	8 4
83.3 kbps	8 Tq 10 Tq 12 Tq 15 Tq 20 Tq 24 Tq	90 72 60 48 36 30	8 Tq 12 Tq 16 Tq 24 Tq	60 40 30 20	8 Tq 12 Tq 16 Tq 24 Tq	48 32 24 16	8 Tq 10 Tq 12 Tq 15 Tq 20 Tq 24 Tq	45 36 30 24 18 15	8 Tq 12 Tq 16 Tq 24 Tq	36 24 18 12	8 Tq 10 Tq 12 Tq 15 Tq 16 Tq 20 Tq 24 Tq	30 24 20 16 15 12 10	8 Tq 12 Tq 16 Tq 24 Tq	24 16 12 8	8 Tq 10 Tq 12 Tq 15 Tq 20 Tq 24 Tq	15 12 10 8 6 5	8 Tq	12
33.3 kbps	10 Tq 12 Tq 15 Tq 20 Tq	180 150 120 90	8 Tq 12 Tq 16 Tq 20 Tq 24 Tq	150 100 75 60 50	8 Tq 10 Tq 12 Tq 15 Tq 16 Tq 20 Tq 24 Tq	120 96 80 64 60 48 40	10 Tq 12 Tq 15 Tq 20 Tq	90 75 60 45	8 Tq 10 Tq 12 Tq 15 Tq 16 Tq 20 Tq 24 Tq	90 72 60 48 45 36 30	8 Tq 10 Tq 12 Tq 15 Tq 20 Tq 24 Tq	75 60 50 40 30 25	8 Tq 10 Tq 12 Tq 15 Tq 16 Tq 20 Tq 24 Tq	60 48 40 32 30 24 20	10 Tq 12 Tq 15 Tq 20 Tq	30 25 20 15	8 Tq	30

Note 1. Minimum frequency to achieve a nominal bit rate of 1 Mbps.

For optimum clock tolerance in networks using the CAN FD frame, the length of the time quantum should be the same in nominal bit time and in data bit time. This means NBCR.BRP[9:0] = DBCR.BRP[7:0].
In addition, do not set the DBCR.BRP[7:0] bits greater than 1 when using transceiver delay compensation.
Table 36.18 lists examples of setting the nominal and data bit rate for CAN FD frame.

Table 36.18 Nominal and Data Bit Rate Setting Examples for CAN FD Frame

Bit Rate		DLL Clock Frequency											
		60 MHz			40 MHz			30 MHz			20 MHz		
		No. of Tq			No. of Tq			No. of Tq			No. of Tq		
Nominal	Data	Nom.	Data	P+1	Nom.	Data	P+1	Nom.	Data	P+1	Nom.	Data	P+1
1 Mbps	8 Mbps	—	—	—	40 Tq	5 Tq	1	—	—	—	—	—	—
1 Mbps	5 Mbps	60 Tq	12 Tq	1	40 Tq	8 Tq	1	30 Tq	6 Tq	1	—	—	—
1 Mbps	4 Mbps	60 Tq	15 Tq	1	40 Tq	10 Tq	1	—	—	—	20 Tq	5 Tq	1
1 Mbps	2 Mbps	60 Tq	30 Tq	1	40 Tq	20 Tq	1	30 Tq	15 Tq	1	20 Tq	10 Tq	1
500 kbps	2 Mbps	120 Tq	30 Tq	1	80 Tq	20 Tq	1	60 Tq	15 Tq	1	40 Tq	10 Tq	1

36.4.1.4 Setting of CAN Clock, Bit Timing and Bit Rate

Figure 36.17 shows the procedure for setting the bit timing and the bit rate.
These settings should be configured when the CAN channel is in CH_RESET mode.
The bit rate must be configured before moving to the channel communication state. Otherwise the mode does not switch correctly.

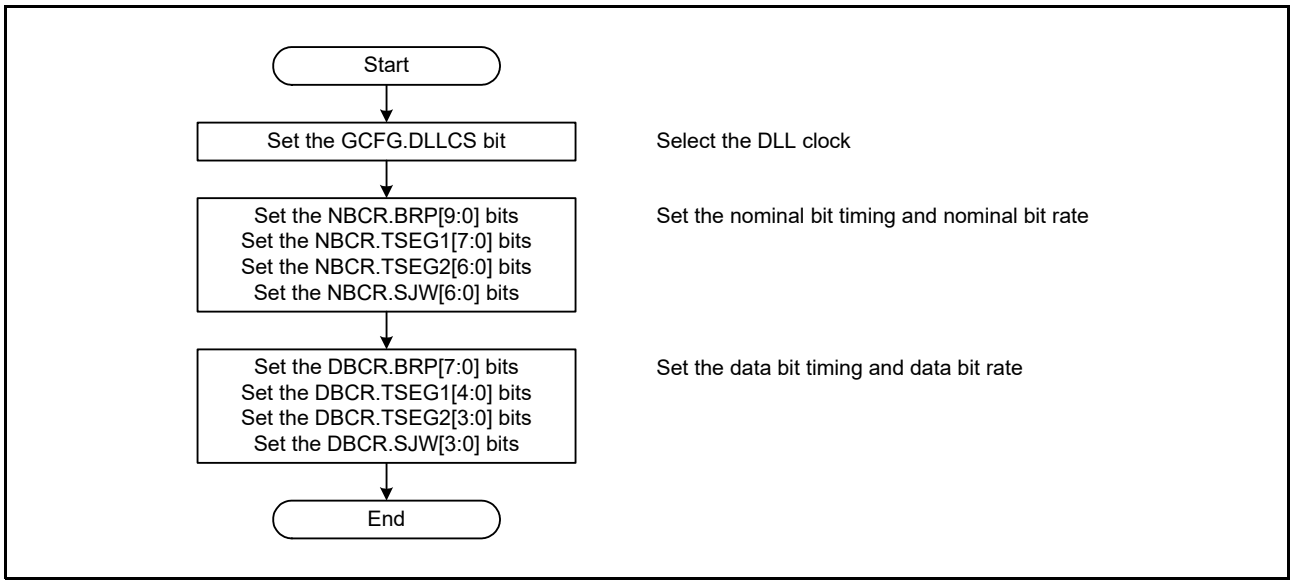


Figure 36.17 Procedure for Setting the Bit Timing and Bit Rate

36.4.1.5 Transceiver Delay Compensation

When a high bit rate is used such as 5 to 8 Mbps for the data phase, the transceiver delay can become greater than TSEG1. In this case, the transmitter always detects a bit-error in the data phase of the CAN FD frame. The TDC compensates for the inability of the transmitter to receive its own transmitted bit at the sample point of that bit.
There is another symbolic sample point known as the Secondary Sample Point (SSP) that is used only during the data phase of CAN FD frames. This is derived from the Transceiver Delay Compensation Result bit (FDSTS.TDCR[7:0]) as shown in Figure 36.18.
The resolution of the configuration, measured and offset values is based on the CAN channel DLL clock.

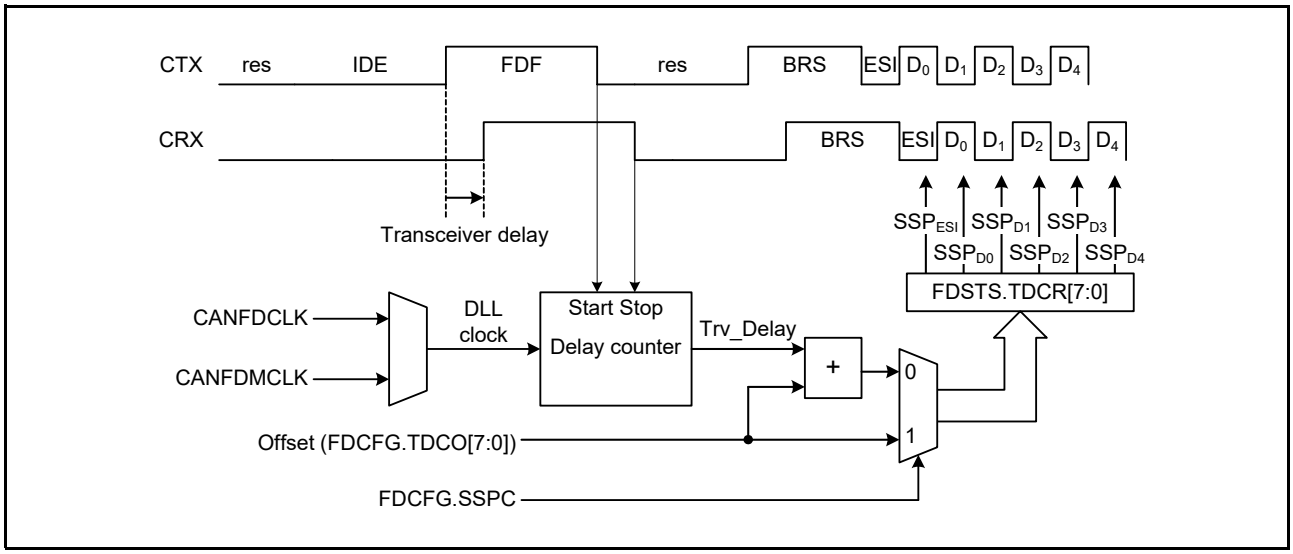


Figure 36.18 Transceiver Delay Compensation

The measured Trv_Delay is based on the number of DLL clock cycles. It counts up by one for each start clock until the dominant value can be observed at the CRX0 pin. Figure 36.19 shows the measurement example. Trv_Delay is counted up to 127 on each DLL clock.

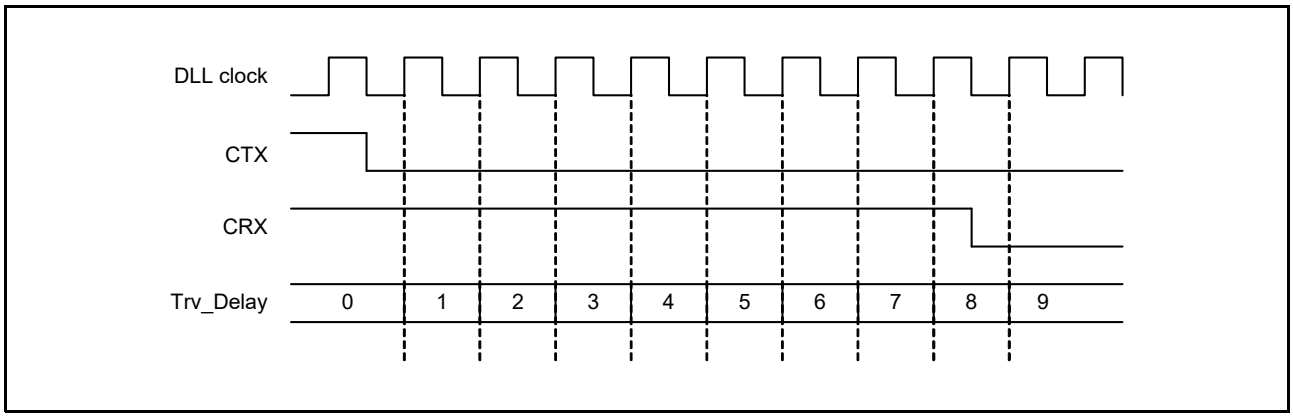


Figure 36.19 Trv_Delay Measurement Example

The SSP is calculated by taking the result from the FDSTS.TDCR[7:0] bits and rounding the value down to the nearest integer number of data Tq.
Figure 36.20 shows the positioning of the secondary sample point (SSP). When the FDCFG.SSPC bit is set to 0, the SSP is equal to the Trv_Delay (measured delay) + FDCFG.TDCO[7:0], rounded down to the nearest integer number of Tq. Normally, the TDCO[7:0] value has the magnitude of SS + TSEG1 in the data phase to position the SSP to a theoretical location of the sample point.
If the FDCFG.SSPC bit is set to 1, the SSP is defined by the FDCFG.TDCO[7:0] bits. If the DBCR.BRP[7:0] bits are greater than 00h, the value is also rounded down to the nearest integer number of Tq.

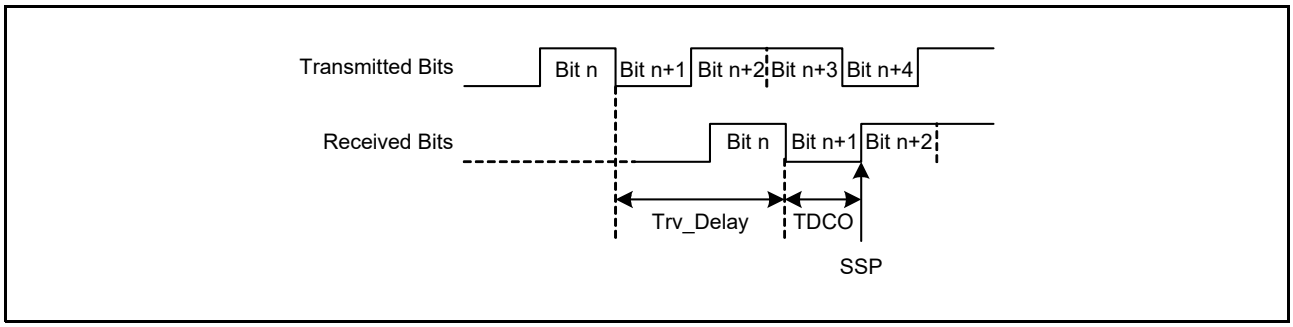


Figure 36.20 Positioning of the Secondary Sample Point (SSP)

The maximum delay time (Trv_Delay + TDCO[7:0]) which can be compensated by the CANFD module is (6 data bits – 2 DLL clock). The ISO 11898-1 allows you to set different values for BRP_data and BRP_nom.
If different values are used for the NBCR.BRP[9:0] bits and DBCR.BRP[7:0] bits, then two CAN nodes may be out of synchronization at the point when the bit rate changes from nominal bit rate to data bit rate after sample point of the BRS bit. This condition is shown in Figure 36.21.
The length of the time quantum should be the same in the nominal bit time and in the data bit time. This means NBCR.BRP[9:0] = DBCR.BRP[7:0].
The bit rate can be changed by selecting different settings for the time segments. The nominal bit rate can be set from 8 to 385 Tqs and the data bit rate from 5 to 49 Tqs.

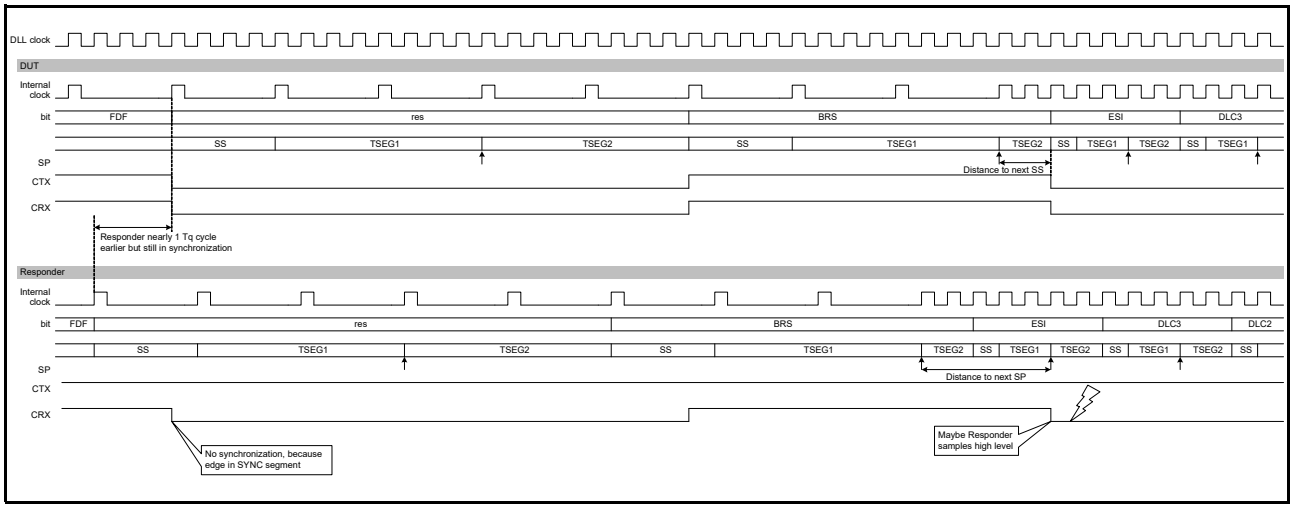


Figure 36.21 Loss of Synchronization Between Two CAN Nodes

The transceiver delay compensation measurement result is updated at the falling edge from FDF bit to res bit when configured accordingly (FDCFG.TDCE = 1, FDCFG.SSPC = 0).
Figure 36.22 shows the read flow to get the measured transceiver delay compensation result.

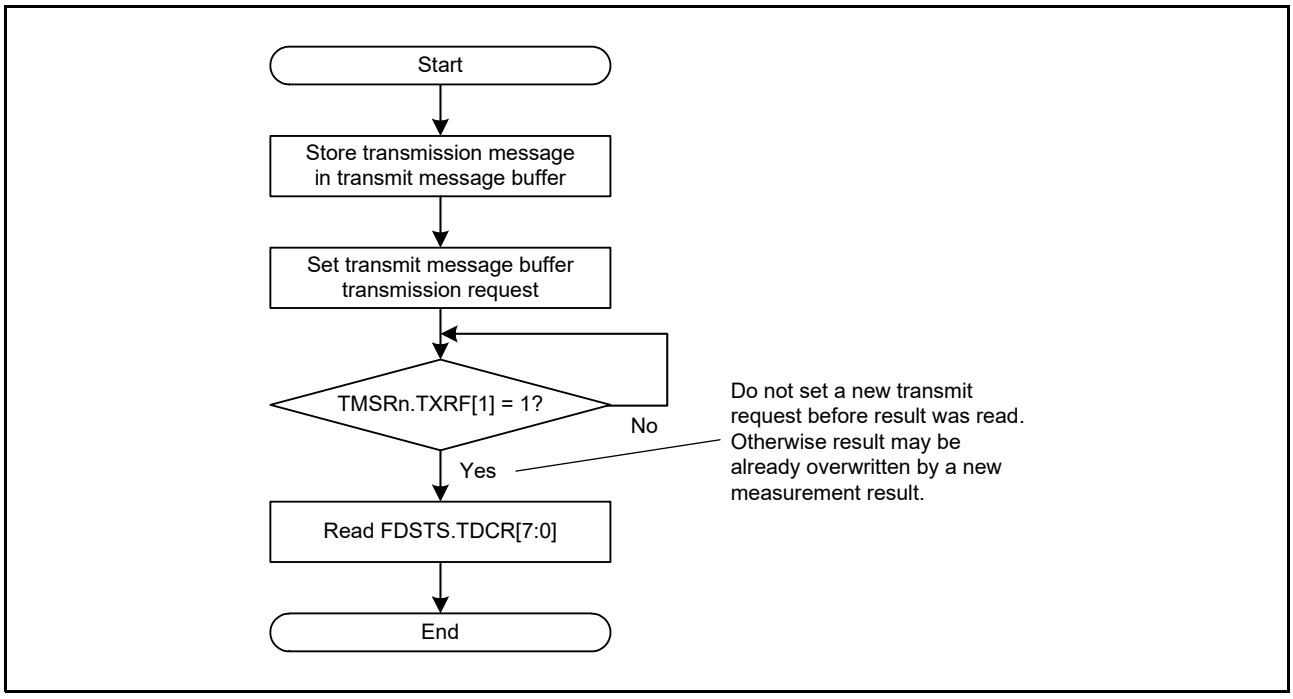


Figure 36.22 Transceiver Delay Compensation Result Read Flow

36.4.2 CANFD Module Configuration after a Reset

After release from a MCU reset or after setting and clearing the GRCR.SRST bit, the CANFD module enters GL_SLEEP mode automatically.

To configure the CANFD module settings, exit GL_SLEEP mode by setting the GCR.SLPRQ bit to 0.

After release from a MCU reset, the CANFD module starts RAM initialization. At this time, the GSR.RAMST flag is automatically set to 1 indicating that the CANFD module logic is initializing the RAM.

After RAM initialization is complete, this bit is automatically set to 0.

RAM initialization is necessary to prevent indefinite data in RAM from detecting the wrong ECC error after release from a MCU reset.

Do not access (read or write) other CANFD registers until the RAM initialization is complete and the GSR.RAMST flag is becomes 0.

Before going to communication mode, the Acceptance Filter List and message FIFO buffers must be configured. In addition, the CAN channel setting such as CAN bit timing must be configured. To make this configuration, the CAN channel must be released from CH_SLEEP mode and must be set for communication in CH_RESET mode (configuration mode).

Figure 36.23 shows the configuration procedure. For details about each step, refer to section 36.5, Filtering Using Acceptance Filter List (AFL), section 36.6, FIFO Buffers and Message Buffer Configuration, section 36.10, Interrupts and DTC/DMA Requests, and section 36.4.1.3, Bit Rate.

The CANFD module does not perform the RAM initialization sequence after executing a software reset by setting the GRCR.SRST bit to 1.

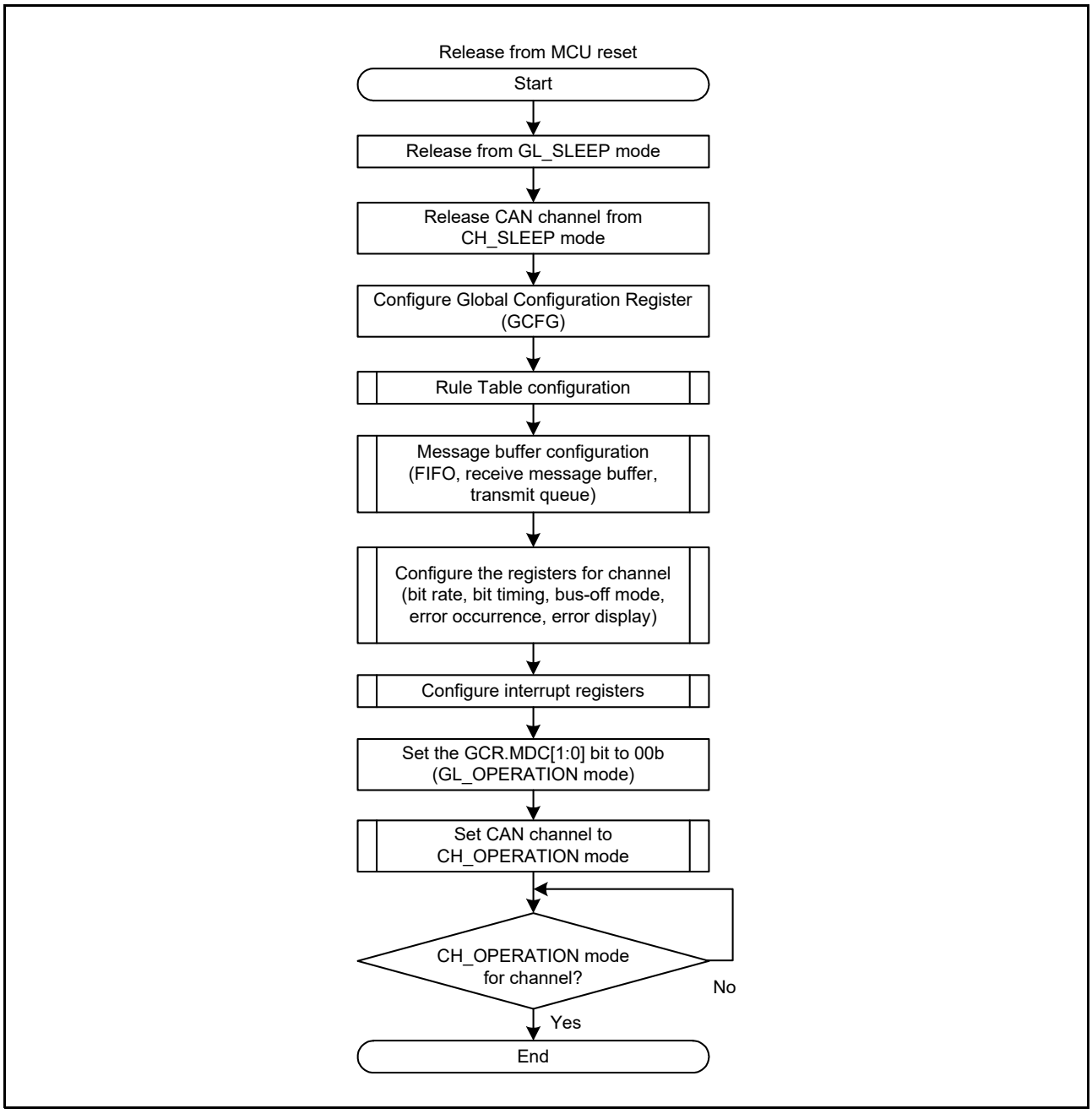


Figure 36.23 Configuration Procedure After A MCU Reset

36.5 Filtering Using Acceptance Filter List (AFL)

The CANFD module allows you to use the Acceptance Filter List (AFL) to filter message acceptance for channels. Each entry of the AFL defines a filter rule for received messages.
The followings are performed based on the AFL entries:

- Acceptance filtering based on the RTR value, IDE value, and ID value of received message
- DLC filtering based on DLC value of received message
- Payload overflow based on the GCFG.OMRC bit
- Storage of accepted messages in the specified message buffer/FIFO buffer
- Attaching a 16-bit pointer to the stored messages, for example to support AUTOSAR applications
- Attaching a 2-bit information label to the stored messages.

The CANFD module allows a maximum of 32 AFL entries.

36.5.1 Acceptance Filtering Process

Acceptance filtering matches each AFL entry with the received message. Matching starts with the lowest AFL entry number.
The AFL search stops when the received message ID matches the specified ID/mask combination, or when the received message ID has been matched against all defined AFL entries. If no match occurs, then the received message is discarded. No notification is given to the application in this case.

36.5.2 DLC Filtering Process

DLC filtering is performed for each accepted message if DLC check is enabled (GCFG.DCE bit = 1). If the DLC value of the received message is equal to or higher than the DLC value specified for the AFL entry whose ID matches in the Acceptance Filtering process (the matching AFL entry), the DLC check is passed.
If DLC replacement (GCFG.DRE bit) is enabled, DLC value configured in the matching AFL entry is greater than 0000b and DLC check passes, then the configured value of DLC in the matching AFL entry is stored in the destination receive message buffer (RMBn) or FIFO buffer. If DLC value of the received message is greater than the DLC value specified in the matching AFL entry, the excess data bytes are not stored in the RMBn/FIFO buffer. These excess data bytes are stored as 00h in the RMBn/FIFO buffer.
If DLC replacement is enabled and the DLC value specified in the matching AFL entry is 0000b (DLC filter function is disabled), the DLC value of the received message is stored in the RMBn/FIFO buffer.
If DLC replacement is disabled (GCFG.DRE bit = 0) and DLC check passes, the DLC value of the received message is stored in the RMBn/FIFO buffer. If the DLC value of the received message is greater than the DLC value specified in the matching AFL entry, the excess data bytes are also stored in the RMBn/FIFO buffer.
If DLC value of the received message is less than the DLC value specified in the matching AFL entry, DLC check fails. In this case, the received message is discarded and is not stored in anywhere.
Additionally, when the DLC check fails, the GESR.DEDF flag is set to 1. If interrupts are enabled, an error interrupt is also generated. If the DLC check fails, the DLC replacement setting has no effect.

36.5.3 Message Storage

If a received message has passed both acceptance filtering and DLC filtering, the message is stored in receive message buffers 0 to 31, receive FIFO 0, 1, or common FIFO 0 configured in receive FIFO mode.
This message storage target information is also defined in the AFL entry. Do not set a target at the AFL entry which is not configured.
Up to two message storage destinations can be specified. Do not specify more than 3 destinations.

36.5.4 Payload Overflow Process

There is a protection mechanism in case the received message contains data with a payload size that is longer than the size that can be stored in the storage (RMCR.PLS[2:0], RFCR0.PLS[2:0], RFCR1.PLS[2:0], or CFCR0.PLS[2:0]).
If GCFG.OMRC = 0 (message is discarded), the message with data byes that exceed the specified payload size are discarded and not stored. In this case, even if the FIFO is full, the corresponding FMLSR.RFML0, RFML1, or CFML0 flag is not set to 1.
If GCFG.OMRC = 1 (cut to specified size), only data bytes that exceed the specified payload size are discarded. In this case, if the FIFO is full, the corresponding FMLSR.RFML0, RFML1, or CFML0 flag is set to 1 (message lost has occurred).
Depending on the GCFG.DRE bit setting, either the DLC value of the received message or the DLC value specified in the AFL entry is stored.
Regardless of the GCFG.OMRC bit setting, the GESR.PODF flag is set to 1 if a payload overflow condition is detected.
The DLC filtering is performed before the payload overflow process. So for one reception frame, only one flag can be set to 1 at the same time with the GESR.DEDF flag or GESR.PODF flag.

36.5.5 Allocation of AFL Entries

The number of AFL entries (number of rules) can be configured using the AFCFG.RN0[5:0] bits (refer to Figure 36.24).

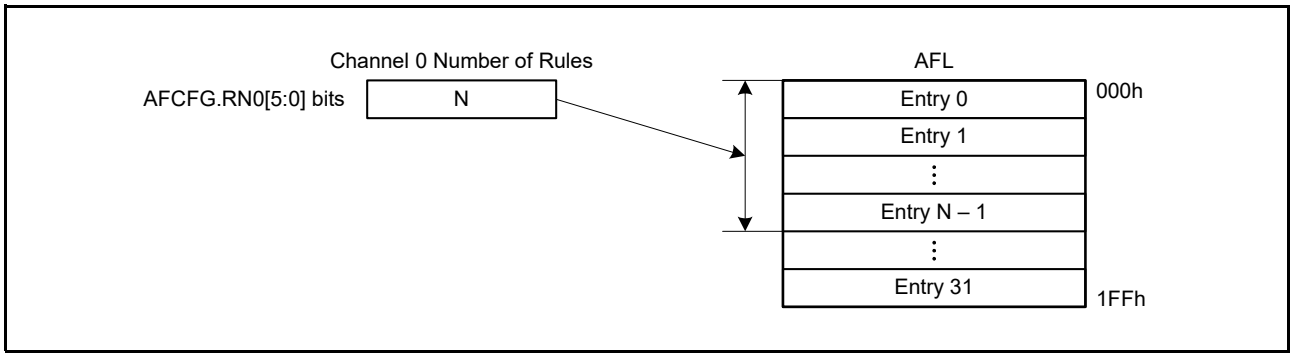


Figure 36.24 Configuration of AFL

The minimum number of entries for one channel is 0 (no entries defined for the channel) and the maximum number of entries for one channel is 32.
The CANFD module does not flag errors related to the configuration of the AFL.

36.5.6 AFL Entry Description

Each AFL entry consists of 16 bytes. The fields in all entries are identical.
Each entry contains the following information for acceptance filtering and DLC filtering:

- Identifier field (11 bits for Standard Frame format, 29 bits for Extended Frame format):
  Acceptance filter unit matches this filed with the identifier field of the received message (29 bits of identifier field can be masked individually, refer to the description of Mask for Identifier field below).
- IDE bit:
  Acceptance filter unit matches this bit with the IDE bit of the received message and selects the relevant part of the identifier field for acceptance filtering (masking of IDE bit is possible, refer to the description of Mask for IDE bit below).
- RTR bit:
  Acceptance filter unit only accepts data frames (RTR = 0) or remote frames (RTR = 1) according to the setting of this bit (masking of RTR bit is possible, refer to the description of Mask for RTR bit below).
- Loopback Configuration bit:
  This bit can enable or disable the AFL entry depending on the Loopback Configuration or Mirror mode condition.
- Mask for Identifier field (29 bits):
  Each bit in the identifier mask field can mask the corresponding identifier bit in the AFL entry (refer to Figure 36.25).
- Mask for IDE bit:
  If this Mask bit masks the IDE bit of an AFL entry, the AFL entry can accept messages in both Standard Identifier format and Extended Identifier format. For messages in Standard Identifier format, the Standard Identifier part of the AFL entry is compared, and for messages in Extended Identifier format, the Extended Identifier part of the AFL entry is compared.
- Mask for RTR bit:
  If this Mask bit masks the RTR bit of an AFL entry, the AFL entry can accept frame formats in both data frame and remote frame.
- Pointer (16 bits):
  This 16-bit pointer is attached to a received message accepted by the related AFL entry. The pointer is added when storing a message in the message buffer area and can be used as a support function in the application.
  For example, the pointer information can be used to support the assignment of PDU IDs to the received message in AUTOSAR systems.
- Information label (2 bits):
  This 2-bit label is attached to a message accepted by the related AFL entry. This label is added when storing a message in the message buffer area and can be used as support function in the application.
- DLC field:
  If the DLC value of the received message is equal to or higher than the value set in this field, the DLC check is passed.
  If the DLC value of an AFL entry is set to 0000b, the DLC filtering for this entry is effectively disabled (all accepted messages pass DLC filtering).

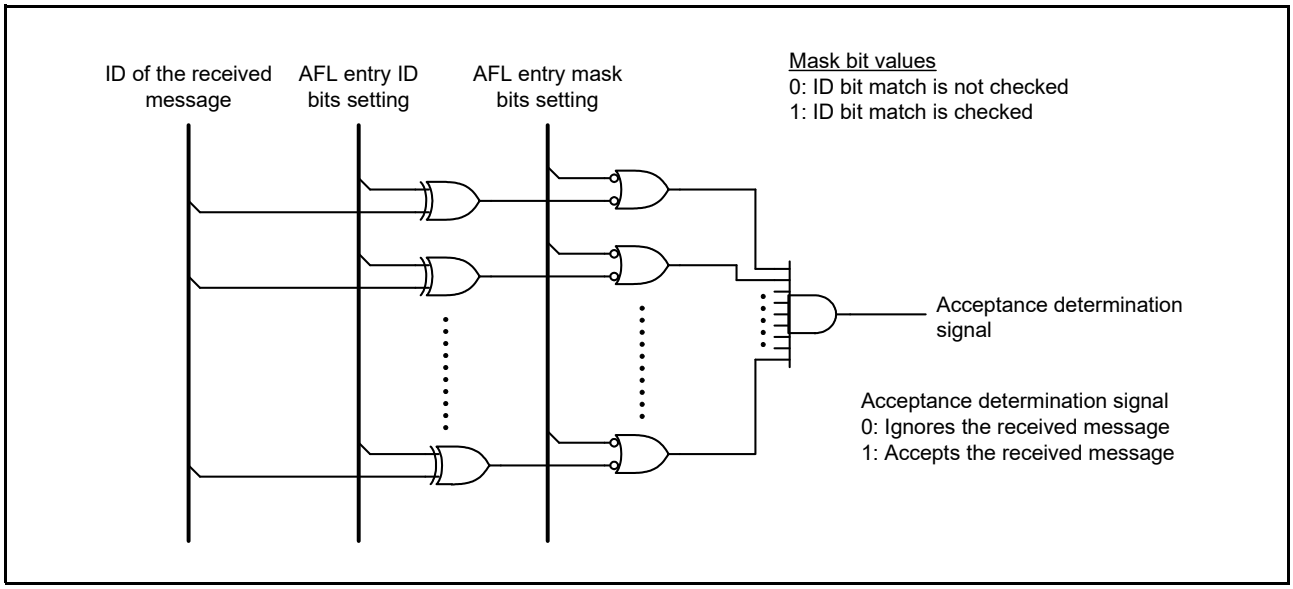


Figure 36.25 Acceptance Filter Function

Each AFL entry contains the following information for processing the received messages:

- Message buffer number of the receive message buffer used as the storage destination for received messages
- The Destination Message Buffer Setting Enable bit to specify the receive message buffer as the storage destination for received messages
- The FIFO Destination Enable bit to specify the FIFO as the storage destination for received messages.

There is no protection function for storing messages. Therefore, the FIFOFIFO Destination Enable bit must be set carefully.

36.5.7 Entering Entries in the AFL

One complete entry can be entered into AFL via the following registers:

- AFLn.IDR register: First part of the AFL entry
- AFLn.MASK register: Second part of the AFL entry
- AFLn.PTR0 register: Third part of the AFL entry
- AFLn.PTR1 register: Fourth part of the AFL entry.

These 16 sets of registers make up one page of AFL entries. There are 32 entries in the CANFD module, and all of these entries can be accessed by specifying the page with the AFCR.PAGE bit. The AFL should only be configured in CH_RESET or CH_HALT mode. Table 36.19 shows pages linked to the AFL entries.

Table 36.19 Pages Linked to the AFL Entries

Page	Linked AFL Entries
Page 0	Entry 0 to 15
Page 1	Entry 16 to 31

AFL access control is performed using the AFCR register (Figure 36.26). This register has the following bits:

- The PAGE bit for selecting the AFL page number
- The AFLWE bit that enables or disables the writing of data to prevent unnecessary write access to the AFL.

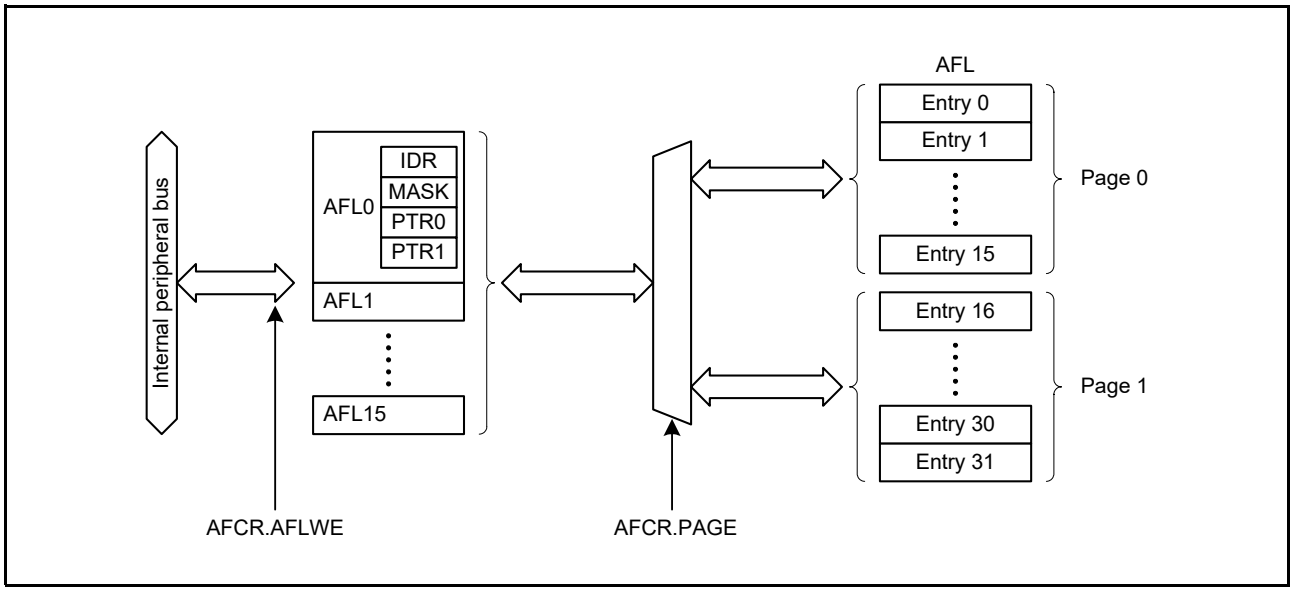


Figure 36.26 AFL Page Access

Follow the flow shown in Figure 36.27 to configure the AFL.
After entering all the entries, writing to the AFL must be disabled to prevent unnecessary write access to the AFL.
If the AFCR.AFLWE bit is set to 0, write protection is enabled during all Global modes (GL_RESET, GL_HALT, and GL_OPERATION).
Even if the AFCR.AFLWE bit is set to 0, reading from AFL is possible during all Global modes (the consistency of AFL contents can be checked during execution).

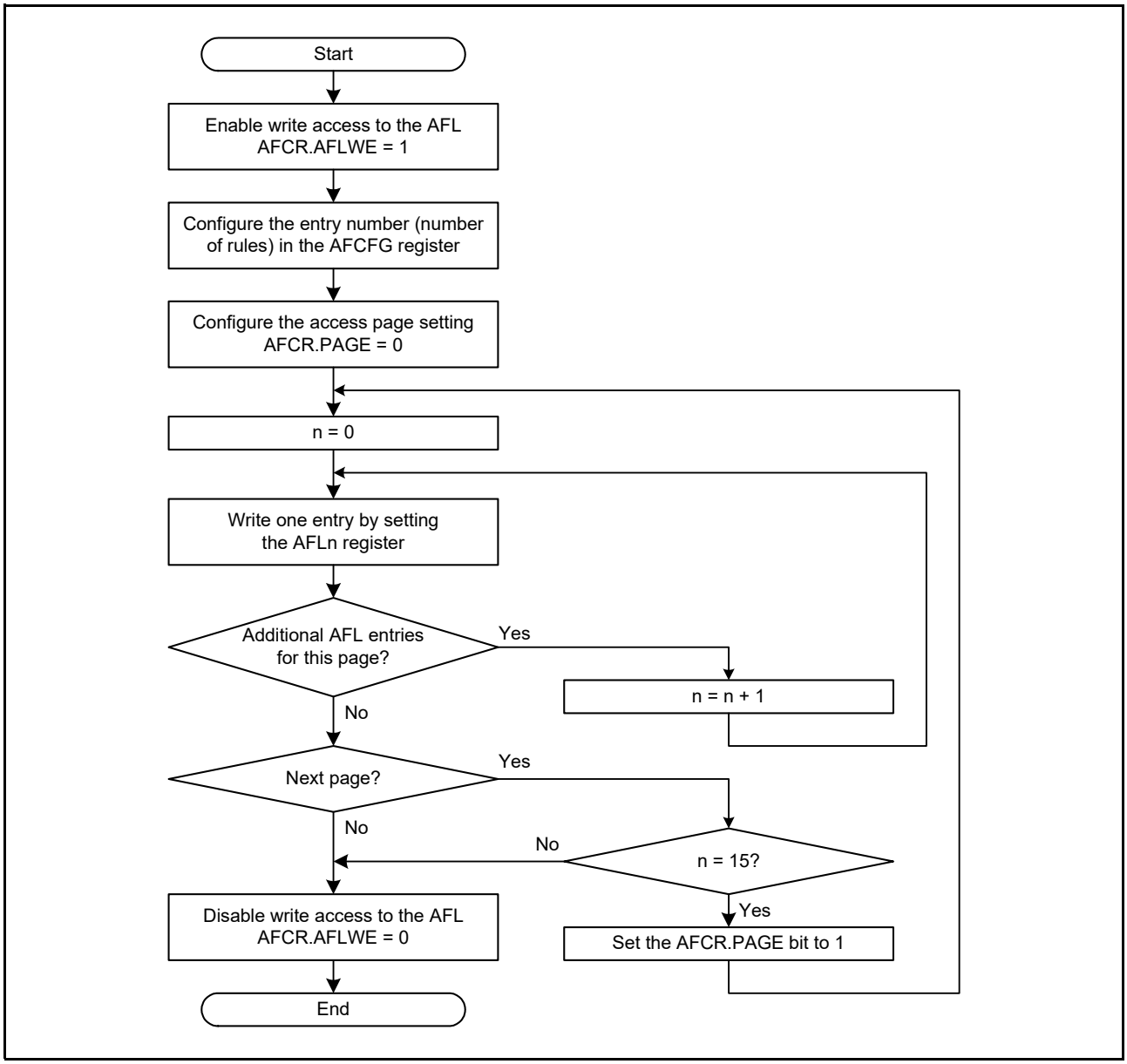


Figure 36.27 AFL Configuration Flow

36.5.8 Loopback Modes

AFL entries with the AFLn.IDR.LPC bit set to 1 are used only in loopback mode (self-test mode 0 or self-test mode 1) or mirror mode. If a message transmitted by another node on the CAN bus is received while in loopback mode, the AFL entry will not be used.
AFL entries with the AFLn.IDR.LPC bit set to 0 are used only for:

- Received messages transmitted by other nodes in normal (non-loopback mode) and mirror modes
- Received messages transmitted by other nodes or the local node in loopback mode.

The mirror mode can be enabled with the GCFG.MME bit. If the message is successfully transmitted when the GCFG.MME bit is 1, the message is stored in the received message buffer or FIFO buffer if there is a matching entry in the AFL. The AFLn.IDR.LPC bit in the matching AFL entry must be set to 1 to store this frame.
If mirror mode and loopback mode are set at the same time, the loopback mode behavior is applied. Table 36.20 shows the behavior of the acceptance filter unit depending on the setting of the related input signals.

Table 36.20 Behavior of Acceptance Filter Based on the Loopback Setting in AFL Entry

Mirror Mode (MME Bit)	Loopback Mode (Self-test Mode 0 or Self-test Mode 1)	Channel Mode	LPC Bit	AFL Entry
0	0	Receiver	0	Valid
			1	Invalid
		Transmitter	0	Invalid
			1	Invalid
	1	Receiver	0	Valid
			1	Invalid
		Transmitter	0	Valid
			1	Valid
1	0	Receiver	0	Valid
			1	Invalid
		Transmitter	0	Invalid
			1	Valid
	1	Receiver	0	Valid
			1	Invalid
		Transmitter	0	Valid
			1	Valid

Note: The expression valid or invalid for the related entry means that this AFL entry is or is not compared against the received message ID, respectively.

36.5.9 IDE Masking

The IDE bit set in the AFL entry with the AFLn.MASK.IDEM bit set to 0 is not used for ID matching. In this case, the use of ID[10:0] or ID[28:0] matching is selected based on the received IDE bit.
The following lines show the examples.

- The ID and mask fields of an AFL entry x is configured as follows:
  – AFLx.IDR = C0553A20h → IDE = 1, RTR = 1, LPC = 0, ID[10:0] = 220h/ID[28:0] = 0553A20h
  – AFLx.MASK = 0000FFFFh → IDEM = 0. RTRM = 0, IDM[10:0] = 7FFh/IDM[28:0] = 0000FFFFh
- The comparison result for the four different received IDs with AFL entry x is described as follows:
  – If a frame with IDE = 0 and ID = 220h is received, this is considered as a match
  – If a frame with IDE = 0 and ID = 320h is received, this is not a match
  – If a frame with IDE = 1 and ID = 1FFF3A20h is received, this is considered as a match
  – If a frame with IDE = 1 and ID = 08803220h is received, this is not a match.

36.5.10 Updating AFL Entry during Communication

The AFL entry can be updated without disabling CAN communications.
Set the AFL entry number to update to the ignore entry select bits and set the ignore entry enable bit to 1.
The entry number specified here is ignored by AFL matching while the entry is being updated.
Figure 36.28 shows the update flow for an AFL entry.

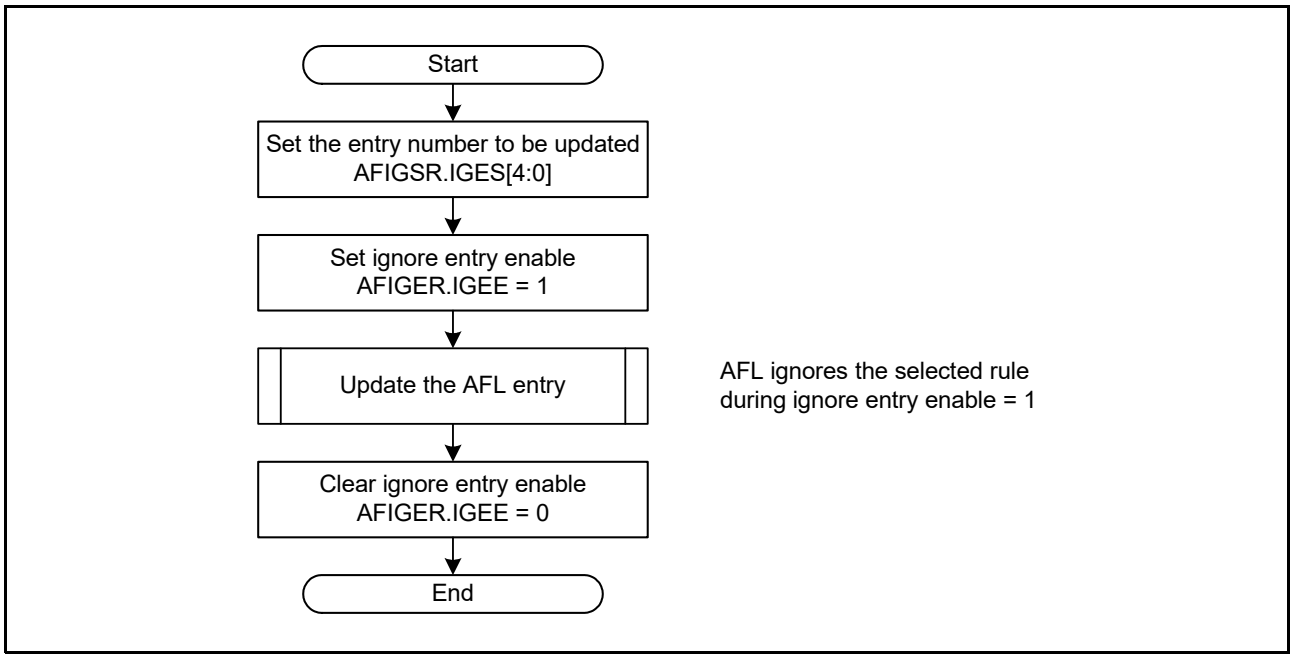


Figure 36.28 Update Flow for An AFL Entry

The method to update an AFL entry is as follows:

1. Set the entry number*1 to be updated to the AFIGSR.IGES[4:0] bit.
2. Set the value 0000C401h (key code and ignore entry enabled) to the AFIGER register.
3. Set the page number which includes the entry to be updated to the AFCR.PAGE bit. Set the AFCR.AFLWE bit to 1.
4. Set the new rule to the AFLn.IDR, AFLn.MASK, AFLn.PTR0, AFLn.PTR1 registers.
5. The AFCR.AFLWE bit is set to 0.
6. Set the value 0000C400h (key code and ignore entry disabled) to the AFIGER register.

Note 1. This entry number is not used for acceptance filtering between (2) and (5).

(1) Example 1: Deleting an Entry

The following describes how to delete entry 3 when the total number of entries is 6.

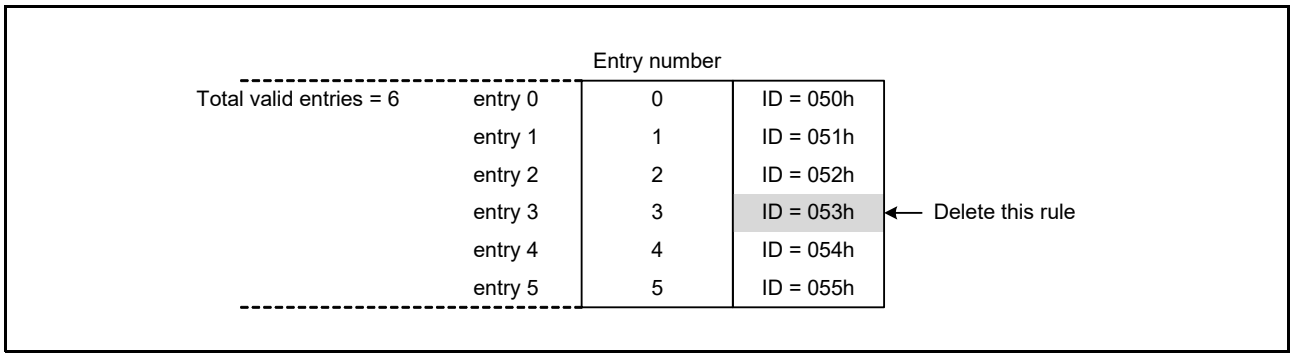


Figure 36.29 Example of Deleting an Entry (Before Deleting Entry 3)

[How to delete an entry]

(1) Set 00000003h to the AFIGSR register.
(2) Set 0000C401h to the AFIGER register.
(3) Set 00000100h to the AFCR register.
(4) Set the same rule as the previous rule by accessing the AFL3.IDR, AFL3.MASK, AFL3.PTR0, AFL3.PTR1 registers.
(5) Set 00000000h to the AFCR register.
(6) Set 0000C400h to the AFIGER register.

Entry 3 is now deleted.

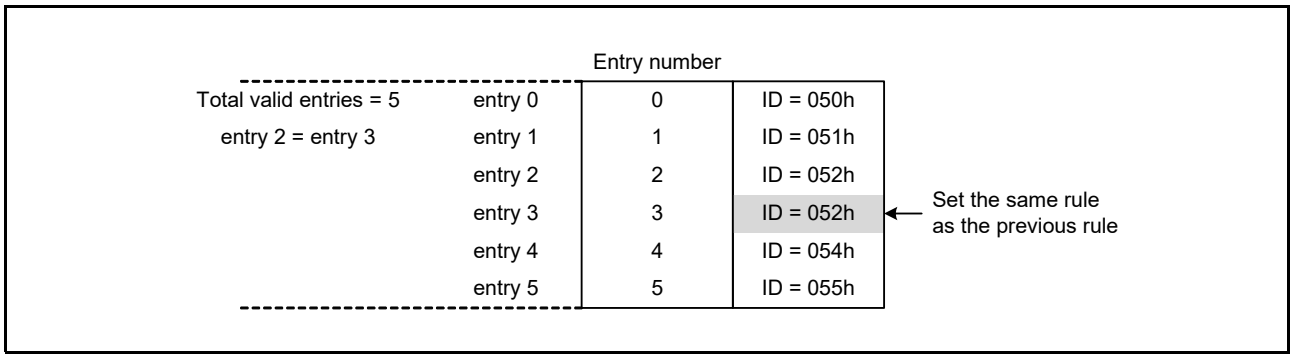


Figure 36.30 Example of Deleting an Entry (After Deleting Entry 3)

(2) Example 2: Adding an Entry (Update Unused Entry)

The following describes how to add a new entry to entry 3 when the total number of entries is 6.

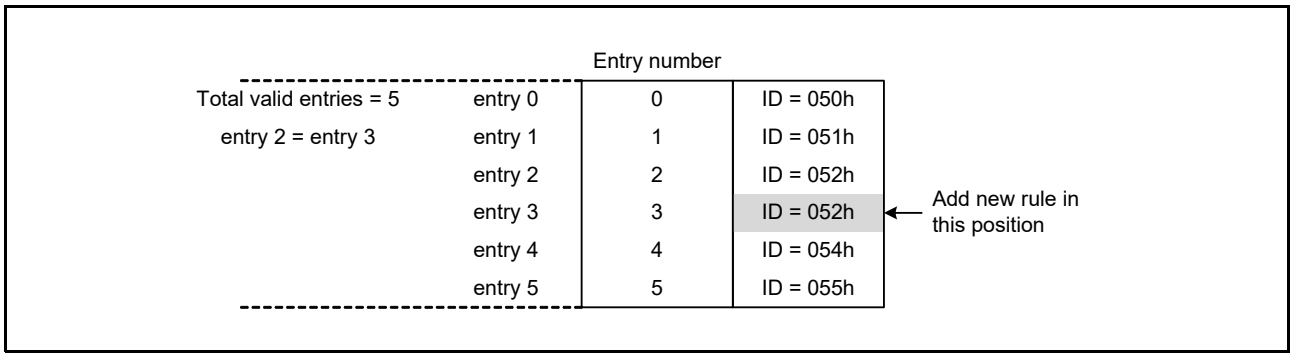


Figure 36.31 Example of Adding an Entry (Before Updating Entry 3)

[How to add an entry]

(1) Set 00000003h to the AFIGSR register.
(2) Set 0000C401h to the AFIGER register.
(3) Set 00000100h to the AFCR register.
(4) Set the new rule by accessing the AFL3.IDR, AFL3.MASK, AFL3.PTR0, AFL3.PTR1 registers.
(5) Set 00000000h to the AFCR register.
(6) Set 0000C400h to the AFIGER register.

The new entry is now added.

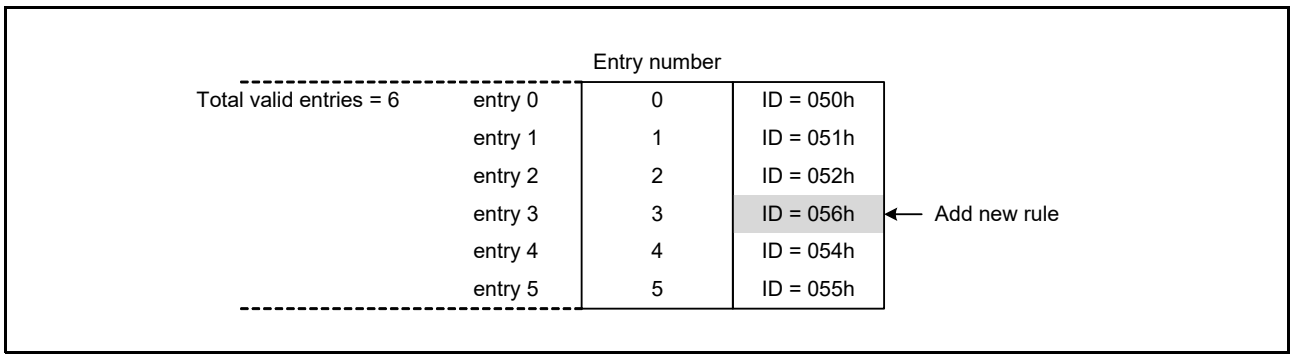


Figure 36.32 Example of Adding an Entry (After Updating Entry 3)

The acceptance filter can use entries in the range of value set in the AFCFG register and entries can be added/deleted within that range. Therefore, the AFCFG register should be set to the maximum number of entries to use.

36.6 FIFO Buffers and Message Buffer Configuration

This section describes the process for configuring the number of receive message buffers, the FIFO buffers, and the transmit message buffers in the CANFD module. The message buffers are mapped as shown in Figure 36.33.
The receive message buffers can be accessed with the RMBn register (n = 0 to 31).
The receive FIFOs can be accessed with the RFBn register (n = 0, 1).
The common FIFO can be accessed with the CFB0 register.
If the common FIFO is configured in transmit mode, only data can be written to the FIFO buffer by the CFB0 register.
If the common FIFO is configured in receive mode, only data can be read by the CFB0 register.
The transmit message buffers can be accessed with the TMBn register (n = 0 to 3).
If unused message buffer is read, it is read as undefined.

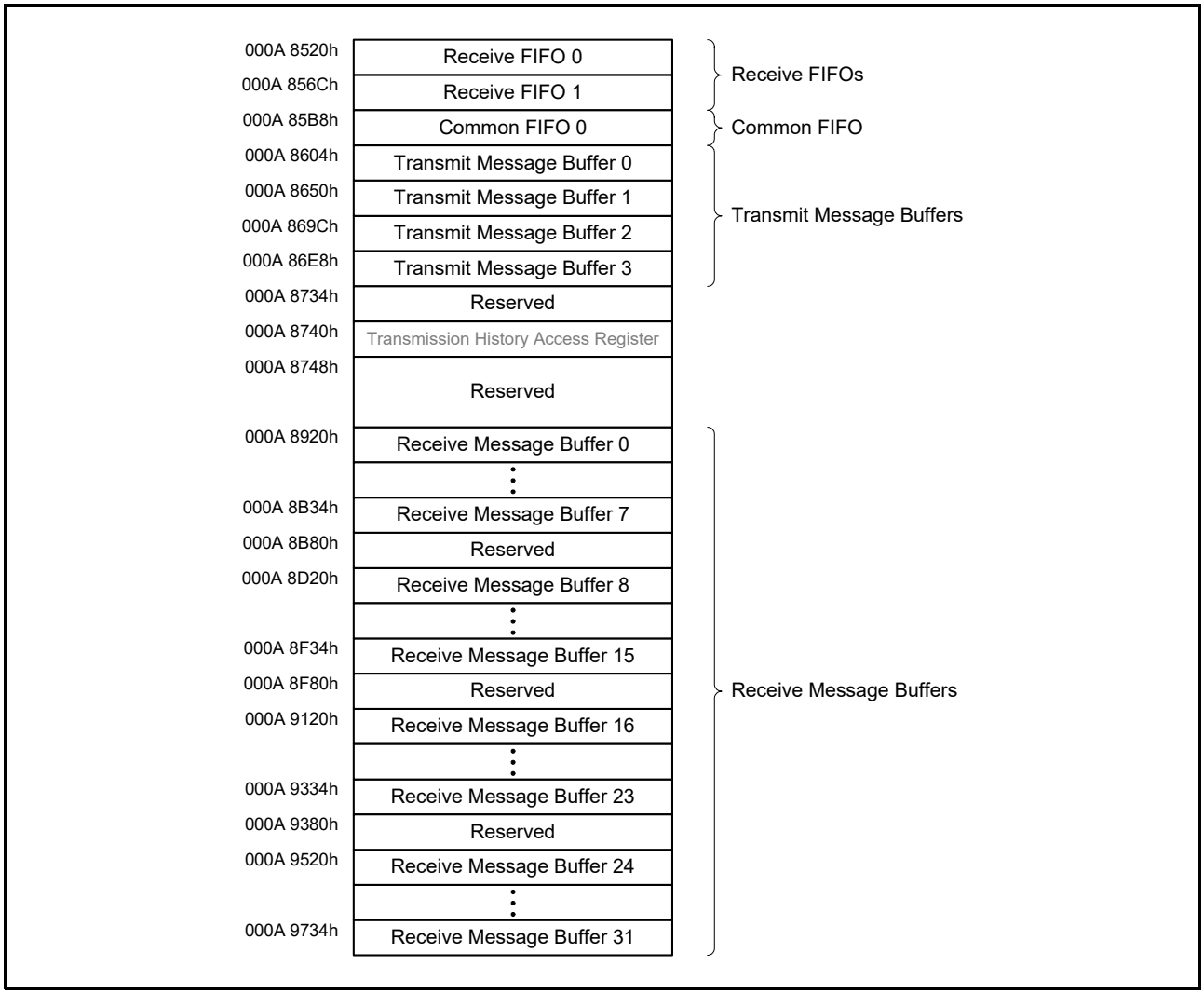


Figure 36.33 Message Buffer Configuration

36.6.1 Receive Message Buffers

In CANFD module, the received frames can be stored in receive message buffers based on the configuration of the AFL entries.
Additionally, the number of receive message buffers required by the system can be selected from 0 to 32.

36.6.1.1 Receive Message Buffer Configuration

The number of receive message buffers in CANFD module can be configured by writing to the RMCR.NMB[5:0] bits. Set the number of message buffers in the range 0 (no receive message buffers) to 32. Do not set a value larger than this.
The AFL entries for routing the received messages to receive message buffers must be configured to match the requirements of the system.
The AFL entries must also be configured properly, and an AFL entry for receive message buffers should not exceed the number of message buffers configured in the RMCR.NMB[5:0] bits.

Note: There is no internal check procedure provided in CANFD module against wrong configuration of the AFL.

The payload size of the receive message buffer can be configured with the RMCR.PLS[2:0] bits. The default size is 8 bytes and the maximum size is 64 bytes.
If the payload size of the received frame exceeds the specified payload size, the message is discarded or the payload is cut according to the setting of the GCFG.OMRC bit.

36.6.2 FIFO Buffers

The CANFD module provides a FIFO buffers to store each receive/transmit frame.
There are two receive-only FIFOs, but common FIFO can be configured to store messages for transmission or reception.
These FIFO buffers can be enabled or disabled, and the following parameters can be configured to match the system requirements:

- FIFO depth
- Interrupt structure
- Message lost mechanism
- Message overwrite mechanism of the FIFO buffers
- Location of the transmit FIFO

If the payload size of the received frame exceeds the specified payload size, the message is discarded or the payload is cut according to the setting of the GCFG.OMRC bit.

36.6.2.1 FIFO Buffers Configuration

In CANFD module, the FIFO buffers can be configured to match the system requirements.
The total number of FIFO buffers is three (two receive FIFOs + one common FIFO).

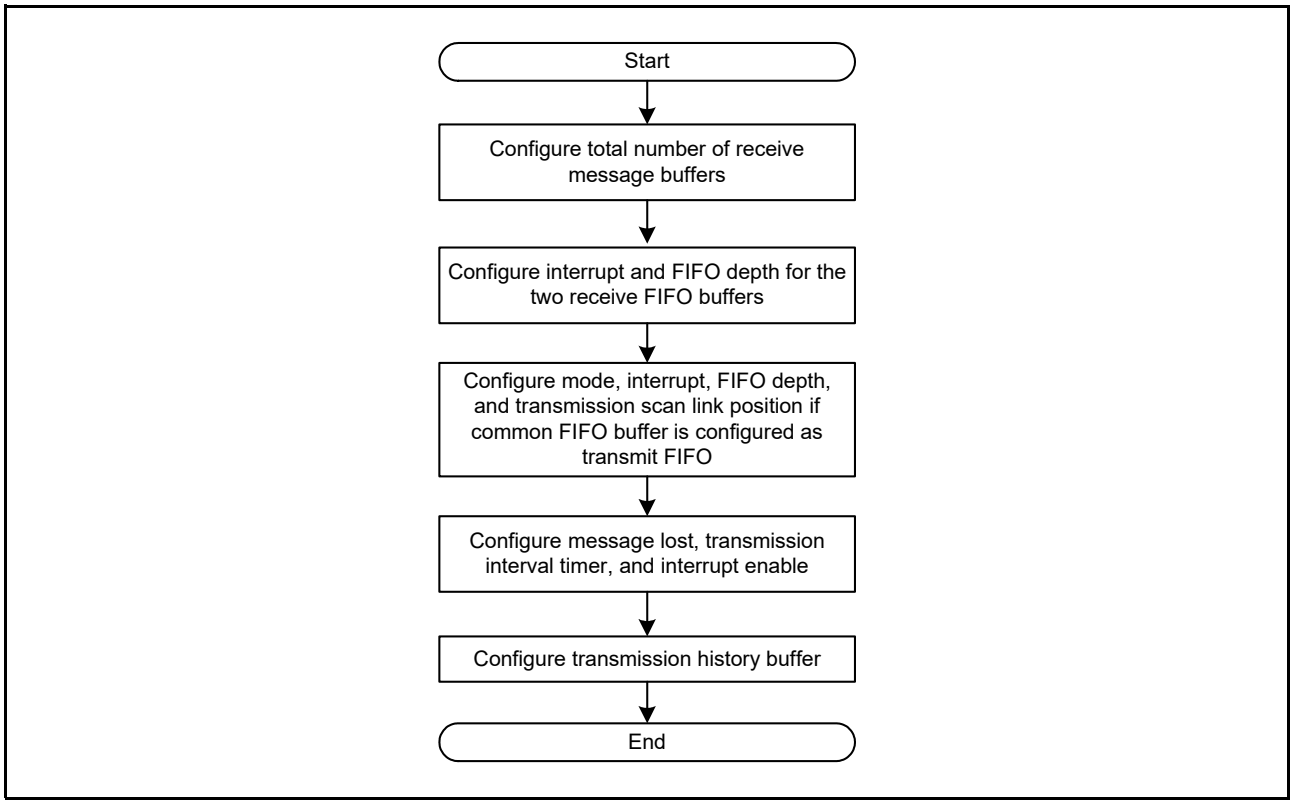


Figure 36.34 FIFO Buffer Configuration Flow in CANFD Module

As shown in Figure 36.34, the various FIFO buffers can be configured by writing to the Receive FIFO n Configuration Register and the Common FIFO 0 Configuration Register.

For the two receive FIFOs, the following parameters can be configured:

- Interrupts
- FIFO depth
- Payload size

For the common FIFO, the following parameters can be configured:

- Mode
- Interrupts
- FIFO depth
- Payload size
- Transmission scan link position

(1) FIFO mode configuration of common FIFO

The mode of the common FIFO can be configured by writing to the CFCR0.MODE bit. The modes that can be configured in the common FIFO are as follows:

- 00b: Receive FIFO mode (default mode after a MCU reset)
- 01b: Transmit FIFO mode.

Messages can only be read from the receive FIFOs and the common FIFO configured in receive FIFO mode. Messages are stored by the CANFD module in these FIFO buffers based on the AFL entries.
Messages can be read and written into the common FIFO configured in transmit FIFO mode.
The pointers can be incremented only when a new message is stored in the FIFO buffer and decremented only when a message is transmitted on the corresponding CAN channel by the CANFD module.
After a MCU reset, all the common FIFO is configured in receive FIFO mode by default. Only enable the FIFO buffers after configuring the common FIFO in the required modes.

(2) FIFO transmit message buffer link configuration

When the common FIFO is configured as transmit FIFO, the FIFO buffer must be linked to a normal transmit message buffer to participate in the transmission scan of a CAN channel.
Do not write data into a transmit message buffer that is linked to a common FIFO. Also, the transmit message buffer linked to a common FIFO should not be a part of the transmit queue.
The link to the transmit message buffer of common FIFO can be configured by writing to the CFCR0.LTM[1:0] bits. The options available for linking the transmit message buffer are:

- 00b: Transmit message buffer 0
- 01b: Transmit message buffer 1
- 10b: Transmit message buffer 2
- 11b: Transmit message buffer 3.

(3) FIFO depth configuration

The depth of each FIFO buffer can be configured by writing to the RFCRn.FDS[2:0] bits and CFCR0.FDS[2:0] bits. The six available options for depth configuration are:

- 000b: 0 messages (FIFO buffer cannot be used)
- 001b: 4 messages
- 010b: 8 messages
- 011b: 16 messages
- 100b: 32 messages
- 101b: 48 messages.

The RAM allocated to the receive message buffers and FIFO buffers is limited to 16 messages (1216 bytes) when the payload size is set to 64 bytes. Do not configure the receive message buffers and FIFO buffers that exceed this maximum limit. CANFD module does not have the function to check the validity of the configuration.

Note: If the FIFO depth of the common FIFO is 4 messages or more (CFCR0.FDS[2:0] > 000b), the link between the common FIFO and the transmit message buffer is valid regardless of whether the FIFO is disabled or enabled. If the FIFO depth is 0 messages, the link between the common FIFO and transmit message buffer is invalid regardless of whether the FIFO is disabled or enabled.

(4) FIFO payload size configuration

The payload size of each FIFO buffer can be configured by writing to the RFCRn.PLS[2:0] bits and CFCR0.PLS[2:0] bits. The eight available options for depth configuration are:

- 000b: 8 bytes
- 001b: 12 bytes
- 010b: 16 bytes
- 011b: 20 bytes
- 100b: 24 bytes
- 101b: 32 bytes
- 110b: 48 bytes
- 111b: 64 bytes.

The RAM allocated to the receive message buffers and FIFO buffers is limited to 16 messages (1216 bytes) when the payload size is set to 64 bytes. Do not configure the receive message buffers and FIFO buffers that exceed this maximum limit. CANFD module does not have the function to check the validity of the configuration.

(5) FIFO interrupt configuration

The interrupt generation conditions for the FIFO buffers can be configured by writing to the RFCRn.RFIM bit and CFCR0.CFIM bit. The two available options are:

- RFIM/CFIM = 0:
  - Receive FIFO mode: An interrupt is generated when the FIFO fill level reaches RFCRn.RFITH[2:0] or CFCR0.CFITH[2:0] value
  - Transmit FIFO mode: An interrupt is generated when the FIFO transmits the last message successfully
- RFIM/CFIM = 1:
  - Receive FIFO mode: An interrupt is generated each time the received message is stored
  - Transmit FIFO mode: An interrupt is generated each time a message is successfully transmitted.

If the RFCRn.RFIM bit for the receive FIFO is 0, an interrupt is generated based on the setting of the RFCRn.RFITH[2:0] bits.
Similarly, if the CFCR0.CFIM bit for a common FIFO set to receive FIFO mode is 0, an interrupt is generated based on the setting of the CFCR0.CFITH[2:0] bits.
The eight options available to set the FIFO fill level for generating interrupts are:

- 000b: Interrupt generated when FIFO is 1/8 Full
- 001b: Interrupt generated when FIFO is 1/4 Full
- 010b: Interrupt generated when FIFO is 3/8 Full
- 011b: Interrupt generated when FIFO is 1/2 Full
- 100b: Interrupt generated when FIFO is 5/8 Full
- 101b: Interrupt generated when FIFO is 3/4 Full
- 110b: Interrupt generated when FIFO is 7/8 Full
- 111b: Interrupt generated when FIFO is Full.

In this case, an interrupt is generated when the message fill level matches the set value.
However, as shown in Table 36.21, there are some restrictions on the RFITH[2:0] and CFITH[2:0] bit settings, depending on the FDS[2:0] bits (FIFO depth setting) of each register.

Table 36.21 FIFO Interrupt Threshold and FIFO Depth Settings

	RFITH[2:0]/CFITH[2:0]							
FDS[2:0]	**111b (full)**	**110b (7/8)**	**101b (3/4)**	**100b (5/8)**	**011b (1/2)**	**010b (3/8)**	**001b (1/4)**	**000b (1/8)**
000b (0 messages)	Invalid (FIFO cannot be enabled)							
001b (4 messages)	Allowed	Prohibited	Allowed	Prohibited	Allowed	Prohibited	Allowed	Prohibited
010b (8 messages)	Allowed							
011b (16 messages)	Allowed							
100b (32 messages)	Allowed							
101b (48 messages)	Allowed							

36.6.2.2 FIFO Buffers Control

To enable the receive FIFO interrupt, set the RFIE bit in the RFCRn register (n = 0, 1) to 1.
To enable the common FIFO interrupt, set either the CFRIE or CFTIE bit in the CFCR0 register:
After configuration is complete, each FIFO can be enabled by setting the RFCRn.RFE bit and CFCR0.CFE bit to 1 to allow transmission and reception of messages.

36.7 Reception and Transmission

36.7.1 Reception

In the CANFD module, messages received on any of the channels, will be stored in receive message buffers or in receive FIFOs or common FIFO configured in receive FIFO mode depending upon the Acceptance Filter List entries:

- up to 32 receive message buffers can be configured
- two receive FIFOs available
- up to one common FIFO can be configured in receive mode

36.7.1.1 Message Storage in Receive Message Buffers

When a message is successfully received and stored in a receive message buffer, the corresponding NDR[n] flag in the RMNDR register is set to 1.
The stored message can be read from the corresponding receive message buffer.
If a new message is stored into a receive message buffer before the previous message in this message buffer is read, the original message is overwritten. There is no mechanism for preventing a new message from overwriting the current message in the receive message buffer. If such a loss of messages is not acceptable, store related messages by using receive FIFO.

Note: Unused data bytes are filled with 00h depending on the DLC value.

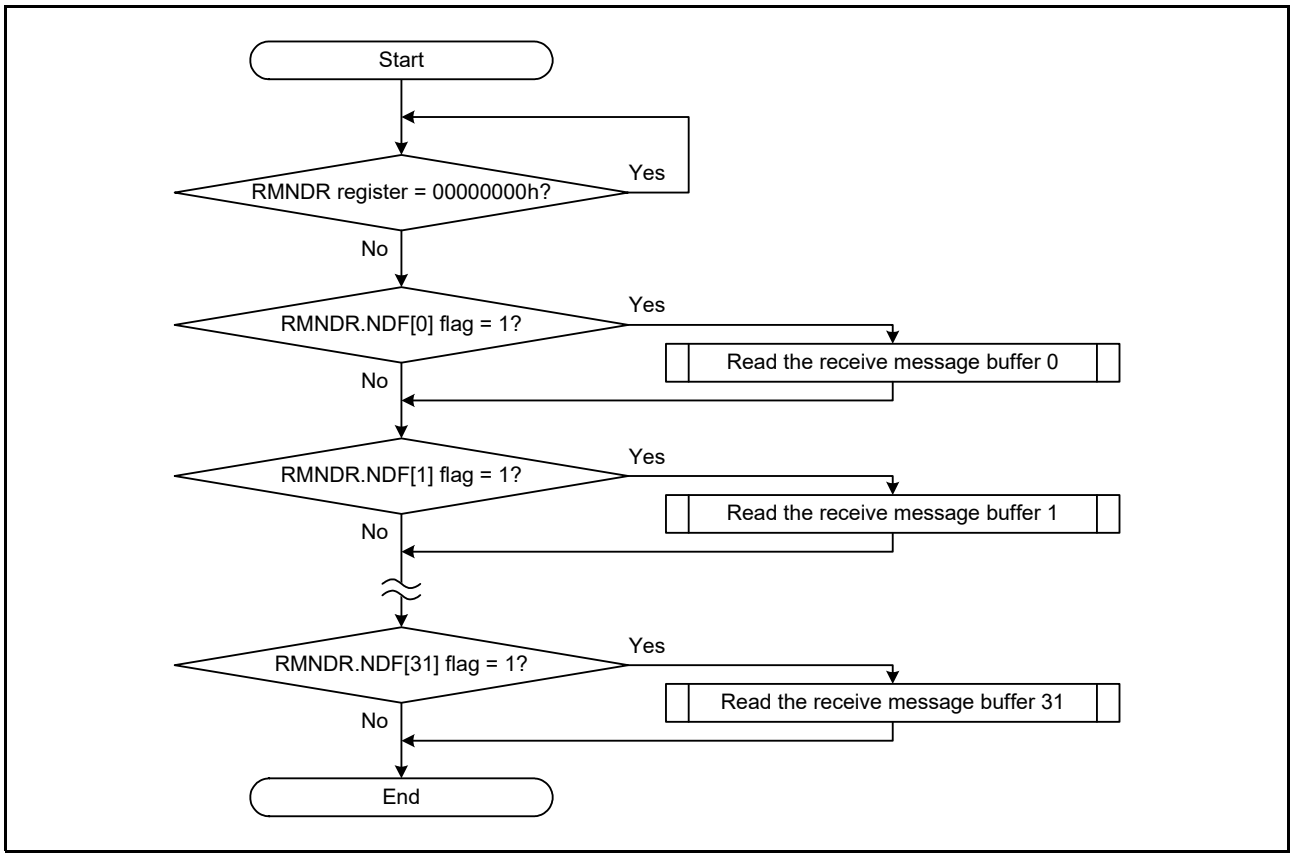


Figure 36.35 Receive Message Buffer Message Access Flow (Example for Polling Case)

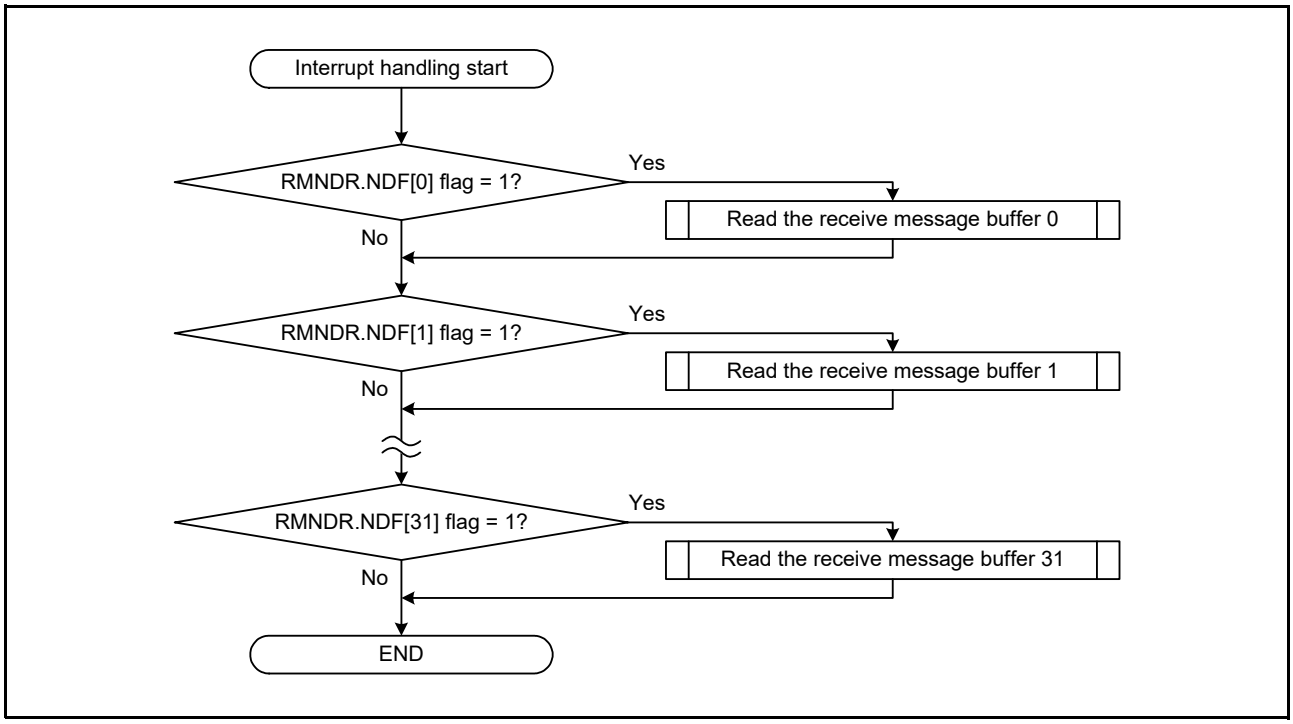


Figure 36.36 Receive Message Buffer Message Access Flow (Example for Interrupt Case)

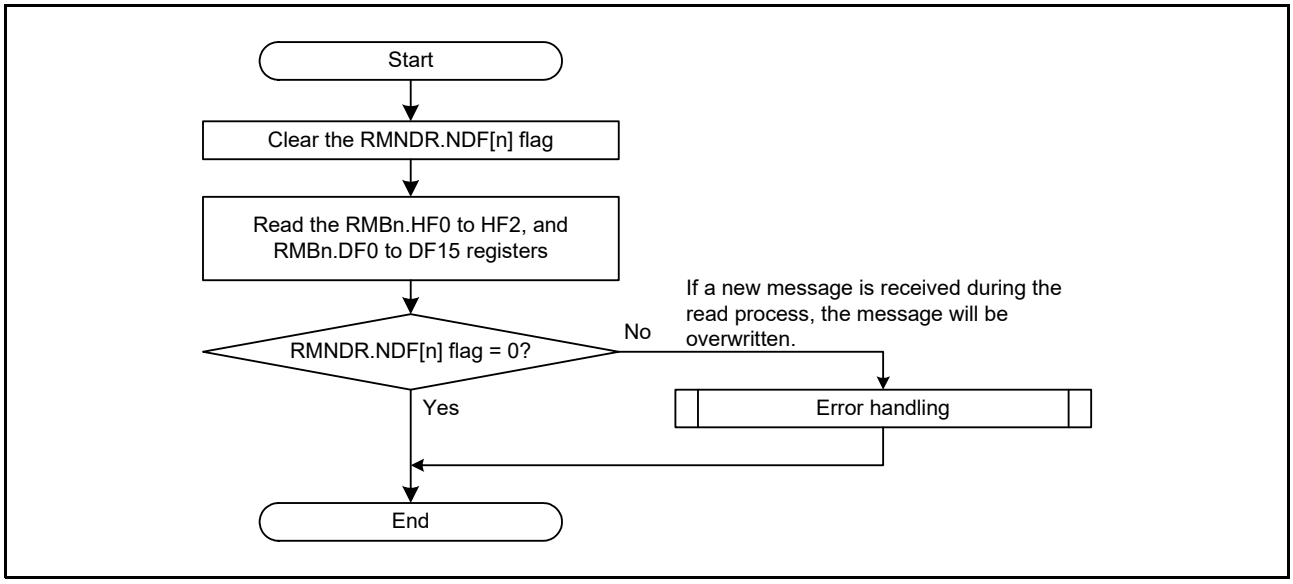


Figure 36.37 Receive Message Buffer n Read Flow

36.7.1.2 Message Storage in FIFO Buffers

The AFL entries for routing the received messages to receive FIFOs or common FIFO configured in receive mode should be configured based on the requirements of the system.
The AFLn.PTR1.RF0E, RF1E, or CF0E bit in the matching AFL entry selects the FIFO buffers to which the related reception message will be stored.
When the received message is stored in one or more receive FIFOs or common FIFO configured in receive FIFO mode, the message counter value is incremented in the corresponding Receive FIFO n Status Register or Common FIFO 0 Status Register.
Depending upon the configuration of the FIFO buffers, an interrupt may also be generated. The message can be read from the corresponding FIFO access registers.

Note: Since many messages can be stored in the FIFO buffers, reading more than one message may be required to read the latest message stored in a FIFO buffer.

If the message count value matches the FIFO depth, then the FIFO full flag is set.
When the value 000000FFh is written to the corresponding FIFO Pointer Control Register, then the message count is decremented by 1.

Only write 000000FFh to the FIFO Pointer Control Register after completely reading the message from the FIFO Access Register of the corresponding FIFO.
When all the messages stored in the FIFO are read, then the FIFO empty flag is set.
If a new message is stored into the FIFO when the FIFO message count matches the FIFO depth (FIFO full condition), the FIFO message lost flag is set and the new message will be lost (messages already stored will not be overwritten).
To prevent message loss due to overrun, set an appropriate interrupt generation threshold and generate an interrupt before the FIFO becomes full.

The receive FIFOs and the common FIFO configured in receive FIFO mode can be disabled at any time by clearing the RFCRn.RFE bit or CFCR0.CFE bit.
When the RFCRn.RFE bit or CFCR0.CFE bit is cleared, the message read and write pointers of the FIFO are cleared and are no longer active. Hence, all messages in the FIFO buffers will be lost and no further messages can be stored into the FIFO.
When the receive FIFOs or common FIFO configured in receive FIFO mode is set to be read by DTC/DMA transfer, do not read the FIFO buffer on the CPU or write 000000FFh to the FIFO Pointer Control Register (RFPCR0, RFPCR1, or CFPCR0). The FIFO read pointer is automatically updated when read by DTC / DMA transfer.

Note: If the interrupt flag is set for a FIFO buffer and then the FIFO is disabled, then the interrupt flag will not be cleared automatically. The interrupt flag should be cleared before disabling the FIFO.
Note: When the next frame is received before clearing the receive interrupt flag, the receive interrupt flag is not set again.

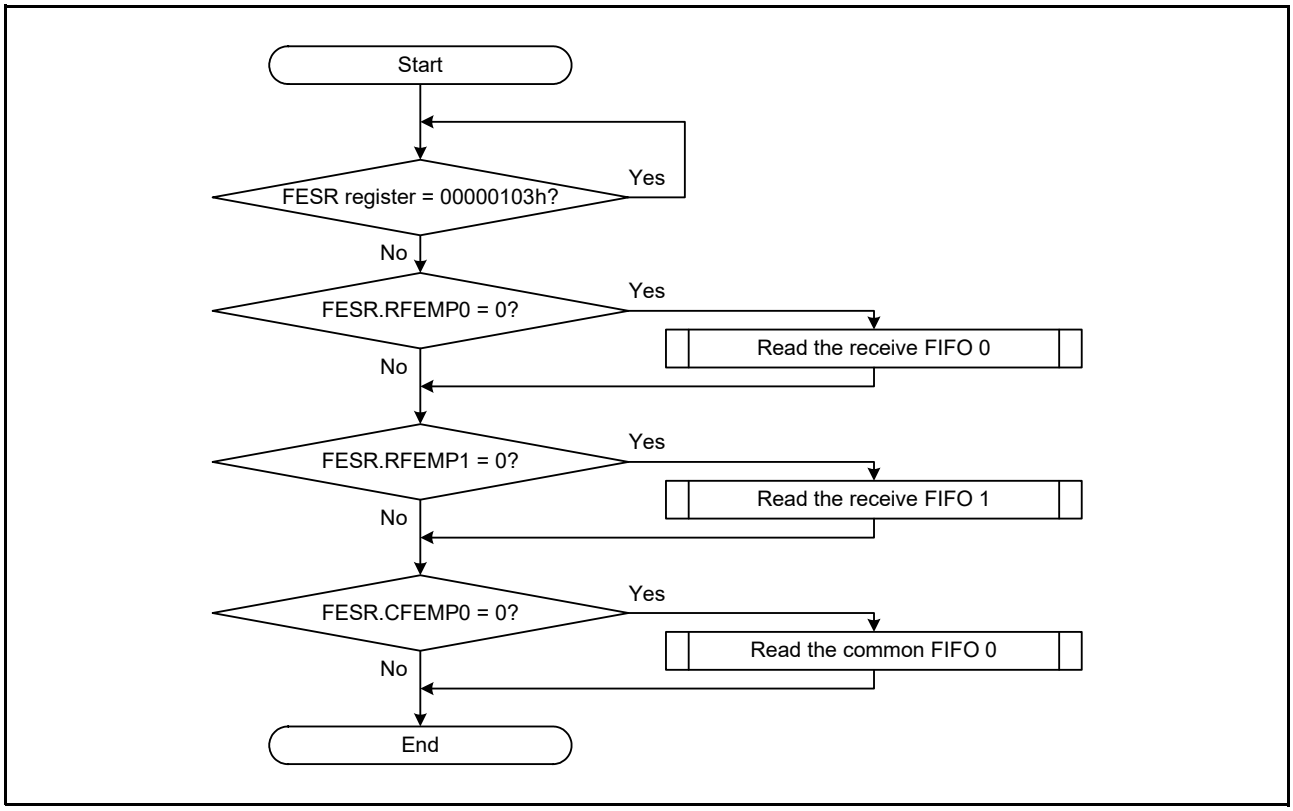


Figure 36.38 FIFO Buffer Message Access Flow (Example for Polling Case)

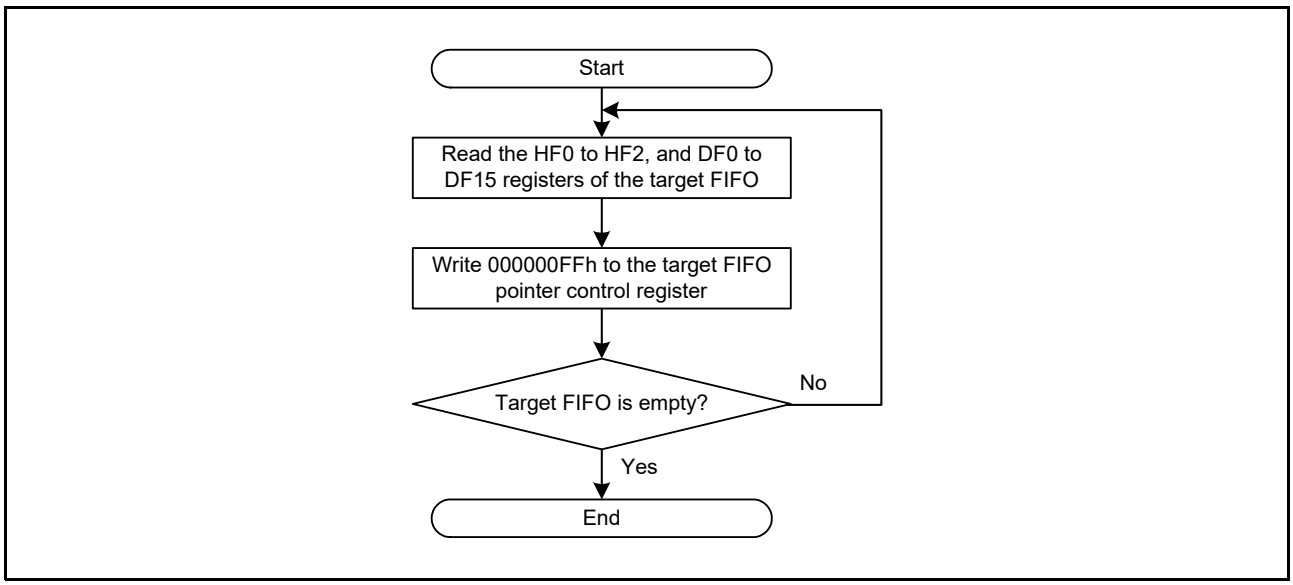


Figure 36.39 Receive FIFO Read Flow (Example for Polling Case)

If the interrupt flag is cleared after the FIFO read process is completed, the interrupt flag is not set even if the next frame has already been received. The FIFO read process must be performed and the interrupt flag must be cleared before the reception of the next frame is completed. If the process is not in time, make sure the FIFO is empty, clear the interrupt flag, and make sure the FIFO is empty again.

36.7.1.3 Timestamp

The timestamp counter is a free-running counter that can be used to check reception time of a received message or transmission time of a successful transmitted message. The timestamp counter value will be captured based on the GFDCFG.TSCPS[1:0] bit setting (sample point of SOF, EOF when the frame is taken valid, or sample point of res bit following the FDF bit in case of a CAN FD frame). For reception the timestamp counter value is stored together with the message ID and data into the target receive message buffer or receive FIFO.
For transmit message the timestamp counter value is stored as part of the transmission history entry.
The counter can be clocked from PCLKB or the bit timing clock of the CAN channel. The counter count source can be set with the GCFG.TSCS bit. If the GCFG.TSCS bit is 0, PCLKB is used. If it is 1, the bit time clock of CAN channel is used.
The count source of the timestamp counter can be divided by the coefficient defined by the GCFG.TSP bit (timestamp prescaler).
The timestamp counter can be reset to 0000h with the GCR.TSCR bit (timestamp counter reset).

36.7.2 Transmission

There are several possible transmission configurations:

- Normal transmission
- FIFO transmission
- Transmit queue transmission

The CANFD module has four transmit message buffers. These message buffers are only used for transmission and cannot be configured for reception.
Additionally transmission from transmit queue and/or common FIFO configured in transmit FIFO mode can be set in the following way (refer to Figure 36.33):

- Transmit queue
  3 or 4 transmit message buffers for one channel can be grouped to form a transmit queue with a common access window.
  Transmit message buffer 0 acts as an access window for transmit queue 0 (TXQ0).
- Common FIFO (transmit FIFO mode)
  The CANFD module has one common FIFO. Common FIFO configured in transmit FIFO mode is linked to any of the transmit message buffers 0 to 3.
  The linked transmit message buffer replaces the common FIFO. Do not access the TMCRn or TMSRn registers of the linked send message buffer.

Note: Common FIFO should not be linked to transmit message buffers that are already part of a transmit queue.

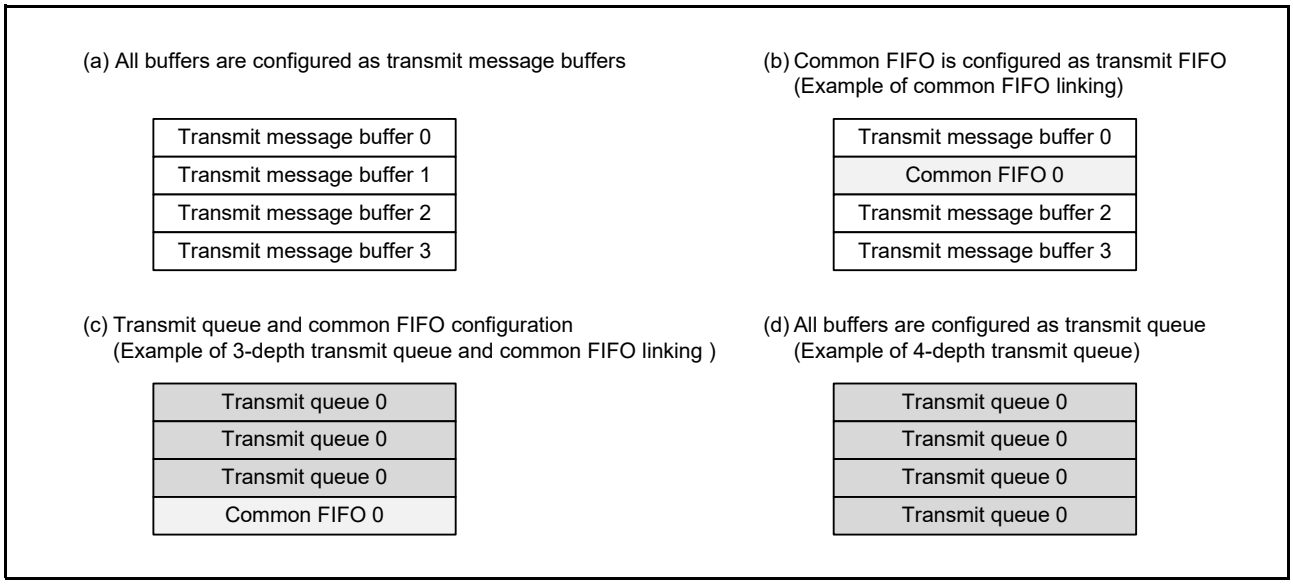


Figure 36.40 Channel Transmit Message Buffer Configuration

36.7.2.1 Transmission Priority

If two or more transmit message buffers of a channel are configured for transmission, the transmission priority in the CANFD module can be selected from the followings:

- CAN ID priority
- Message buffer number priority

The transmission priority is common for all message buffers. It can be configured via the GCFG.TPRI bit.
For message buffer number priority transmission, the smallest message buffer number with transmission request has the highest priority for transmission. This also includes the transmit message buffers linked to the common FIFO configured in transmit FIFO mode.
Do not select message buffer number priority when using the transmit queue.
For CAN ID priority transmission, ID priority complies with the CAN bus arbitration rule (as specified in ISO 11898-1 specification). All transmit message buffers can enter the ID priority comparison for message buffers configured for transmission. This also includes the transmit message buffers linked to the common FIFO configured in transmit FIFO mode and includes the transmit queue message buffers.
If the ID of two or more message buffers is the same, then the smaller message buffer number will have higher priority for transmission.

Note: For common FIFO configured in transmit FIFO mode, only the message currently being pointed to by the FIFO read pointer can be included in the transmission arbitration.
If the message is being transmitted from the FIFO, then the next pending message within the same FIFO will be considered in the transmission arbitration.
In contrast to this, all transmit message buffers of a transmit queue will participate in internal transmission arbitration.

Figure 36.41 below shows the transmission configuration flow.

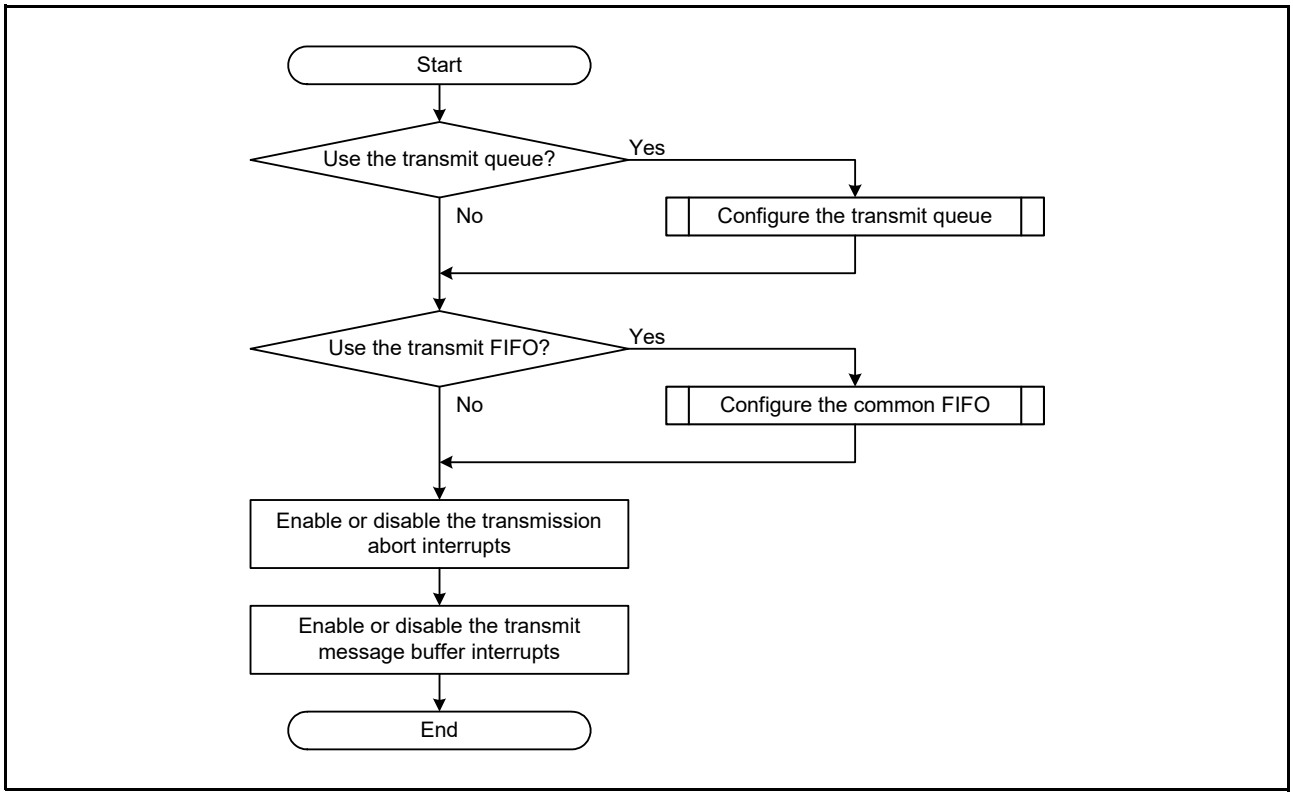


Figure 36.41 Transmission Configuration Flow

36.7.2.2 Message Transmission from Transmit Message Buffer

Each transmit message buffer has two modes of message transmission:

- Normal Transmission Mode
  When the message buffer is set to normal transmission mode, the data frames or remote frames set in that message buffer can be transmitted.
  Completion of normal transmission can be checked by the TMSRn.TXRF[1:0] flags. These flags are set to 10b or 11b when a normal transmission is successful.
  When arbitration is lost or an error occurs during transmission, the message transmission will be retried if no transmission abort request is set in the transmit message buffer.
  A new internal transmission arbitration is performed for all message buffers that have a transmission request.
- One-Shot Transmission Mode
  When the TMCRn.ONESHOT bit is set to 1, the message buffer is placed in one-shot transmission mode and attempts to transmit the message only once.
  Completion of one-shot transmission can be checked by the TMSRn.TXRF[1:0] flags. These flags are set to 10b or 11b when the one-shot transmission is successful.
  The TXRF[1:0] flags are set to 01b when arbitration is lost or an error occurs during transmission. In this case, message transmission will not be retried.

Figure 36.42 shows the transmission request procedure from transmit message buffer.

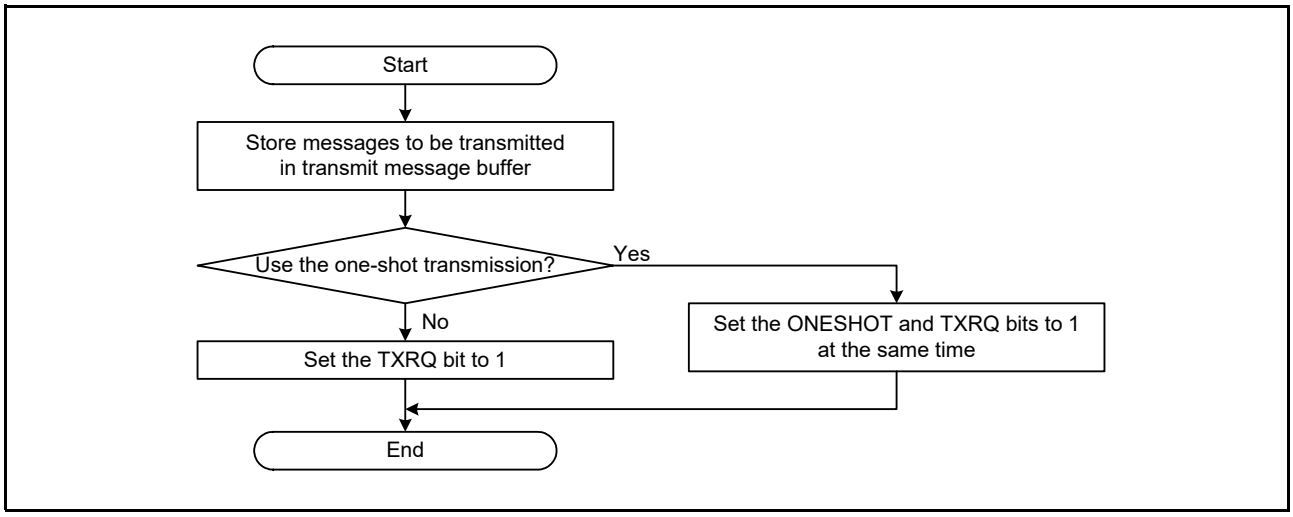


Figure 36.42 Transmission Request Procedure from Transmit Message Buffer

Table 36.22 shows configuration of the TMCRn register.

Table 36.22 Configuration of TMCRn Register

Transmission Request TXRQ Bit	Transmission Abort Request TARQ Bit	One-Shot Transmission Enable ONESHOT Bit	Message Buffer Status
0	0	0	Normal transmission is stopped.
0	0	1	One-shot transmission is stopped.
1	0	0	Data frames or remote frames are transmitted in normal transmission mode.
1	0	1	A data frame or a remote frame is transmitted in one-shot transmission mode.
1	1	0	Transmission abort is requested.
1	1	1	One-shot transmission abort is requested.

Figure 36.43 shows timings for successful transmission from two message buffers.

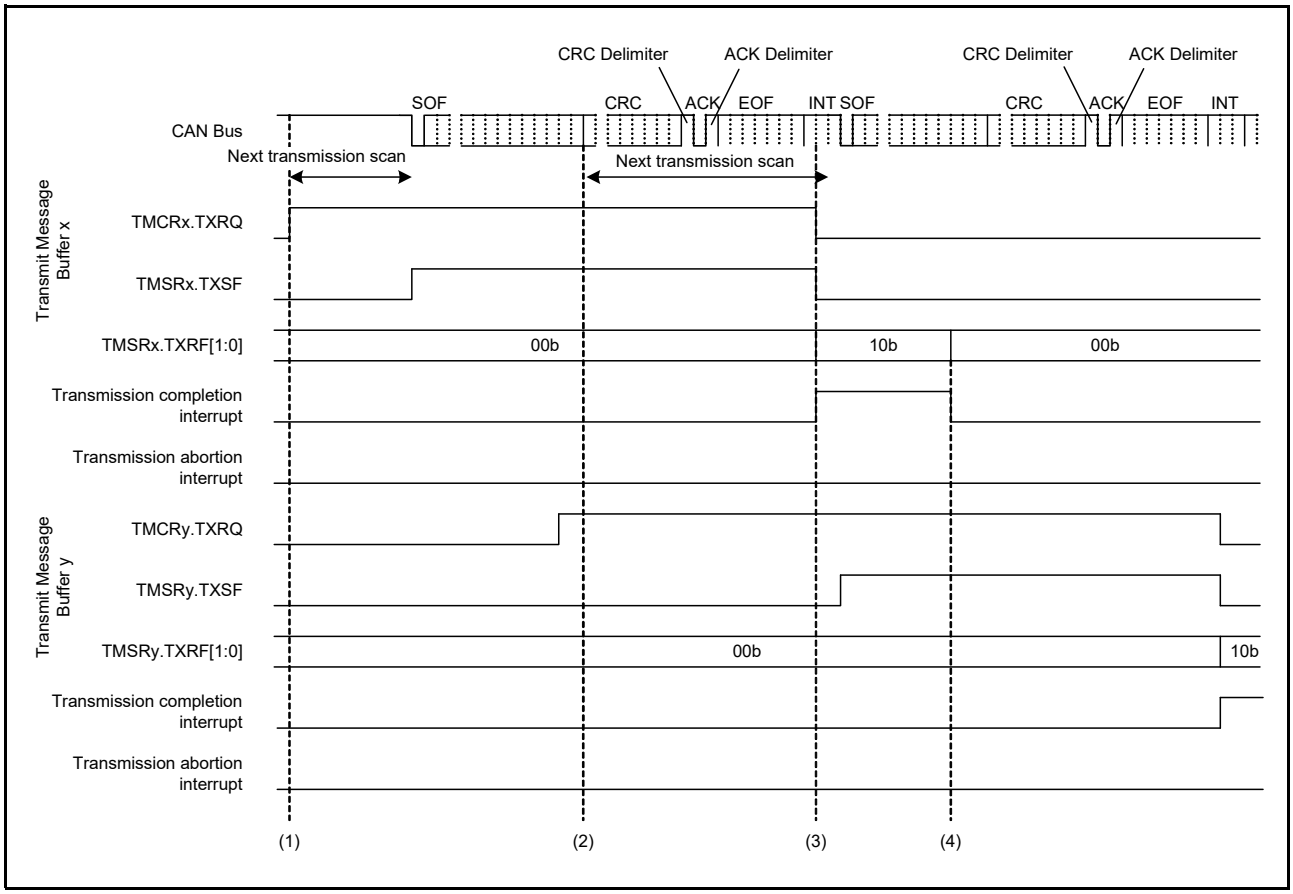


Figure 36.43 Timing of Request and Flag Bits for Successful Transmission

(1) If the TMCRx.TXRQ bit is set to 1 in the bus idle state, message buffer scanning procedure starts to decide the highest priority message buffer for transmission.
When the transmit message buffer is decided, the TMSRx.TXSF bit is set to 1 (transmitting), and CAN channel starts the transmission*1.

(2) At the first bit of CRC, the transmission scanning procedure starts for the next possible transmission when pending transmission requests exist.

(3) If the message has been successfully transmitted, the TMSRx.TXRF[1:0] flags are set to 10b and the TMSRx.TXSF flag and the TMCRx.TXRQ bit are cleared.
When the TMIER0.TMIEx bit is set to 1 (transmit message buffer interrupt enabled), the successful transmission interrupt request is generated. To clear the related interrupt line the TMSRx.TXRF[1:0] flags have to be cleared.

(4) Before starting the next transmission, clear the TMSRx.TXRF[1:0] flags. Load the next message in the transmit message buffer and set the TMCRx.TXRQ bit to 1 again.
The TMCRx.TXRQ bit cannot be set to 1 again before TMSRx.TXRF[1:0] flags are cleared.

Note: The setting point of the TMSRx.TXSF flag is not always fixed at the start of the SOF. It may be delayed up to the start of the standard ID.

Note 1. If arbitration is lost after the CAN channel starts the transmission, the TMSRx.TXSF flag is cleared. Then the transmission scanning procedure is performed again to search for the highest priority transmit message buffer from the beginning of the first bit of CRC.
If an error occurs either during the transmission or following the loss of arbitration, the transmission scanning procedure is performed again during error frame to search for the highest priority transmit message buffer.

The Figure 36.44 shows timings for transmission abort for two message buffers.

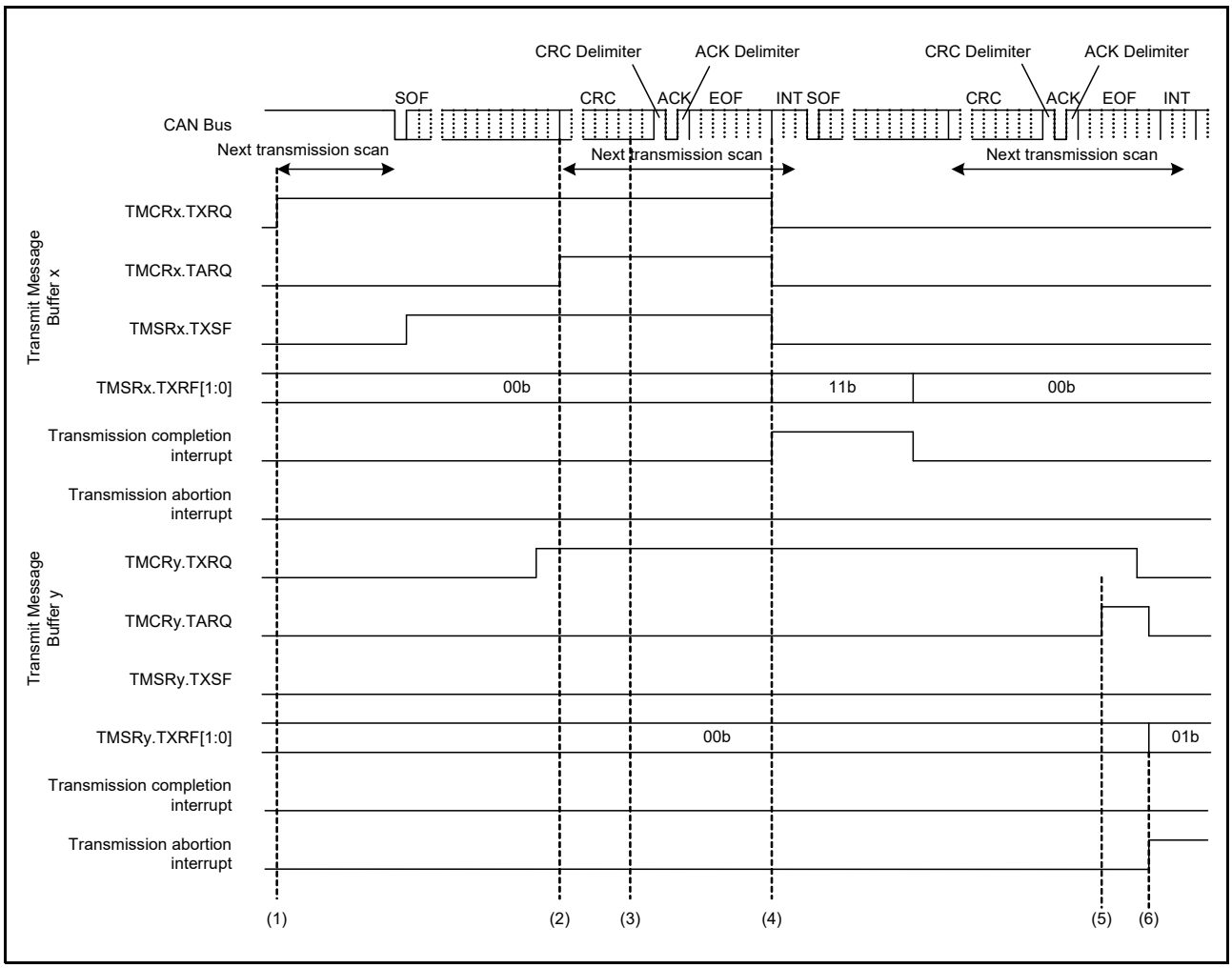


Figure 36.44 Timing of Request and Flag Bits for Transmission Abort

(1) If the TMCRx.TXRQ bit is set in the bus idle state, message buffer scanning procedure starts to decide the highest priority message buffer for transmission.
When the transmit message buffer is decided, the TMSRx.TXSF flag is set (transmitting), and CAN channel starts the transmission*1.

(2) If the TMCRx.TARQ bit is set to 1 when the related message buffer is already selected for transmission or currently transmitting, the message will not be aborted, if no error occurs or arbitration is lost.

(3) At the first bit of CRC, the transmission scanning procedure starts for the next transmission. In this example timing chart message buffer y is not selected as next transmit message buffer.

(4) If the message has been successfully transmitted, the TMSRx.TXRF[1:0] flags are set to 11b and the TMSRx.TXSF flag and TMCRx.TXRQ bit are cleared.
When the TMIER0.TMIEx bit is set to 1 (transmit message buffer interrupt enabled), the CAN successful transmission interrupt request is generated.
To clear the related interrupt line the TMSRx.TXRF[1:0] flags has to be cleared.

(5) Another CAN node is transmitting on the CAN bus (the TMSRy.TXSF flag is not set): if the TMCRy.TARQ bit is set to 1 when the related channel is under transmission scan, the transmission request cannot be cleared.

(6) After internal processing time the transmission is aborted and the TMSRy.TXRF[1:0] flags are set to 01b.
If the message buffer is not transmitting or selected as next transmit message buffer or under transmit scan, the abort is immediately accepted and the corresponding TMSRy.TXRF[1:0] flags are set to 01b.
In addition, the TMCRy.TXRQ and TMCRy.TARQ bits are cleared automatically.

When the CHCR.TAIE bit is set to 1 (transmission abort interrupt enabled), an interrupt is generated for successful transmission abort.
To clear the related interrupt the TMSRy.TXRF[1:0] flags have to be cleared.

Note 1. If arbitration is lost after the CAN channel starts the transmission, the TMSRx.TXSF flag is cleared. Then the transmission scanning procedure is performed again to search for the highest priority transmit message buffer from the beginning of the first bit of CRC.
If an error occurs, either during the transmission, or following the loss of arbitration, the transmission scanning procedure is performed again during error frame to search for the highest priority transmit message buffer.

36.7.2.3 Message Transmission from FIFO Buffer

The CANFD module has one common FIFO. The common FIFO can be linked to the transmit message buffer by the CFCR0.LTM[1:0] bits if configured in transmit FIFO mode.
When a transmission scan is started and the common FIFO corresponding to that transmit message buffer is enabled, the relevant message in the common FIFO will participate in the transmission scan.
Do not configure the transmit message buffer linked to the common FIFO configured in transmit FIFO mode.

(1) Transmit FIFO Operation

Messages can be written into the transmit FIFO by writing to the common FIFO buffer 0 (CFB0).
When the value 000000FFh is written into the CFPCR0 register, the message count of the FIFO is incremented by 1. Before writing to the CFPCR0 register, wait until the message has been completely written to the CFB0. If the message count matches the FIFO depth, the CFSR0.FULL flag is set to 1.
The oldest message in the transmit FIFO is included in the scan for transmission.
When a message is successfully transmitted from the transmit FIFO, the message count value is decremented by 1. When all the messages from the FIFO are transmitted, the CFSR0.EMPTY flag is set to 1.
The interrupt generation conditions for the transmit FIFO buffers can be set by the CFCR0.CFIM bit.
If the CFCR0.CFIM bit is set to 0, interrupt is generated when last message is successfully transmitted from the transmit FIFO buffer. If the CFCR0.CFIM bit is set to 1, interrupt is generated for every successfully transmitted message from the transmit FIFO buffer.
Common FIFO can set interrupt, when CAN frame transmission is completed.
The common FIFO configured in transmit FIFO mode can be disabled by setting the CFCR0.CFE bit to 0. If this bit is set to 0, the CFSR0.EMPTY flag is set to 1 as described below:

- immediately if the message from the transmit FIFO is neither scheduled for the next transmission nor in transmission
- following the transmission completion, the detection of an error on the CAN bus, arbitration lost or transition to CH_HALT or GL_HALT mode if the transmission from the transmit FIFO is already scheduled for transmission or already in transmission.

Note: The common FIFO is considered as disabled after setting the CFCR0.CFE bit to 0 only when the CFSR0.EMPTY flag is set to 1 for the corresponding common FIFO.

If there are other pending messages in the transmit FIFO buffer, they will be lost and the transmission needs to be requested again. Before the CFCR0.CFE bit is set to 1 again ensure that the CFSR0.EMPTY flag is set to 1 and that there is no pending abort request from the transmit FIFO.
When the CFCR0.CFE bit is set to 0, the message read and write pointers of the FIFO are cleared and are no longer active. Hence, all messages in the FIFO buffers will be lost and no further message can be stored into the FIFO.

The FIFO transmission request procedure after a configuration is shown in Figure 36.45.

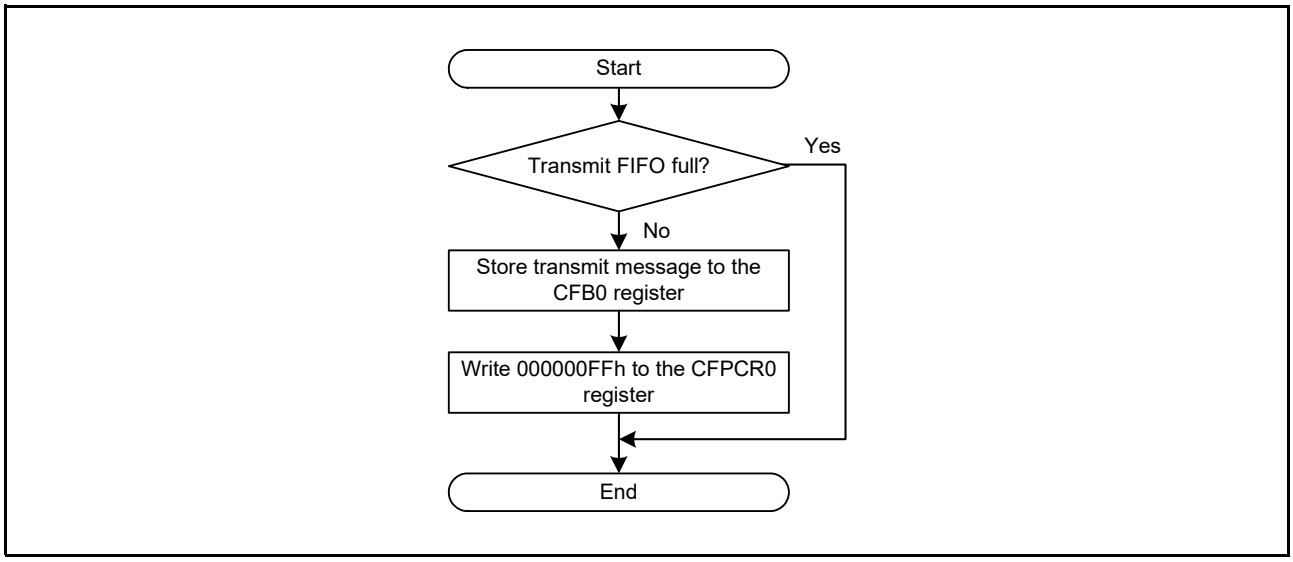


Figure 36.45 Transmit FIFO Transmission Request Procedure

(2) Interval Timer for FIFO Transmission

For the common FIFO in transmit mode it is possible to specify a delay between two consecutive messages that are configured for transmission from the same FIFO buffer. This delay is called interval time. This interval time starts after the first message has been successfully transmitted from the FIFO buffer after the CFCR0.CFE bit is set to 1.
When the common FIFO in transmit mode is enabled, then the first message will be transmitted without considering this interval time.
The interval timer will stop counting when:

- FIFO is disabled by set the CFCR0.CFE bit to 0
- CAN channel is in CH_RESET mode.

The interval time is specified by the value of the CFCR0.TINT[7:0] bits and can be specified from 0 to 255 timer units. The timer unit can be defined based on two different count sources for the interval timer. Select the value 0 to disable the interval timer for FIFO transmission.
The count source can be selected by the CFCR0.ITCS bit. For the count source the CAN bit timing clock of the related channel or a reference clock could be selected.
If CAN channel bit time clock is configured as count source and the CAN channel enters CH_HALT, CH_RESET, or CH_SLEEP mode, then the interval timer is stopped for that channel.
If peripheral clock is selected as interval timer clock source, then the interval timer is stopped only when the CAN channel is in CH_RESET or CH_SLEEP mode.
The reference clock can be used to configure the interval time in fixed time units. It is based on the PCLKB. The GCFG.ITP[15:0] bits define the relation between the PCLKB frequency/period and the reference clock period.
Refer to Table 36.23 for configuration values of the GCFG.ITP[15:0] bits to achieve different reference clock periods based on the PCLKB frequency/period.

Table 36.23 Configuration Example for Interval Timer Prescaler

	Reference clock period		
PCLKB frequency (period)	**1 µs**	**100 µs**	**500 µs**
16 MHz (62.5 ns)	16	1600	8000
20 MHz (50 ns)	20	2000	10000
32 MHz (31.25 ns)	32	3200	16000
50 MHz (20 ns)	50	5000	25000

Additionally the reference clock resolution can be specified by the CFCR0.ITR bit.
The interval time is based on the reference clock period multiplied by the configured value (×1 or ×10).
The reference clock based interval timer can be used to follow the requirements of the ISO 15765-2 Separation Time.
The whole range for the separation time from 100 µs to 127 ms can be covered.
The specified interval time starts after successful transmission event (after EOF7 state of the CAN protocol).
When the interval time has elapsed, the next transmission request is raised by the related transmit FIFO. Hence, the interval time defines the minimum time between two messages transmitted from one FIFO.
The next message will earliest be sent after this interval time. The Figure 36.46 shows an example timing of the internal processing.

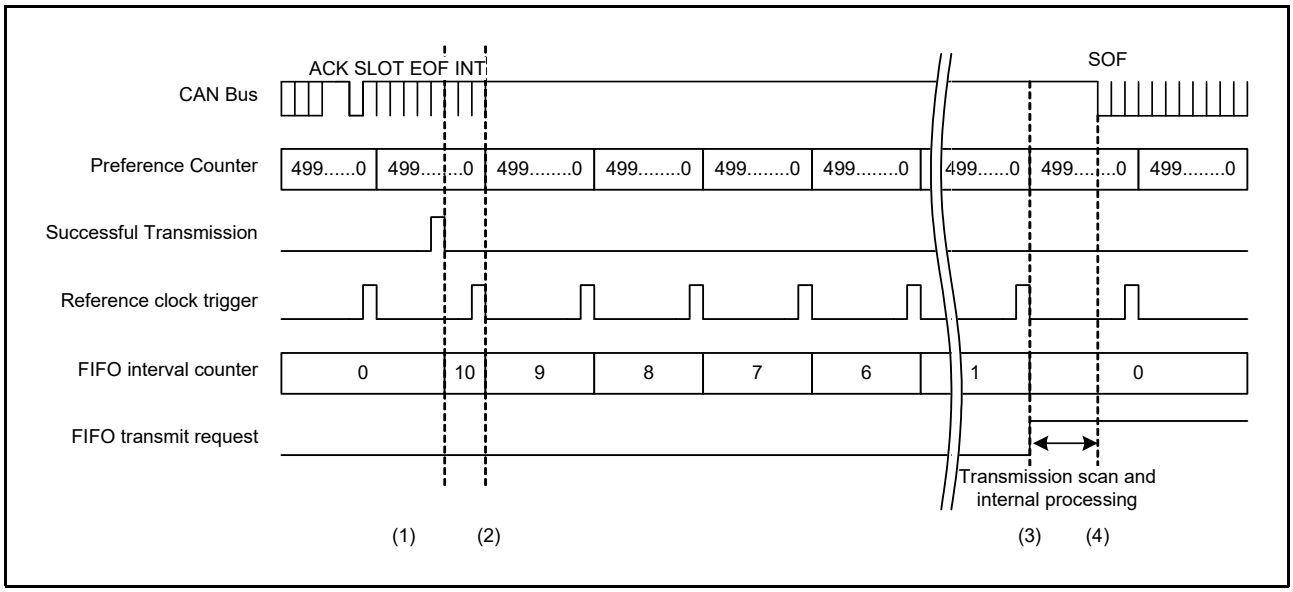


Figure 36.46 Example for Interval Processing Time

The configuration for this timing above is following:

- PCLKB frequency = 50 MHz
- Interval timer prescaler (GCFG.ITP[15:0]) = Divided by 500
- Reference clock due to the settings above = 10 µs
- Common FIFO interval timer count source selection (CFCR0.ITCS) = 0
- Common FIFO interval timer resolution (CFCR0.ITR) = 0
- Common FIFO transmission interval (CFCR0.TINT) = 10 counts
- Theoretical message separation interval = 100 µs

(1) Internal FIFO interval timer is restarted with the occurrence of successful transmission result. This restart is not synchronized to the reference clock trigger. Therefore the first interval is counting less or equal to one reference clock interval.
(2) With the next reference clock trigger the FIFO interval timer is decremented.
(3) When the FIFO interval timer reached the value 0, the FIFO transmit request is set.
(4) When the FIFO is selected for transmission then the transmission will start soon. Due to internal processing this usually takes less than 3 bit time, between internal FIFO transmit request set 3. (shown above) and actual transmission.
When multi events like reception scan, internal message routing, transmit scan happen, then it could take up to 126 PCLKB cycles.

As shown in Figure 36.46, it is not guaranteed that the minimum interval is always equal to the configured value. If a minimum time must never be breached, configure the CFCR0.TINT bit to required minimum value + 1.
If transmit message buffers or transmit FIFOs are configured for transmission for the channel the real delay between two messages transmitted from a transmit FIFO can be much longer than specified by the interval time due to higher priority message transmission from these transmit message buffers or transmit FIFOs.
Figure 36.47 shows a block diagram of the FIFO interval timer.

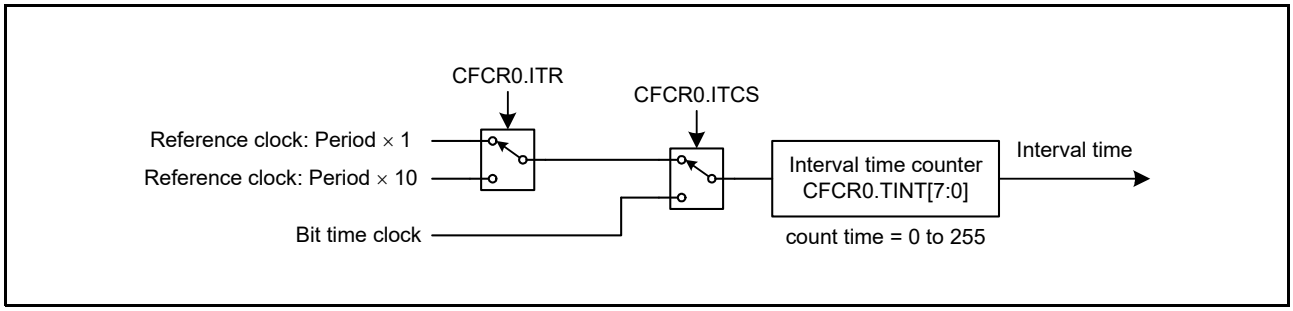


Figure 36.47 Block Diagram of FIFO Interval Timer

36.7.2.4 Transmit Queue

Each enabled transmit queue consists of three or four transmit message buffers, which are accessed via one access window.
The transmit queue 0 (hereafter TXQ0) can be configured with a depth of three to four, and it is using the transmit message buffer 0 as access window. All the message of TXQ0 enter the priority comparison for the transmission, which should be ID priority (GCFG.TPRI = 0).
The registers for TXQ0 are TQCR0, TQSR0, and TQPCR0.
As access window transmit message buffer 0 (TXQ0) is used, refer to related access registers TMBn.HF0, TMBn.HF1, TMBn.HF2, and TMBn.DF0 to TMBn.DF15.
The depth of each TXQ0 buffer can be configured by writing to the TQCR0.QDS[1:0] bits. TXQ0 can be set from TMB0 to TMB3 as a queue buffer at the maximum.
The available options for depth configuration are:

- 10b: 3 messages
- 11b: 4 messages.

Do not access all transmit message buffers except transmit message buffer 0 that configure the transmit queue. Also, do not access the TMCRn registers that correspond to the transmit message buffers that configure the transmit queue.
When writing data to TXQ0, check the status of TXQ0 before writing the data to be transmitted.
The messages stored to the transmit queue access window are internally stored to a free buffer of the transmit queue.
When the TXQ0 buffer is full, do not access the queue anymore. If the transmit data is written when the TXQ0 is full, the

transmit data will be overwritten.
The transmit queue can be disabled by setting the TQCR0.TQE bit to 0. If this bit is set to 0, the TQSR0.EMPTY flag is set as described below:

- immediately: if a message from the transmit queue is not scheduled for the next transmission and is not being transmitted
- after the transmission completion, error detection on the CAN bus, loss of arbitration, or transition to CH_HALT or GL_HALT mode: if a message from the transmit queue is scheduled for transmission or is being transmitted.

Note: The transmit queue is disabled only when the TQSR0.EMPTY flag is set to 1 after setting the TQCR0.TQE bit to 0.

If there are other pending messages in the transmit queue, they will be lost, so their transmission needs to be requested again.
Before the TQCR0.TQE bit is set to 1 again, ensure that the TQSR0.EMPTY bit is set to 1 and that there is no pending abort request from the transmit queue.
When the TQE bit is set to 0, all messages in the transmit queue buffers will be lost and no further message should be stored into the transmit queue.
When a message has been stored to the transmit queue, 000000FFh must be written in to the TQPCR0 register. This will set the transmit request automatically and change the internal message buffer pointer to the next free message buffer location of the transmit queue.

Note: If two messages with the same ID are stored in the transmit queue, then the order of transmission of these messages could be different from the order in which they were stored in the transmit queue.
To avoid this condition, it is important to confirm that the previous message with the same ID was successfully transmitted before a new message with the same ID is stored in the transmit queue.

For the transmit queue a dedicated interrupt can be enabled by setting the TQCR0.TQIE bit.
The interrupt mode can be configured with the TQCR0.TQIM bit either to generate an interrupt for every transmitted message or for the last transmitted message.
The transmit queue transmission request procedure after configuration is shown in Figure 36.48.

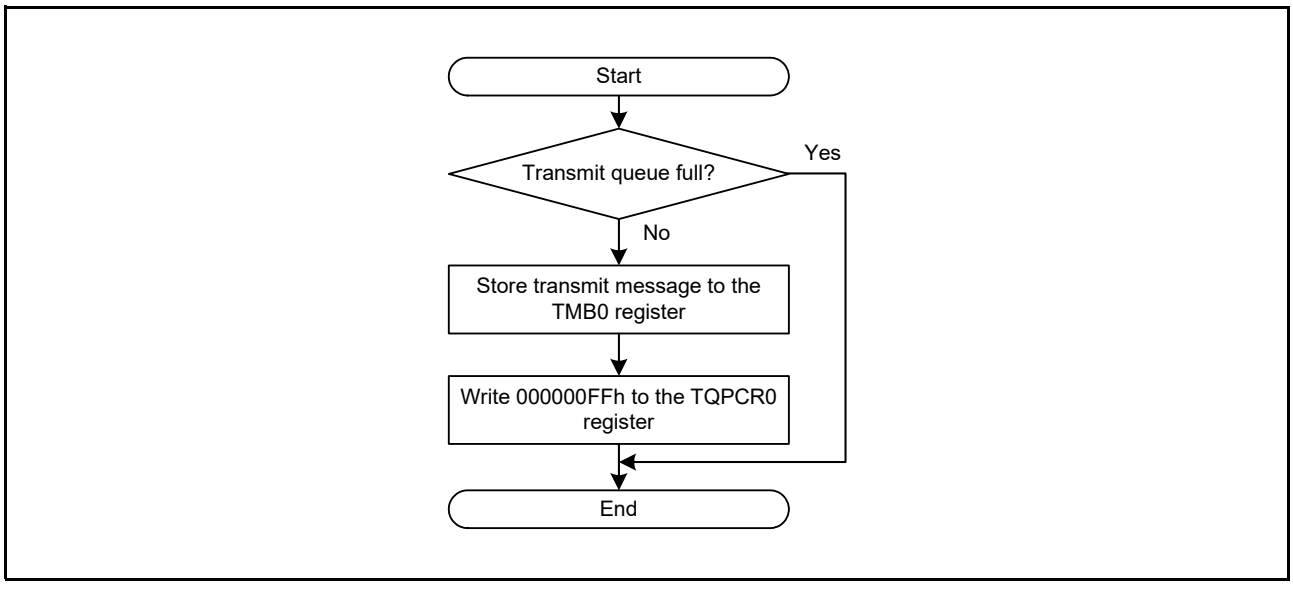


Figure 36.48 Transmit Queue Transmission Request

36.7.2.5 Transmission History

The transmission history function records the information of the successfully transmitted message in the transmission history buffer. The transmission history buffer can store up to eight transmission history entries.
The THCR.THRC bit can select whether to store information only for messages transmitted from transmit FIFO and transmit queue, or also for messages transmitted from the transmit message buffers.
Each transmit message can be individually configured for acceptance to the transmission history by the CFB0.HF0.THENT bit.
The message information is stored to the transmission history buffer of a CAN channel after the message is successfully transmitted on that CAN channel.
Storing to the list is not synchronized with the status of TMSRn.TXRF[1:0] flags.
Due to internal processing, the storage to the list could happen with a delay after the successful transmission indication.
Storing the transmission history data can be recognized by the condition that the THSR.THIF flag is set to 1 when the THCR.THIE bit is set to 1 or when the transmission history counter (THSR.FLVL[3:0]) is increased.
Multiple events could occur, such as reception scans and internal message routing.
Maximum delay time from setting the TMSRn.TXRF[1:0] flags to storing the transmission history data is 76 PCLKB cycles.

The transmission history records following information of the transmitted message:

- Transmitted Buffer type:
  – 001: Transmit message buffer
  – 010: Common FIFO in transmit FIFO mode (hereinafter referred to as transmit FIFO)
  – 100: Transmit queue
- Transmitted Buffer number:
  Transmit message buffer, transmit queue message buffer or transmit message buffer link for the common FIFO from which the transmission occurred. The number depends upon the buffer type, refer to Table 36.24.
- Transmitted pointer:
  Pointer set in header field 2 of transmit message (HF2.PTR[15:0])
- Transmitted timestamp:
  Message timestamp captured at capture point as set by the GFDCFG.TSCPS[1:0] bits
- Transmitted information label:
  Information label set in header field 2 of transmit message (HF2.IFL[1:0])

Table 36.24 Transmission History Buffer Number Entry

Transmitted Buffer Number (THACR0.BN[1:0])	Transmitted Buffer Type (THACR0.BT[2:0])		
	001b	010b	100b
	Transmit Message Buffer	Transmit FIFO	Transmit Queue
00b	TXMB0	The value of the BN[1:0] bits corresponds to the setting of the CFCR0.LTM[1:0] bits.	The value of the BN[1:0] bits indicates the number of the transmit message buffer from which the message was transmitted.
01b	TXMB1		
10b	TXMB2		
11b	TXMB3		

The transmission pointer is used to identify which message of a transmit FIFO or transmit queue has been successfully transmitted because the transmit FIFO or transmit queue number alone is not sufficient.
Therefore, a unique number (pointer) can be attached to each transmission message stored in a transmit FIFO or transmit queue. This unique pointer should be written to the CFB0.HF2.PTR[15:0] part for a transmit FIFO or to the TMB0.HF2.PTR[15:0] part of the transmit queue.
When the message is successfully transmitted then this pointer is stored together with the other message related

information to the transmission history and can be read via the transmission pointer field (PTR[15:0]) in the Transmission History Access Register.
Also for normal transmit message buffers, the TMBn.HF2.PTR[15:0] part will be stored in the transmission history. An information label is the same.
Read access to the Transmission History Access Register will be done for every single entry.
After reading one entry, 000000FFh has to be written to the THPCR register to be able to access the next entry until transmission history is empty.
Figure 36.49 shows an example flow for processing the transmission history.
The transmission history has dedicated interrupts, which can be configured with the THCR.THIM bit and enabled with the THCR.THIE bit, either to generate an interrupt when the transmission history reached a filling level of 75% or for every new transmission history entry.
Loss of transmission history is indicated by the THSR.LOST flag. Status of this flag is also shown by the GESR.THLDF flag.

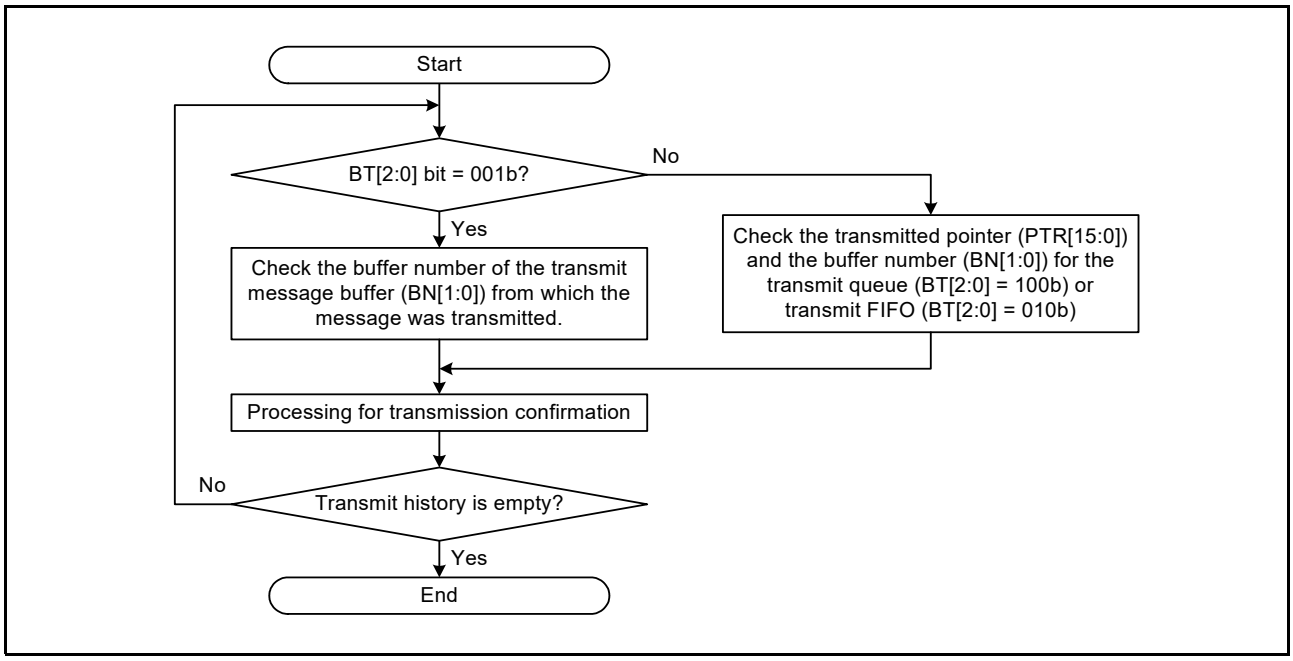


Figure 36.49 Example of Transmission History Processing Flow

36.7.2.6 Transmit Data Padding

If the data length code (DLC) of the transmit message is higher number of data bytes than the buffer size, the data bytes beyond the restricted range will be replaced by bytes with the value of CCh.
This could occur for common FIFO configured in transmit FIFO mode when the DLC of the transmit message is larger than the payload size set in the CFCR0.PLS[2:0] bits.
This can also occur in FD only mode, if the DLC value for a Classical CAN frame is greater than 8.

36.8 ECC Check

Message buffer RAM has ECC function of the 2-bit error detection and 1-bit error detection/correction.*1 The ECC module appends 7-bit ECC data to 32-bit RAM data.

Note 1. The ECC module cannot detect errors larger than 3 bits. In this case, the ECC module detects a 1-bit or 2-bit error, does not detect the error, or corrects the error bit to error data by configuration. If all RAM data is fixed at 0 or 1, it will be detected as a 2-bit ECC error.

36.8.1 ECC Function Setting

Figure 36.50 shows a procedure for ECC function setting.

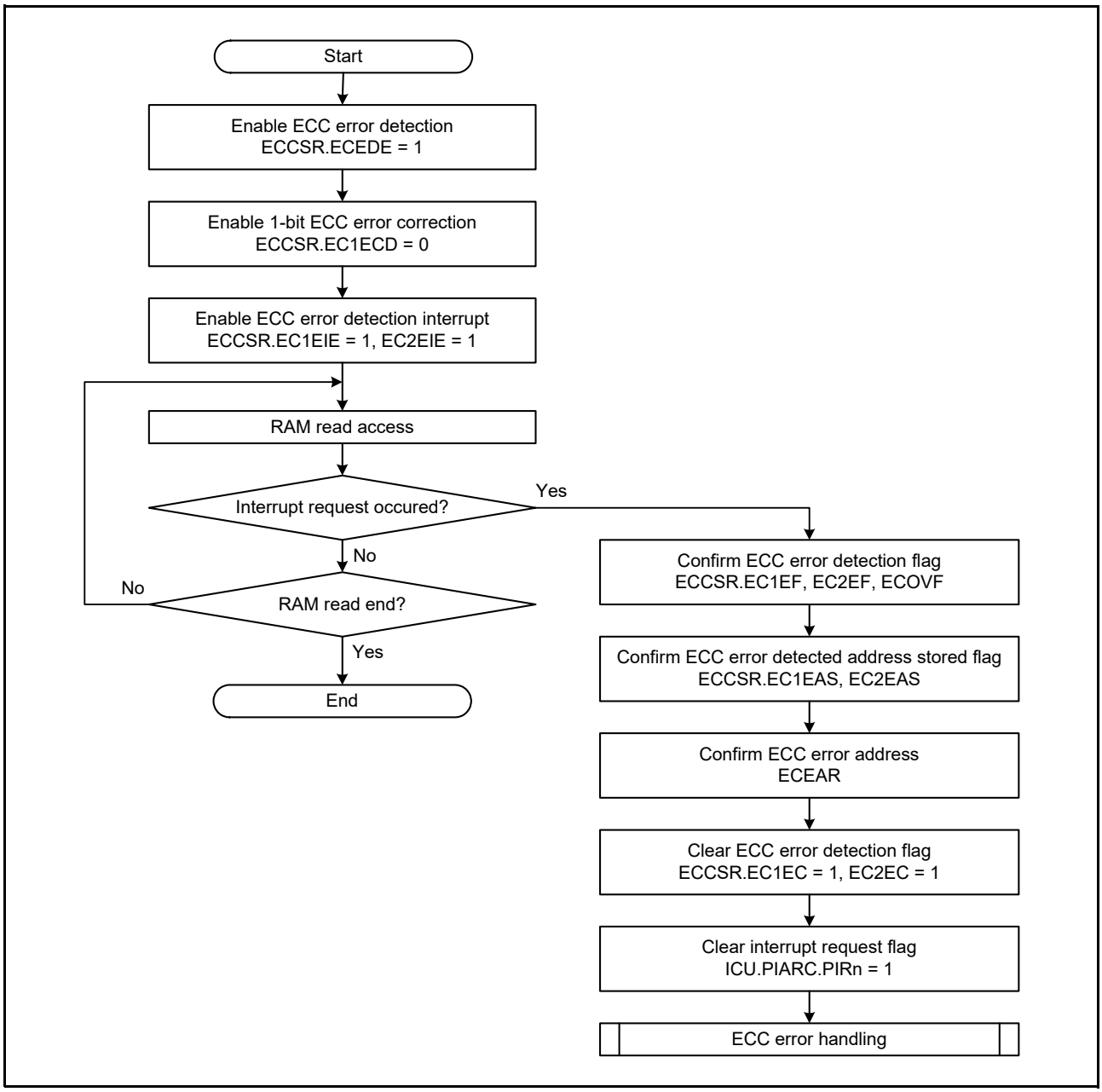


Figure 36.50 Setting Procedure for ECC Function

36.8.2 ECC Decoder Testing

ECC interrupts can be intentionally generated by ECC test mode. Figure 36.51 shows a procedure for ECC decoder testing.

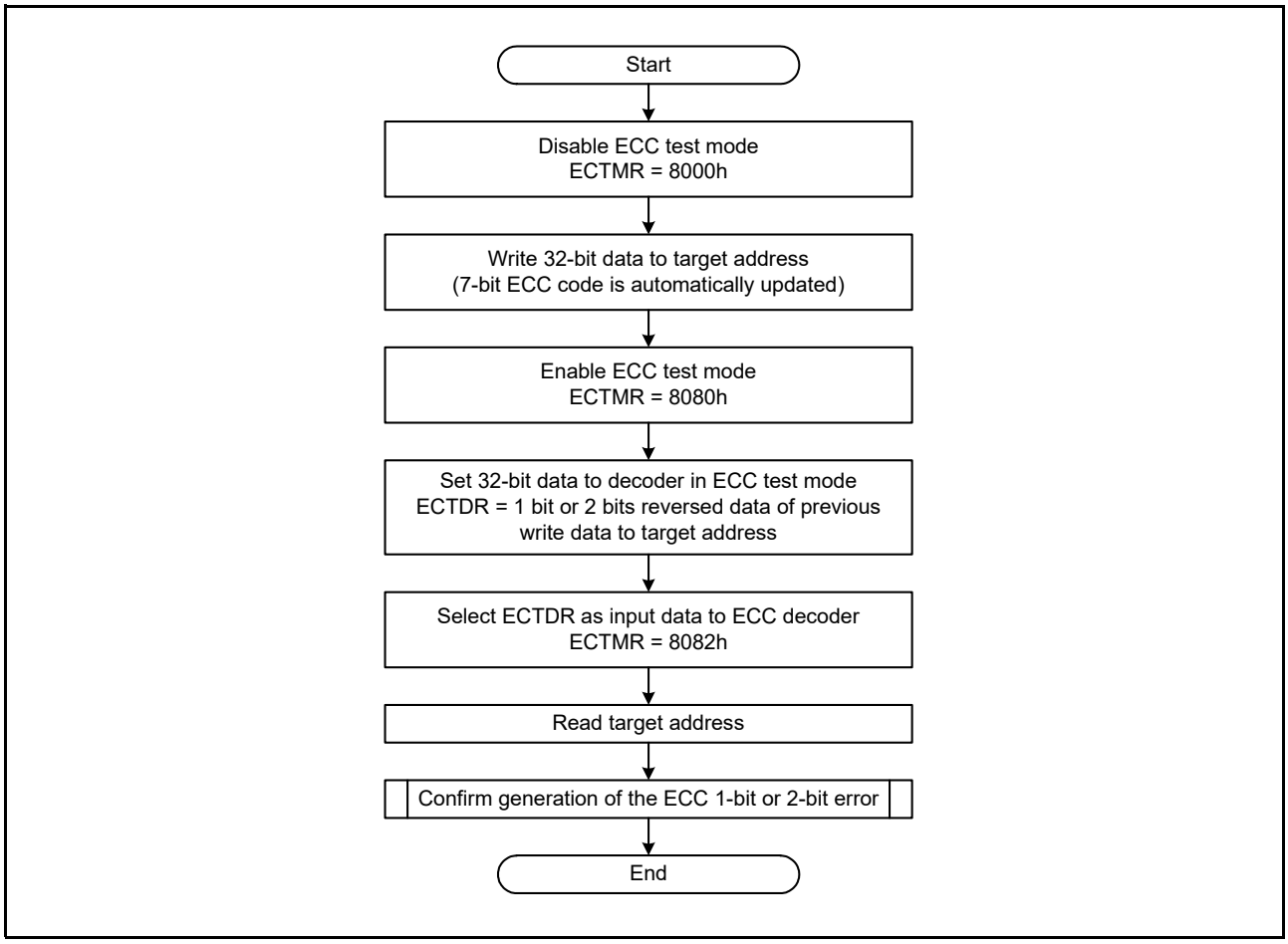


Figure 36.51 Testing Procedure for ECC Decoder

36.9 Test Mode

The CANFD module can be configured into test modes to allow testing of certain features. These features are provided only for special purposes and care must be taken when configuring the CANFD module in the test modes.

Note: All test modes are mutually exclusive unless it is explicitly stated that some functions can be enabled across other test modes.

Do not enable any combinations of the various test modes specified in this section.
The test modes can be broadly split into two groups:

- Channel specific test modes
- Global test modes

36.9.1 Channel Specific Test Modes

CAN channel can be configured into following test modes:

- Basic test mode
- Listen-only mode
- Self test mode 0 (external loop back mode)
- Self test mode 1 (internal loop back mode)
- Restricted operation mode

36.9.1.1 Basic Test Mode

The basic test mode should be used when a particular test setting needs to be enabled other than when in listen-only and self-test modes.

36.9.1.2 Listen-Only Mode

The ISO 11898-1 recommends an optional bus monitoring mode. In this mode, the CAN channel can receive valid data frames and valid remote frames. However, it sends only recessive bits on the CAN bus and is not allowed to transmit. If the CAN engine is required to send a dominant bit (ACK bit, overload flag, active error flag), the bit is rerouted internally so that the CAN engine monitors this dominant bit, although the CTX0 pin remain in recessive state.
This mode can be used for bit rate detection. In this mode, an error interrupt is generated if a bus error occurs and the interrupt is enabled.
In this mode, it is not permitted to request transmission from any transmit message buffer, transmit queue, or common FIFO.

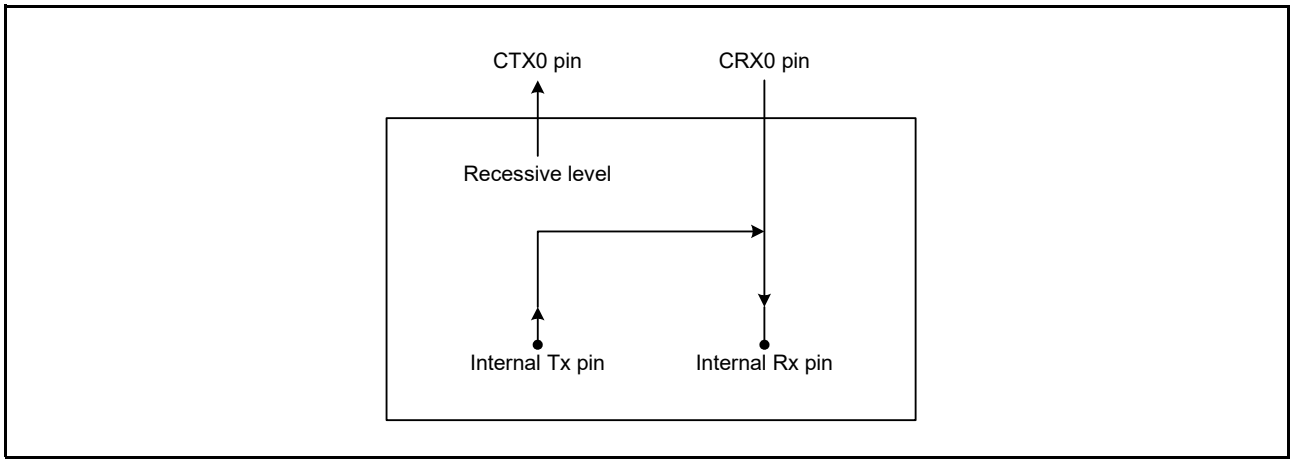


Figure 36.52 Listen-only Mode Configuration

36.9.1.3 Self Test Mode 0 (External Loop Back Mode)

In self test mode 0, the CAN engine treats its own transmitted messages as received messages via the CAN transceiver and can store them into its receive message buffers.
To be independent from external stimulation the engine generates its own acknowledge bit. This test can be used for CAN transceiver tests.
The CRX0 and CTX0 pins should be connected to the transceiver.

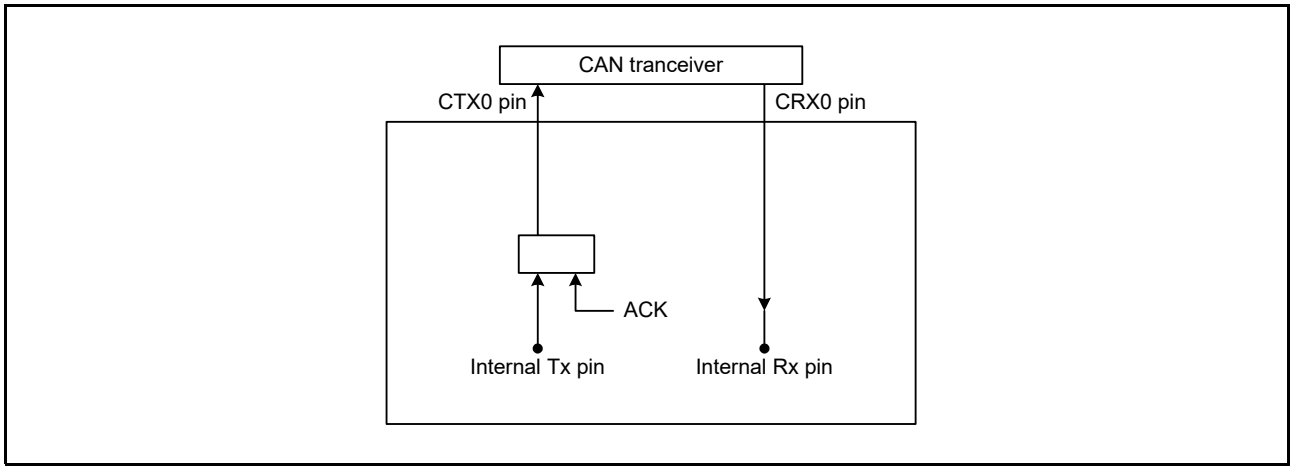


Figure 36.53 Self Test Mode 0 Configuration

36.9.1.4 Self Test Mode 1 (Internal Loop Back Mode)

In self test mode 1, the CAN engine treats its own transmitted messages as received messages and stores them into the receive buffer. This mode is provided for self-test functions. To be independent from external stimulation the CAN engine generates its own acknowledge bit. In this mode the CAN engine performs an internal feedback from internal Tx pin to internal Rx pin. The actual input level of the CRX0 pin is disregarded by the CAN engine.
The CTX0 pin outputs only recessive bits. The CRX0 and CTX0 pins do not need to be connected to the CAN bus or any external device.

Note: The channel pins are also disconnected from the Internal CAN bus communication line.

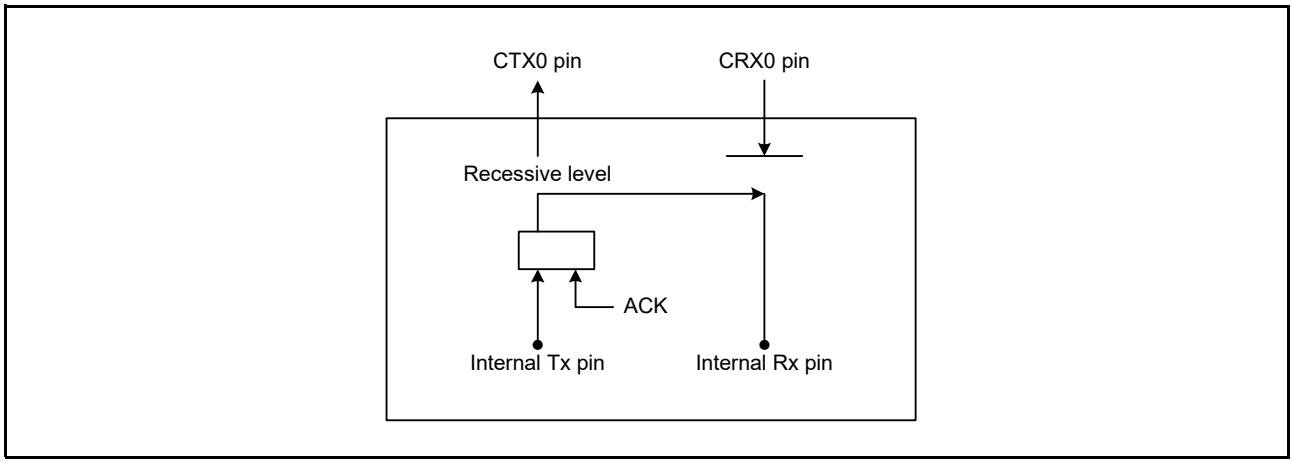


Figure 36.54 Self Test Mode 1 Configuration

36.9.1.5 Restricted Operation Mode

In restricted operation mode the CAN node is able to receive valid data and remote frames generating the acknowledge bit.
Active error and overload frames cannot be transmitted instead it waits for the occurrence of bus idle condition to resynchronize itself to the CAN communication after an error or overload condition occurs.
Moreover the receive error counter (CHSR.REC[7:0]) and transmit error counter (CHSR.TEC[7:0]) are frozen independently from the occurrence of errors.
The mode is specified as in ISO 11898-1; however it is permitted to set any transmit requested.

36.9.2 Global Test Modes

The CANFD module can be configured into following test modes:

- RAM test mode
- Bit flip test

For following test modes are protected by a special software procedure to enable the mode. This software procedure enables the write access to the test mode by specific unlock key, the related unlock key can be seen in the table below:

Table 36.25 Test Mode Unlock Key

Test Mode	Unlock Key 1	Unlock Key 2
RAM Test Mode	00007575h	00008A8Ah

If the software sequence of the two consecutive unlock key write accesses is interrupted by any other write access to the SFR or if incorrect data is written to the Global Test Mode Lock Key Register then the corresponding test mode cannot be set and the sequence should be re-started.
After the two unlock key write accesses, the next write access should be to set the corresponding test mode enable bit. If this is not followed, the unlock mechanism resets and the test mode enable bit cannot be set and then the unlock sequence should be restarted.

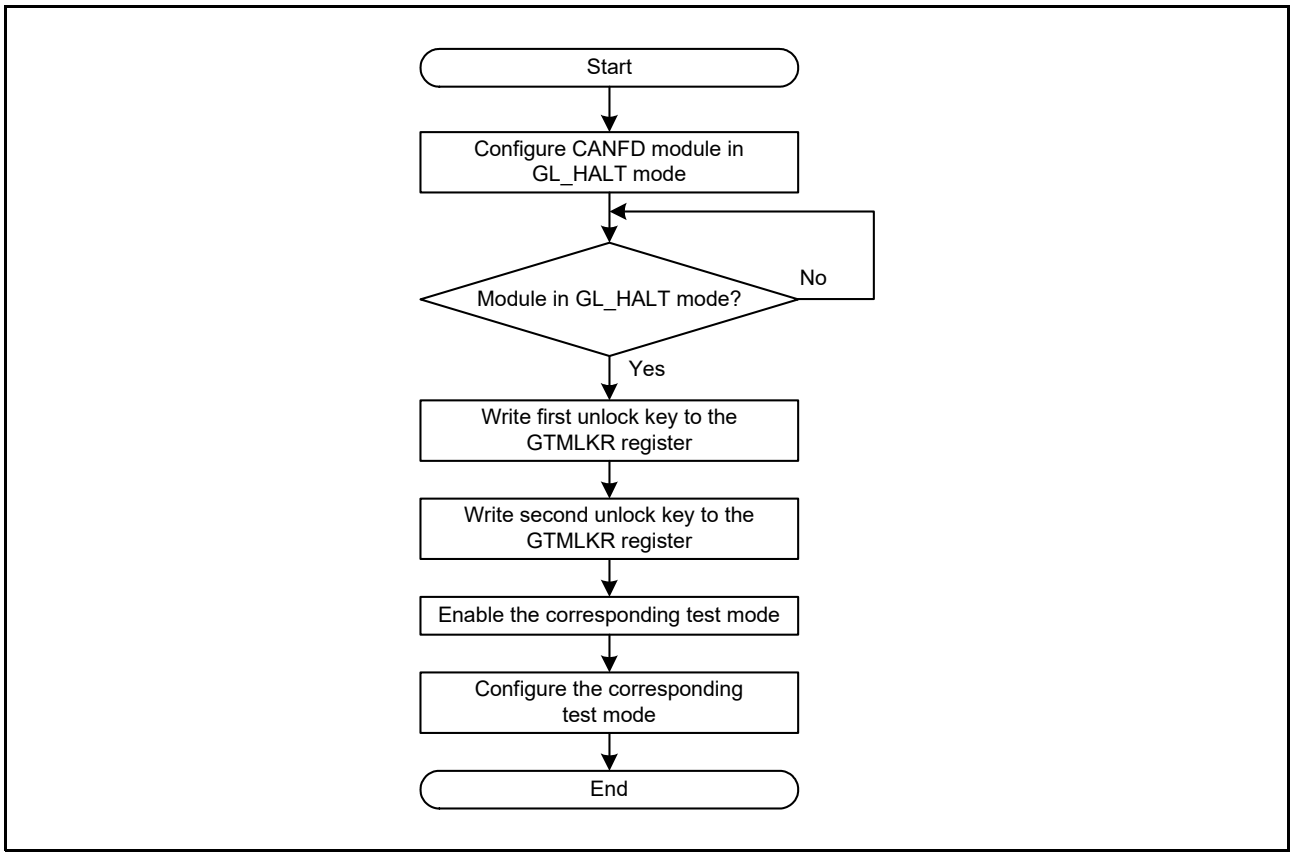


Figure 36.55 Unlock Software Protection Routine

36.9.2.1 RAM Test Mode

The CANFD module can be configured in RAM test mode by setting the GTMER.RTME bit when the corresponding lock key is written before. This is a special test mode, in which, the complete RAM area can be accessed.

Note: The actual RAM size is bigger than the RAM area initialized after MCU reset. Hence, ECC error flag (of the ECC macro) may be set if CPU reads data from this uninitialized RAM area while CANFD module is in RAM test mode.

In this mode, the RAM area is split into number of pages of 256 bytes each. Which can be accessed via RTPARk register (k = 0 to 63).
The page should be selected for read/write access by writing to the GTMCR.RTPS[3:0] bits. Then, data can be read from or written in to the RAM Test Page Access Register.
Figure 36.56 shows the structure of the pages in the RAM when performing a RAM test mode.

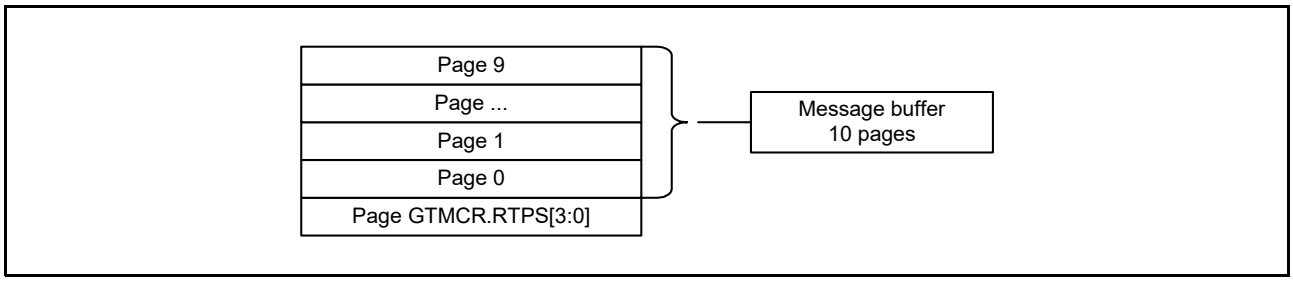


Figure 36.56 RAM Page Structure

The total available RAM size is, 2328 bytes for the message buffer RAM.
Total number of pages for the RAMs and the GTMCR.RTPS[3:0] values are calculated in the following way:

Total number of pages = ceil (total RAM size in bytes / number of bytes per page)

Message buffer RAM:

Total number of pages = ceil (2328 / 256) = 10 pages
GTMCR.RTPS[3:0] = 0 to 9

Figure 36.57 below shows the software flow for RAM test mode.

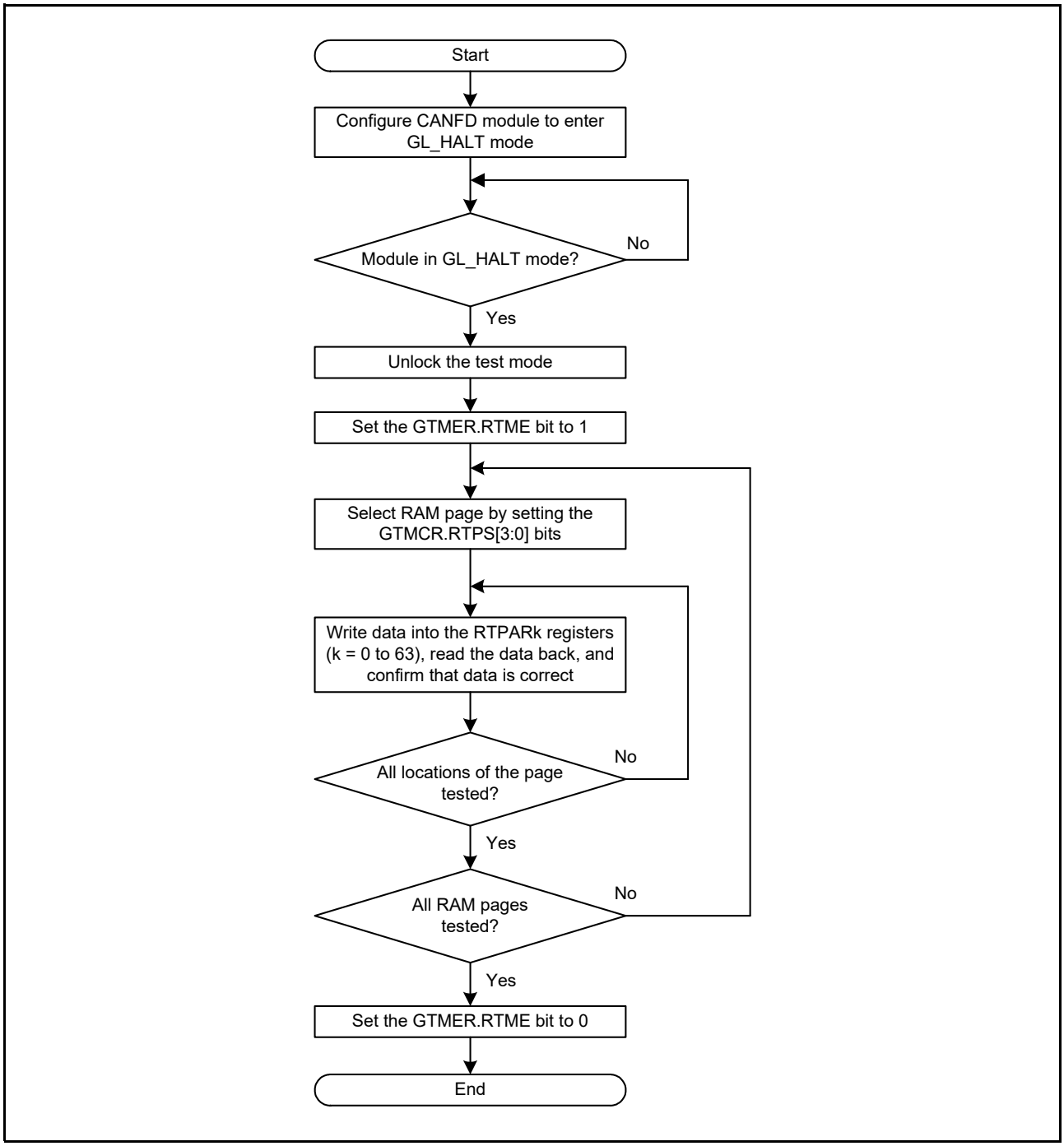


Figure 36.57 RAM Test Mode Software Flow

To exit RAM test mode, set the GTMER.RTME bit to 0.
The GTMER.RTME bit is automatically set to 0 when the CANFD module enters GL_RESET mode from the test mode.

36.9.2.2 Bit Flip Test

Bit flip test can invert the bit (the first bit of ID) of the beginning of the bit stream to receive.
If this function is used by a transmitting node, a bit error or an arbitration lost will occur.
If this function is used by a receiving node, a CRC error or a stuff error will occur.
Users should refer to the bit stuffing rule when using this feature, as there is the possibility of receiving a stuff error (due to the inversion) rather than a CRC error.
The following sequence should be used to perform CRC error testing. In the sequence below CANFD module is the receiver.

1. Set the CTR.BFT bit to 1, in order to invert the first bit of the incoming bit stream from sending node
2. Wait for the can_cherr_int output signal to set to 1
3. Read either the CHESR.CRC15[14:0] or FDCRC.CRC21[20:0] bits (depending on the received frame type: Classical CAN or CAN FD). The value should be different from the received CRC value of the reference message from sending node.
4. Check that CHESR.CEDF is 1
   As the CRC generator logic is shared for receive and transmit there is no need to create a separate transmit CRC error test.

36.10 Interrupts and DTC/DMA Requests

36.10.1 CANFD Interrupts

The CANFD module generates several Interrupts. The interrupt output, which is connected to the interrupt controller, can be controlled by the corresponding interrupt enable bit.
The status flag will be set independent from this enable bit.
The channel transmit interrupt has an additional status flag register; these status bits will only be set when the corresponding interrupt enables are set.
This register supports the identification of the interrupt source for the channel transmission, as this interrupt is driven by several trigger sources.
The interrupts in the CANFD module can be classified into two groups: global interrupts and channel interrupts.

(1) Global Interrupts

The CANFD module can generate the following three global interrupts.

1. Receive FIFO interrupt
2. Global error interrupt
3. Receive message buffer interrupt

(2) Channel Interrupts

The CAN channel can generate the following three channel interrupts.

1. Channel transmit interrupt
   (a) Successful transmission interrupt
   (b) Transmission abort interrupt
   (c) Transmit queue interrupt
   (d) Common FIFO transmission interrupt
   (e) Transmission history interrupt
2. Channel error interrupt
3. Common FIFO receive interrupt

The interrupts are cleared when the corresponding flag bits are cleared or interrupt enable bits are cleared.

Table 36.26 lists the interrupt sources for CANFD module.
To clear each interrupt request, clear all flags set to 1 from among the sources for which interrupts are enabled. The interrupt request can also be cleared by setting all corresponding interrupt enable bits to 0.

Table 36.26 Interrupt Sources

Interrupt Name			Interrupt Source Flag	Interrupt Enable Bit	Interrupt Status Flag
Global interrupts	Receive FIFO interrupt (RFRI)		RFSRn.RFIF	RFCRn.RFIE	—
	Global error interrupt (GLEI)		GESR.DEDF GESR.MLDF GESR.THLDF GESR.PODF	GCR.DEIE GCR.MLIE GCR.THLIE GCR.POIE	—
	Receive message buffer interrupt (RMRI)		RMNDR.NDF[n]	RMIER.RMIEn	—
Channel interrupts	Channel transmit interrupt (CHTI)	Transmission successful interrupt*1	TMSRn.TXRF[1]	TMIER0.TMIEn	TISR.TSIF0
		Transmission abort interrupt*1	TMSRn.TXRF[1:0] (TXRF[1:0] = 01b)	CHCR.TAIE	TISR.TAIF0
		Transmit queue interrupt	TQSR0.TQIF	TQCR0.TQIE	TISR.TQIF0
		Common FIFO transmission interrupt	CFSR0.CFTIF	CFCR0.CFTIE	TISR.CFTIF0
		Transmission history interrupt	THSR.THIF	THCR.THIE	TISR.THIF0
	Channel error interrupt (CHEI)		CHESR.BEDF CHESR.EWDF CHESR.EPDF CHESR.BOEDF CHESR.BORDF CHESR.OLDF CHESR.BLDF CHESR.ALDF FDSTS.ECOV FDSTS.SCOV FDSTS.TDCV	CHCR.BEIE CHCR.EWIE CHCR.EPIE CHCR.BOEIE CHCR.BORIE CHCR.OLIE CHCR.BLIE CHCR.ALIE CHCR.ECOVIE CHCR.SCOVIE CHCR.TDCVIE	—
	Common FIFO receive interrupt (CFRI)		CFSR0.CFRIF	CFCR0.CFRIE	—

Note 1. These interrupts are only generated for transmit message buffers that are not part of a valid transmit queue and are not linked to a common FIFO. The common FIFO and the transmit queue have other interrupts respectively.

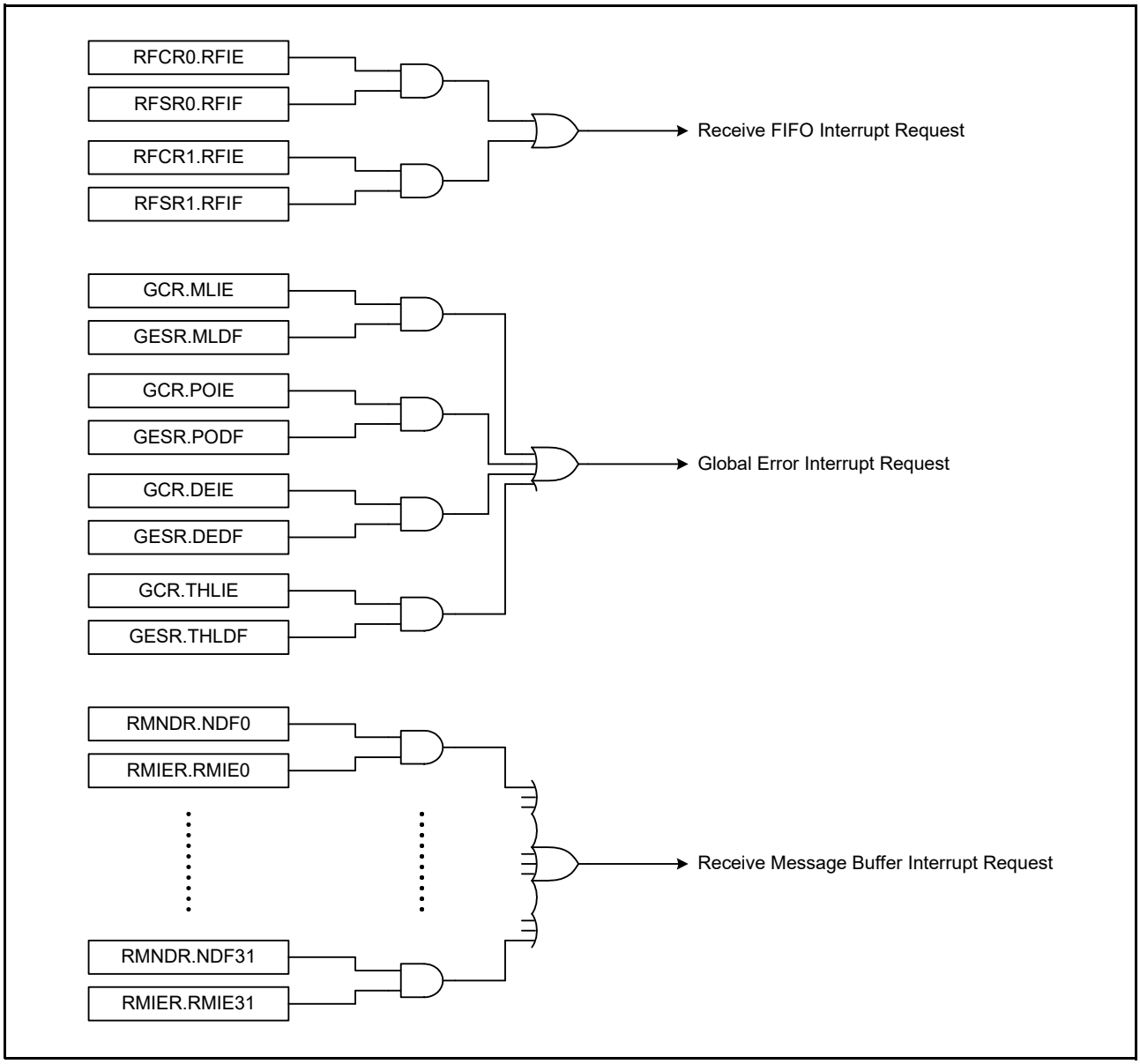


Figure 36.58 Global Interrupt Block Diagram

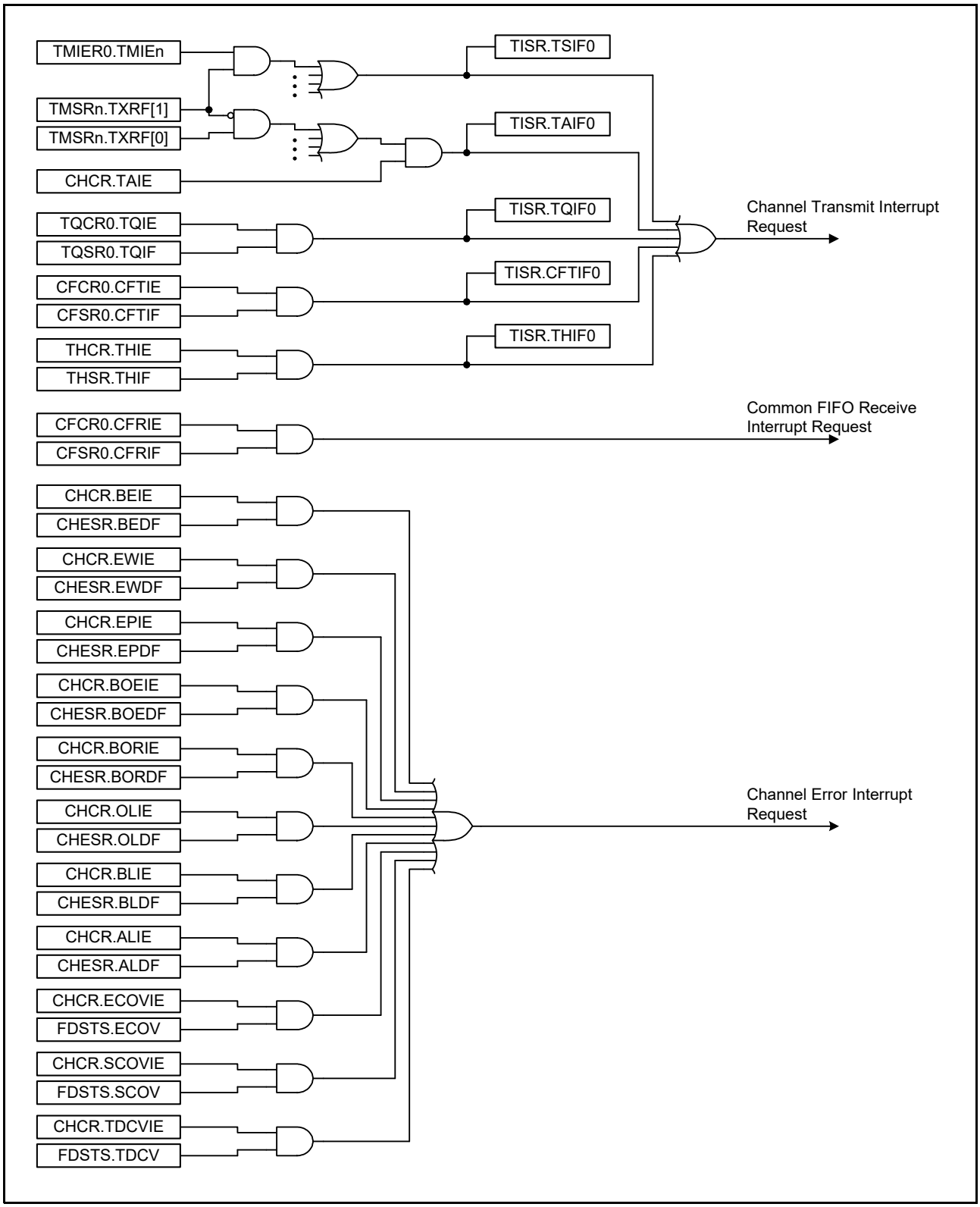


Figure 36.59 Channel Interrupt Block Diagram

36.10.2 ECC Interrupts

There are three types of interrupts generated by the ECC decoder:

- 1-bit ECC error detection interrupt
- 2-bit ECC error detection interrupt
- ECC overflow interrupt.

36.10.3 DTC/DMA Transfer Requests

The CANFD module has message buffers that can be read by DTC/DMA transfer:

- Two receive FIFO message buffers
- Common FIFO message buffer

A DTC/DMA transfer request is generated when the DTCR.RFDTE0, RFDTE1, or CFDTE0 bits are set to 1 and the corresponding FIFO is not empty.
For FIFOs with DTC / DMA transfer enabled, disable receive FIFO interrupts (the RFCR0.RFIE, RFCR1.RFIE or CFCR0.CFRIE bits).
Use the regular start address for the DMA access window address.
When the data of the specified payload size (the RFCR0.PLS[2:0], RFCR1.PLS[2:0] or CFCR0.PLS[2:0] bit) is read*1, the FIFO read pointer is automatically read.
When DTC/DMA transfer is permitted, do not write to the FIFO pointer control register (RFPCR0, RFPCR1, or CFPCR0).

Note 1. The DTC/DMA should read the exact length of the specified data payload size (the RFCR0.PLS[2:0], RFCR1.PLS[2:0] or CFCR0.PLS[2:0] bit).

The permission for DTC/DMA transfer (the DTCR.RFDTE0, RFDTE1 or CFDTE0 bit) can be set to 1 at any time. Figure 36.60 shows the DTC/DMA transfer configuration flow.

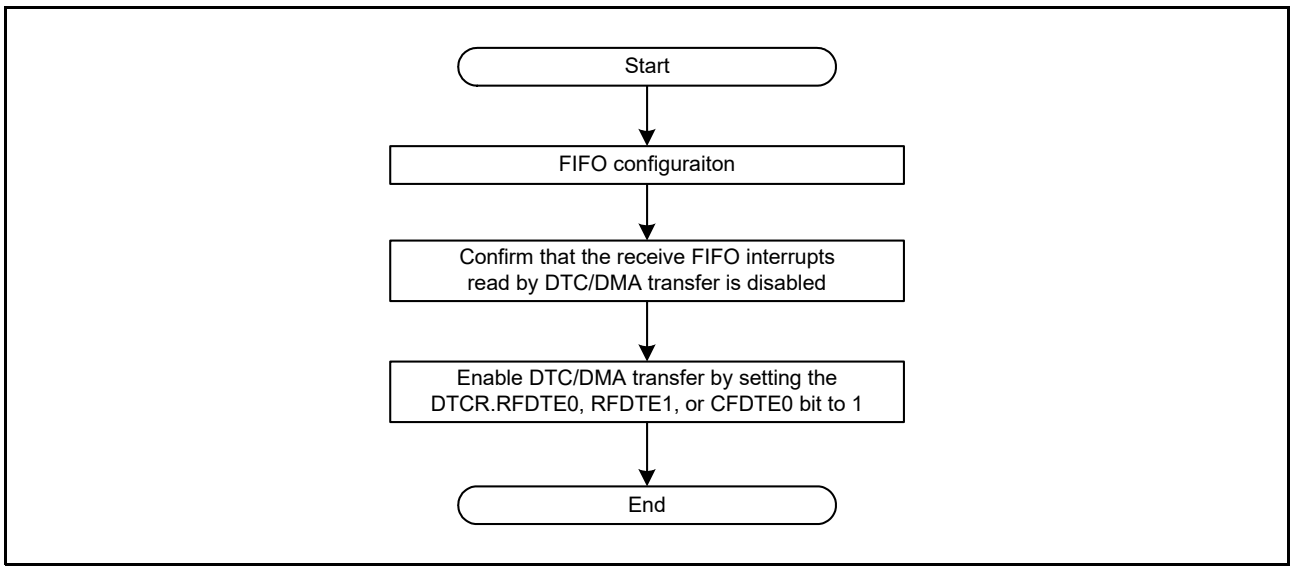


Figure 36.60 DTC/DMA Transfer Enable Flow

To disable the DTC/DMA transfer, set the corresponding DMA transfer enable bit (the DTCR.RFDTE0, RFDTE1 or CFDTE0 bit) to 0. If the disable is made during a DTC/DMA transfer, wait until the ongoing transfer is complete before performing the following operations. The transfer status can be confirmed by the DTSR.RFDTS0, RFDTS1 or CFDTS0

bit. Figure 36.61 shows the DTC/DMA transfer disable flow.
When the DTC/DMA transfer is disabled then consider what to do with the remaining or new incoming messages to this particular reception FIFOs. When the FIFO is not disabled then reception to the FIFO will continue.

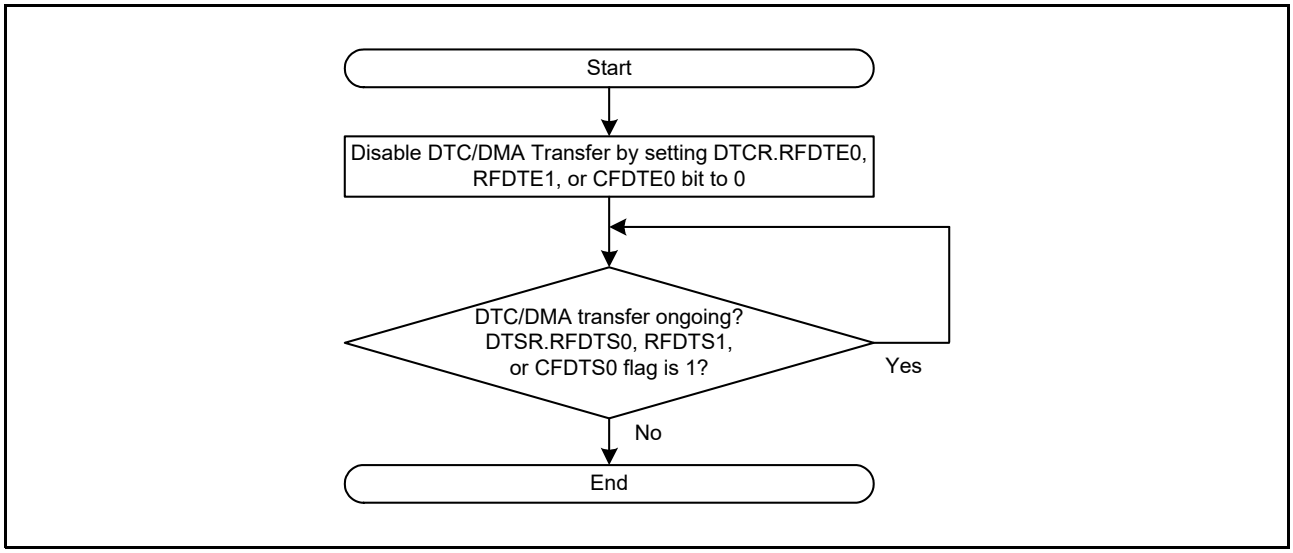


Figure 36.61 DTC/DMA Transfer Disable Flow

36.11 Usage Notes

36.11.1 Setting the Module Stop Function

The module-stop control register D (MSTPCRD) can be used to stop and start the CANFD module operations. After release from a reset, the CANFD module is placed in the module-stop state. The registers of the module become accessible after release from the module-stop state. For details, refer to section 11, Low Power Consumption.

36.11.2 Note on Configuration of Receive Message Buffers and FIFO Buffers

The maximum memory size available for receive message buffers and FIFO buffers is 1216 bytes.
For example, if all payload sizes are set to 8 bytes, each message size becomes 20 bytes, so the total number of messages must be 60 or less. If 32 receive message buffers are reserved, the total FIFO depth must be 28 or less.
Similarly, if all payload sizes are set to 64 bytes, each message size becomes 76 bytes, so the total number of messages must be 16 or less. If all FIFO depths are set to 4 messages, up to four receive message buffers can be used.
Operation is not guaranteed if the setting exceeds 1216 bytes.

37. Serial Peripheral Interface (RSPId)

In this section, “PCLK” is used to refer to PCLKA.

37.1 Overview

This MCU includes one channel of Serial Peripheral Interface (RSPI).
The RSPI channels are capable of high-speed, full-duplex or simplex synchronous serial communications with multiple processors and peripheral devices.
Table 37.1 lists the specifications of the RSPI, and Figure 37.1 shows a block diagram of the RSPI.
In this section, m as used with the RSPI command registers (SPCMDm) indicates 0 to 7.

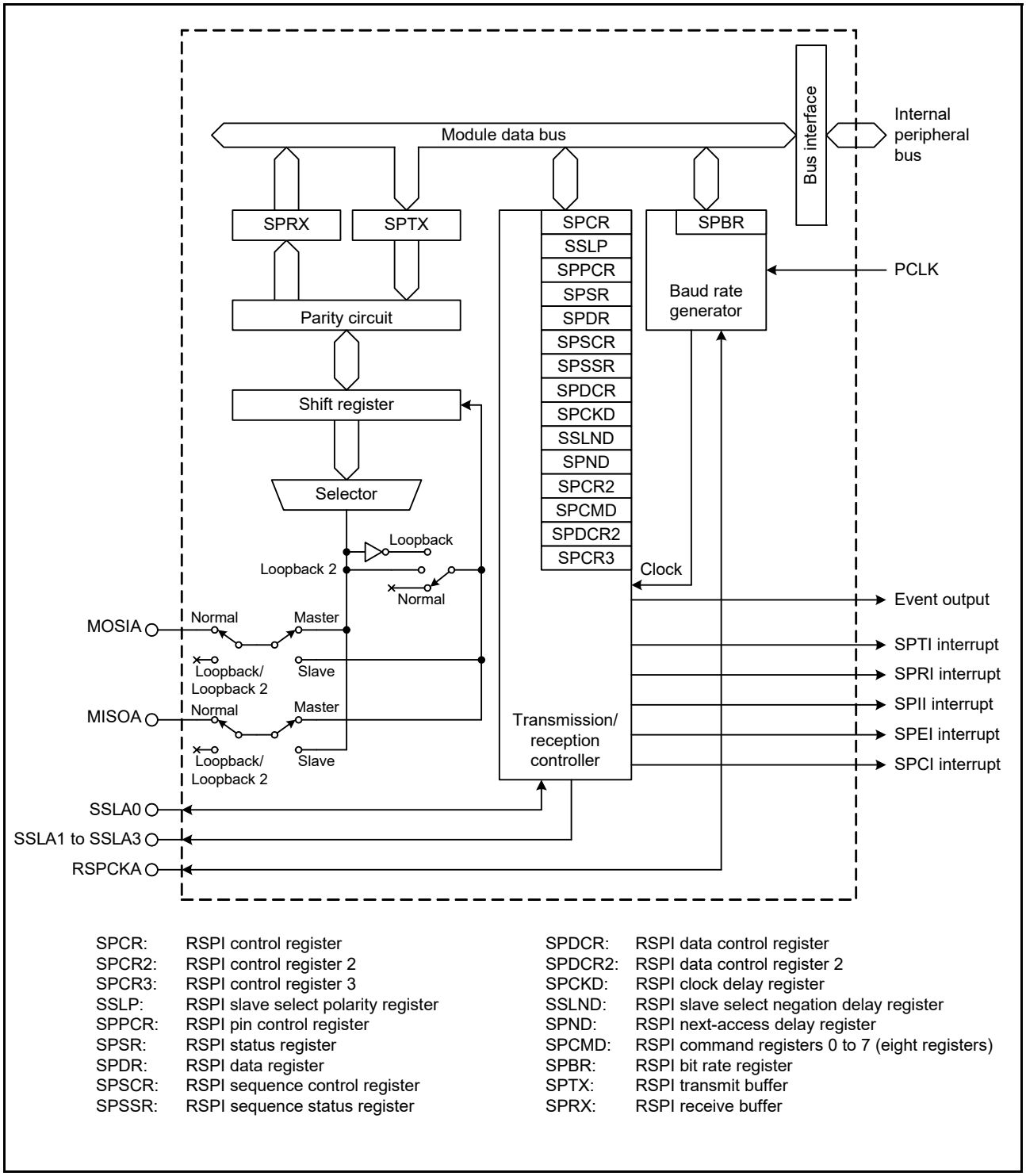
Table 37.1 RSPI Specifications (1/2)

Item	Description
Number of channels	One channel
RSPI transfer functions	• Use of MOSI (master out/slave in), MISO (master in/slave out), SSL (slave select), and RSPCK (RSPI clock) signals allows serial communications through SPI operation (4-wire method) or clock synchronous operation (3-wire method). • Communication mode: Full-duplex or simplex (transmit-only or receive-only (in slave mode)) can be selected. • Switching of the polarity of RSPCK • Switching of the phase of RSPCK
Data format	• MSB first/LSB first selectable • Transfer bit length is selectable as 8, 9, 10, 11, 12, 13, 14, 15, 16, 20, 24, or 32 bits. • 128-bit transmit/receive buffers • Up to four frames can be transferred in one round of transmission/reception (each frame consisting of up to 32 bits). • Byte swapping of transmit and receive data is selectable. • Logic level of transmit and receive data can be inverted.
Bit rate	• In master mode, the on-chip baud rate generator generates RSPCK by frequency-dividing PCLK (the division ratio ranges from divided by 2 to divided by 4096). • In slave mode, the minimum PCLK clock divided by 4 can be input as RSPCK (the maximum frequency of RSPCK is that of PCLK divided by 4). Width at high level: 2 cycles of PCLK; width at low level: 2 cycles of PCLK
Buffer configuration	• Double buffer configuration for the transmit/receive buffers • 128 bits for the transmit/receive buffers
Error detection	• Mode fault error detection • Overrun error detection*1 • Parity error detection • Underrun error detection
SSL control function	• Four SSL pins (SSLA0 to SSLA3) for each channel • In single-master mode, SSLA0 to SSLA3 pins are output. • In multi-master mode: SSLA0 pin for input, and SSLA1 to SSLA3 pins for either output or unused. • In slave mode: SSLA0 pin for input, and SSLA1 to SSLA3 pins for unused. • Controllable delay from SSL output assertion to RSPCK operation (RSPCK delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Controllable delay from RSPCK stop to SSL output negation (SSL negation delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Controllable wait for next-access SSL output assertion (next-access delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Function for changing SSL polarity
Control in master transfer	• A transfer of up to eight commands can be executed sequentially in looped execution. • For each command, the following can be set: SSL signal value, bit rate, RSPCK polarity/phase, transfer data length, MSB/LSB first, burst, RSPCK delay, SSL negation delay, and next-access delay • A transfer can be initiated by writing to the transmit buffer. • MOSI signal value specifiable in SSL negation • RSPCK auto-stop function • Delay between data bytes under burst transfer can be reduced.

Table 37.1 RSPI Specifications (2/2)

Item	Description
Interrupt sources	• Interrupt sources Receive buffer full interrupt Transmit buffer empty interrupt Error interrupt (mode fault, overrun, underrun, or parity error) Idle interrupt Communication end interrupt
Event link function (output)	• The following events can be output to the event link controller. (RSPI0) Receive buffer full event Transmit buffer empty event Error event (mode fault, overrun, underrun, or parity error) Idle event Communication end event
Others	• Function for initializing the RSPI • Loopback mode
Low power consumption function	Module stop state can be set.

Note 1. In master reception and when the RSPCK auto-stop function is enabled, an overrun error does not occur because the transfer clock is stopped at the timing of overrun error detection.



SPCR:	RSPI control register	SPDCR:	RSPI data control register
SPCR2:	RSPI control register 2	SPDCR2:	RSPI data control register 2
SPCR3:	RSPI control register 3	SPCKD:	RSPI clock delay register
SSLP:	RSPI slave select polarity register	SSLND:	RSPI slave select negation delay register
SPPCR:	RSPI pin control register	SPND:	RSPI next-access delay register
SPSR:	RSPI status register	SPCMD:	RSPI command registers 0 to 7 (eight registers)
SPDR:	RSPI data register	SPBR:	RSPI bit rate register
SPSCR:	RSPI sequence control register	SPTX:	RSPI transmit buffer
SPSSR:	RSPI sequence status register	SPRX:	RSPI receive buffer

Figure 37.1 RSPI Block Diagram

Table 37.2 lists the I/O pins used in the RSPI.
The RSPI automatically switches the I/O direction of the SSLA0 pin. SSLA0 is set as an output when the RSPI is a single master and as an input when the RSPI is a multi-master or a slave. Pins RSPCKA, MOSIA, and MISOA are automatically set as inputs or outputs according to the setting of master or slave and the level input on the SSLA0 pin. Refer to section 37.3.2, Controlling RSPI Pins for details.

Table 37.2 RSPI Pin Configuration

Channel	Pin Name	I/O	Function
RSPI0	RSPCKA	I/O	Clock I/O
	MOSIA	I/O	Master transmit data I/O
	MISOA	I/O	Slave transmit data I/O
	SSLA0	I/O	Slave selection I/O
	SSLA1	Output	Slave selection output
	SSLA2	Output	Slave selection output
	SSLA3	Output	Slave selection output

37.2 Register Descriptions

37.2.1 RSPI Control Register (SPCR)

Address(es): RSPI0.SPCR 000D 0100h

	b7	b6	b5	b4	b3	b2	b1	b0
	SPRIE	SPE	SPTIE	SPEIE	MSTR	MODFEN	TXMD	SPMS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SPMS	RSPI Mode Select*1	0: SPI operation (4-wire method) 1: Clock synchronous operation (3-wire method)	R/W
b1	TXMD	Communications Operating Mode Select*1	0: Full-duplex communications (enables the receiver) 1: Transmit-only simplex communications (disables the receiver)	R/W
b2	MODFEN	Mode Fault Error Detection Enable *1	0: Disables the detection of mode fault error 1: Enables the detection of mode fault error	R/W
b3	MSTR	RSPI Master/Slave Mode Select*1	0: Slave mode 1: Master mode	R/W
b4	SPEIE	Error Interrupt Enable	0: Disables the generation of error interrupt requests 1: Enables the generation of error interrupt requests	R/W
b5	SPTIE	Transmit Buffer Empty Interrupt Enable	0: Disables the generation of transmit buffer empty interrupt requests 1: Enables the generation of transmit buffer empty interrupt requests	R/W
b6	SPE	RSPI Function Enable	0: Disables the RSPI function 1: Enables the RSPI function	R/W
b7	SPRIE	Receive Buffer Full Interrupt Enable	0: Disables the generation of receive buffer full interrupt requests 1: Enables the generation of receive buffer full interrupt requests	R/W

Note 1. Do not change the values of the MSTR, MODFEN, TXMD, and SPMS bits while the SPE bit is 1.

SPMS Bit (RSPI Mode Select)

The SPMS bit selects SPI operation (4-wire method) or clock synchronous operation (3-wire method).
The SSLA0 to SSLA3 pins are not used in clock synchronous operation. The RSPCKA, MOSIA, and MISOA pins handle communications. If clock synchronous operation is to proceed in master mode (SPCR.MSTR = 1), the SPCMDm.CPHA bit can be set to either 0 or 1. Set the CPHA bit to 1 if clock synchronous operation is to proceed in slave mode (SPCR.MSTR = 0). Do not set the CPHA bit to 0 when clock synchronous operation is to proceed in slave mode (SPCR.MSTR = 0).

TXMD Bit (Communications Operating Mode Select)

The TXMD bit selects full-duplex communications or transmit-only simplex communications.
When performing communications with the TXMD bit set to 1, the RSPI performs only transmit operations and not receive operations (refer to section 37.3.6, Communications Operating Mode).
When the TXMD bit is set to 1, receive buffer full interrupt requests cannot be used.
When the SPCR3.RXMD bit is set to 1 (receive-only simplex communications), the setting of this bit is ignored.

MODFEN Bit (Mode Fault Error Detection Enable)

The MODFEN bit enables or disables the detection of mode fault error (refer to section 37.3.10, Error Detection). In addition, the RSPI determines the I/O direction of the SSLA0 to SSLA3 pins based on combinations of the MODFEN and MSTR bits (refer to section 37.3.2, Controlling RSPI Pins).

MSTR Bit (RSPI Master/Slave Mode Select)

The MSTR bit selects master/slave mode of the RSPI. According to MSTR bit settings, the RSPI determines the direction of pins RSPCKA, MOSIA, MISOA, and SSLA0 to SSLA3.

SPEIE Bit (Error Interrupt Enable)

The SPEIE bit enables or disables the generation of error interrupt requests when the RSPI detects a mode fault error or underrun error and sets the SPSR.MODF flag to 1, when the RSPI detects an overrun error and sets the SPSR.OVRF flag to 1, or when the RSPI detects a parity error and sets the SPSR.PERF flag to 1 (refer to section 37.3.10, Error Detection).

SPTIE Bit (Transmit Buffer Empty Interrupt Enable)

The SPTIE bit enables or disables the generation of transmit buffer empty interrupt requests when the RSPI detects when the transmit buffer is empty.

A transmit buffer empty interrupt request when transmission starts is generated by setting the SPE and SPTIE bits to 1 at the same time or by setting the SPE bit to 1 after setting the SPTIE bit to 1.

Note that a transmit buffer interrupt is generated when the SPTIE bit is 1 even if the RSPI function is disabled (the SPTIE bit is changed to 0).

SPE Bit (RSPI Function Enable)

The SPE bit enables or disables the RSPI function.

When the SPSR.MODF flag is 1, the SPE bit cannot be set to 1. For details, refer to section 37.3.10, Error Detection.

Setting the SPE bit to 0 disables the RSPI function, and initializes a part of the module function. For details, refer to section 37.3.11, Initializing RSPI. Furthermore, a transmit buffer empty interrupt request is generated by the state of the SPE bit changing from 0 to 1 or from 1 to 0.

SPRIE Bit (Receive Buffer Full Interrupt Enable)

If the RSPI has detected a receive buffer full write after completion of a serial transfer, the SPRIE bit enables or disables the generation of a receive buffer full interrupt request.

37.2.2 RSPI Slave Select Polarity Register (SSLP)

Address(es): RSPI0.SSLP 000D 0101h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	SSL3P	SSL2P	SSL1P	SSL0P
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SSL0P	SSL0 Signal Polarity Setting	0: SSL0 signal is active low 1: SSL0 signal is active high	R/W
b1	SSL1P	SSL1 Signal Polarity Setting	0: SSL1 signal is active low 1: SSL1 signal is active high	R/W
b2	SSL2P	SSL2 Signal Polarity Setting	0: SSL2 signal is active low 1: SSL2 signal is active high	R/W
b3	SSL3P	SSL3 Signal Polarity Setting	0: SSL3 signal is active low 1: SSL3 signal is active high	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Do not change the SSLP register while the SPCR.SPE bit is 1.

37.2.3 RSPI Pin Control Register (SPPCR)

Address(es): RSPI0.SPPCR 000D 0102h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	MOIFE	MOIFV	—	—	SPLP2	SPLP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SPLP	RSPI Loopback	0: Normal mode 1: Loopback mode (data is inverted for transmission)	R/W
b1	SPLP2	RSPI Loopback 2	0: Normal mode 1: Loopback mode (data is not inverted for transmission)	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	MOIFV	MOSI Idle Fixed Value	0: The level output on the MOSIA pin during MOSI idling corresponds to low 1: The level output on the MOSIA pin during MOSI idling corresponds to high	R/W
b5	MOIFE	MOSI Idle Value Fixing Enable	0: MOSI output value equals final data from previous transfer 1: MOSI output value equals the value set in the MOIFV bit	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Do not change the SPPCR register while the SPCR.SPE bit is 1.

SPLP Bit (RSPI Loopback)

The SPLP bit selects the mode of the RSPI pins.

When the SPLP bit is set to 1, the RSPI shuts off the path between the MISOA pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSIA pin and the shift register if the SPCR.MSTR bit is 0, and connects (inverts) the input path and output path for the shift register (loopback mode).

SPLP2 Bit (RSPI Loopback 2)

The SPLP2 bit selects the mode of the RSPI pins.

When the SPLP2 bit is set to 1, the RSPI shuts off the path between the MISOA pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSIA pin and the shift register if the SPCR.MSTR bit is 0, and connects the input path and output path for the shift register (loopback mode).

MOIFV Bit (MOSI Idle Fixed Value)

If the MOIFE bit is 1 in master mode, the MOIFV bit determines the MOSIA pin output value during the SSL negation period (including the SSL retention period during a burst transfer).

MOIFE Bit (MOSI Idle Value Fixing Enable)

The MOIFE bit fixes the MOSIA output value when the RSPI in master mode is in an SSL negation period (including the SSL retention period during a burst transfer). When the MOIFE bit is 0, the RSPI outputs the last data from the previous serial transfer during the SSL negation period to the MOSIA pin. When the MOIFE bit is 1, the RSPI outputs the fixed value set in the MOIFV bit to the MOSIA pin.

37.2.4 RSPI Status Register (SPSR)

Address(es): RSPI0.SPSR 000D 0103h

	b7	b6	b5	b4	b3	b2	b1	b0
	SPRF	SPCF	SPTEF	UDRF	PERF	MODF	IDLNF	OVRF
Value after reset:	0	0	1	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OVRF	Overrun Error Flag	0: No overrun error occurs 1: An overrun error occurs	R/(W) *1
b1	IDLNF	Idle Flag	0: RSPI is in the idle state 1: RSPI is in the transfer state	R
b2	MODF	Mode Fault Error Flag	0: Neither a mode fault error nor an underrun error occurs 1: A mode fault error or an underrun error occurs	R/(W) *1
b3	PERF	Parity Error Flag	0: No parity error occurs 1: A parity error occurs	R/(W) *1
b4	UDRF	Underrun Error Flag	This flag is used with the MODF flag to check the state in terms of mode fault errors and underrun errors. b4 b2 0 0: Neither a mode fault error nor an underrun error occurs 0 1: A mode fault error occurs 1 1: An underrun error occurs	R/(W) *1, *2
b5	SPTEF	Transmit Buffer Empty Flag	0: Transmit buffer has valid data 1: Transmit buffer has no valid data	R*3
b6	SPCF	Communication End Flag	0: Communication does not start or communication is in progress 1: Communication has ended	R/(W) *1
b7	SPRF	Receive Buffer Full Flag	0: Receive buffer has no valid data 1: Receive buffer has valid data	R*3

Note 1. Only 0 can be written to clear the flag after reading 1.
Note 2. Clear the MODF and UDRF flags at the same time.
Note 3. The write value should be 1.

OVRF Flag (Overrun Error Flag)

The OVRF flag indicates the occurrence of an overrun error. In master mode (when the SPCR.MSTR bit is 1) and when the RSPCK clock auto-stop function is enabled (the SPCR2.SCKASE bit is 1), an overrun error does not occur; accordingly this flag does not become 1. For details, refer to section 37.3.10.1, Overrun Error.

[Setting condition]

- When the next data reception ends while the SPCR.TXMD bit is 0 and the receive buffer is full.
- When the next data reception ends while the SPCR.MSTR bit is 0, the SPCR3.RXMD bit is 1, and the receive buffer is full.

[Clearing condition]

- When the SPSR register is read while the OVRF flag is 1, and then 0 is written to the OVRF flag.

IDLNF Flag (Idle Flag)

The IDLNF flag indicates the transfer status of the RSPI.

[Setting condition]

Master mode

- When both of the conditions in master mode under the [Clearing condition] below are not satisfied.

Slave mode

- When the SPCR.SPE bit is set to 1 (enables the RSPI function).

[Clearing condition]

Master mode

- When the SPCR.SPE bit is set to 0 (disables the RSPI function)
- When all of the following conditions are satisfied.
  1. The transmit buffer is empty (the SPTEF flag is 1)
  2. The SPSSR.SPCP[2:0] bits are 000b
  3. The transmission of the last bit has been completed and the time specified by the SSLND.SLNDL[2:0] and SPND.SPNDL[2:0] bits has elapsed

Slave mode

- When the SPCR.SPE bit is set to 0 (disables the RSPI function).

MODF Flag (Mode Fault Error Flag)

Indicates the occurrence of a mode fault error or an underrun error. The UDRF flag indicates whether the error is a mode fault error or an underrun error.

[Setting condition]

Multi-master mode

- When the input level of the SSLAi pin changes to the active level while the SPCR.MSTR bit is 1 (master mode) and the SPCR.MODFEN bit is 1 (mode fault error detection is enabled), the RSPI detects a mode fault error.

Slave mode

- When the SSLAi pin is negated before the RSPCK cycle necessary for data transfer ends while the SPCR.MSTR bit is 0 (slave mode) and the SPCR.MODFEN bit is 1 (mode fault error detection is enabled), the RSPI detects a mode fault error.
- When the serial transfer starts while the SPCR.SPE bit is 1 (enables the RSPI function) and the transmit data are not ready for output, the RSPI detects an underrun error.

The active level of the SSLAi signal is determined by the SSLP.SSLiP bit (SSLi signal polarity setting bit).

[Clearing condition]

- When the SPSR register is read while the MODF flag is 1, and then 0 is written to the MODF flag.

PERF Flag (Parity Error Flag)

Indicates the occurrence of a parity error.

[Setting condition]

- When a data reception ends while the SPCR.TXMD bit is 0 and the SPCR2.SPPE bit is 1, the RSPI detects a parity error.
- When a data reception ends while the SPCR.MSTR bit is 0, the SPCR3.RXMD bit is 1, and the SPCR2.SPPE bit is 1, the RSPI detects a parity error.

[Clearing condition]

- When the SPSR register is read while the PERF flag is 1, and then 0 is written to the PERF flag.

UDRF Flag (Underrun Error Flag)

Indicates the occurrence of an underrun error. When this flag becomes 1, the MODF flag becomes 1 too. When the MODF flag is 1 and this flag is 0, the error is a mode fault error.

[Setting condition]

- When the serial transfer starts while the SPCR.MSTR bit is 0 (slave mode), the SPCR3.RXMD bit is 0, the SPCR.SPE bit is 1 (enables the RSPI function), and the transmit data are not ready for output, the RSPI detects an underrun error

[Clearing condition]

- When 0 is written to the UDRF flag after reading the SPSR register while the UDRF flag is 1

SPTEF Flag (Transmit Buffer Empty Flag)

Indicates whether the transmit buffer (SPTX) in the RSPI data register has valid data.

[Setting condition]

- When the SPCR.SPE bit is set to 0 (disables the RSPI function).
- When the number of frames of transmit data specified by the SPDCR.SPFC[1:0] bits have been transferred from the transmit buffer to the shift register.

[Clearing condition]

- When the number of frames of transmit data specified by the SPDCR.SPFC[1:0] bits is written to the SPDR register.

The SPDR register can be set only when the SPTEF flag is 1. The data in the transmit buffer is not updated when the SPDR register is set while the SPTEF flag is 0.

SPCF Flag (Communication End Flag)

This flag indicates a completion of the RSPI communication.

[Setting condition]

Master mode

- When all of the following conditions are satisfied.
  1. The transmit buffer is empty (the SPTEF flag is 1)
  2. The SPSSR.SPCP[2:0] bits are 000b
  3. The transmission of the last bit has been completed and the time specified by the SSLND.SLNDL[2:0] and SPND.SPNDL[2:0] bits has elapsed

Full-duplex or transmit-only simplex communications in slave mode (SPI operation)

- When all of the following conditions are satisfied.
  1. The transmit buffer is empty (the SPTEF flag is 1)
  2. The transmit shift register is empty
  3. The SSLA0 signal has been negated

Full-duplex or transmit-only simplex communications in slave mode (clock synchronous operation)

- When all of the following conditions are satisfied.
  1. The transmit buffer is empty (the SPTEF flag is 1)
  2. The transmit shift register is empty
  3. The last bit of the last data has been received (the last even edge of the RSPCK)

Receive-only simplex communications in slave mode (SPI operation)

- When the SSLA0 signal is negated after the number of frames set in the SPDCR.SPFC[1:0] bits have been received.

Receive-only simplex communications in slave mode (clock synchronous operation)

- When the number of frames set in the SPDCR.SPFC[1:0] bits have been received (the last even edge of the

RSPCK).

[Clearing condition]

Full-duplex or transmit-only simplex communications

- When the next transmit data is written to the transmit buffer
- When 0 is written to the SPCF flag after reading the SPSR register while the SPCF flag is 1

Receive-only simplex communications in SPI operation

- When the assertion of the SSLA0 signal for the next data has been detected
- When 0 is written to the SPCF flag after reading the SPSR register while the SPCF flag is 1

Receive-only simplex communications in slave mode (clock synchronous operation)

- When the first edge of the RSPCK signal for the next data has been detected
- When 0 is written to the SPCF flag after reading the SPSR register while the SPCF flag is 1

SPRF Flag (Receive Buffer Full Flag)

Indicates whether the receive buffer (SPRX) in the RSPI data register has valid data.

[Setting condition]

- When the number of frames of receive data specified by the SPDCR.SPFC[1:0] bits is transferred from shift register to the receive buffer (SPRX) while the SPCR.TXMD bit is 0 (full duplex) and the SPRF flag is 0.
  Note that the SPRF flag does not become 1 when the OVRF flag is 1.
- When the number of frames of receive data specified by the SPDCR.SPFC[1:0] bits is transferred from shift register to the receive buffer (SPRX) while the SPCR.MSTR bit is 0, the SPCR3.RXMD bit is 1, and the SPRF flag is 0.
  Note that the SPRF flag does not become 1 when the OVRF flag is 1.

[Clearing condition]

- When all of the received data are read from the SPDR register.

37.2.5 RSPI Data Register (SPDR)

- When accessing in longword size

Address(es): RSPI0.SPDR 000D 0104h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- When accessing in word size

Address(es): RSPI0.SPDR.H 000D 0104h

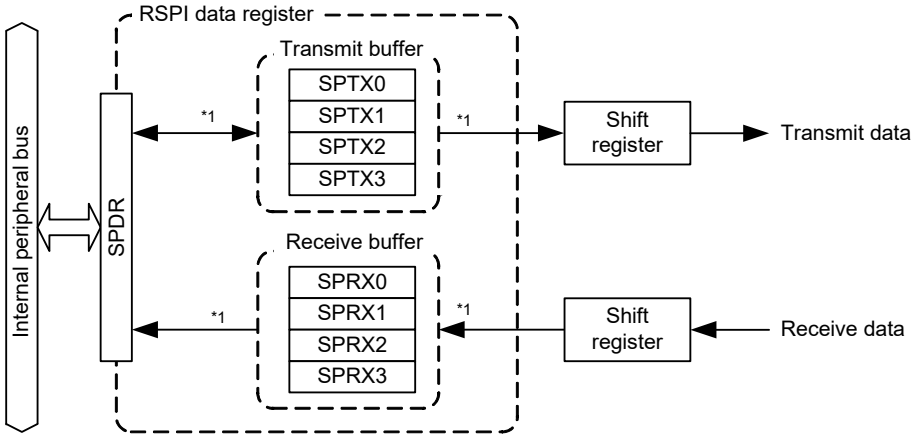
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	H															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- When accessing in byte size

Address(es): RSPI0.SPDR.HH 000D 0104h

	b7	b6	b5	b4	b3	b2	b1	b0
	HH							
Value after reset:	0	0	0	0	0	0	0	0

The SPDR register is the interface with the buffers that hold data for transmission and reception by the RSPI.
When accessing in longwords (the SPLW bit is 1 and the SPBYT bit is 0), access the SPDR register in 32-bit units.
When accessing in words (the SPLW bit is 0 and the SPBYT bit is 0), access the SPDR.H register in 16-bit units.
When accessing in bytes (the SPBYT bit is 1), access the SPDR.HH register in 8-bit units.
The transmit buffer (SPTX) and receive buffer (SPRX) are independent but are both mapped to the SPDR register.
Figure 37.2 shows the Configuration of the SPDR Register.



Note 1. The destination buffer and stage for access is automatically switched by hardware.

Figure 37.2 Configuration of the SPDR Register

The transmit and receive buffers each have four stages. The number of stages to be used is selectable by the number of frames specification bits in the RSPI data control register (SPDCR.SPFC[1:0]). The eight stages of the buffer are all mapped to the single address of the SPDR register.

Data written to the SPDR register are written to a transmit-buffer stage (SPTXn) (n = 0 to 3) and then transmitted from the buffer. The receive buffer holds received data on completion of reception. The receive buffer is not updated if an overrun is generated.

Furthermore, if the data length is other than 32 bits, bits not referred to in SPTXn (n = 0 to 3) are stored in the corresponding bits in SPRXn. For example, if the data length is 9 bits, received data are stored in the SPRXn[8:0] bits and the SPTXn[31:9] bits are stored in the SPRXn[31:9] bits.

(1) Bus Interface

The SPDR register is the interface with 32-bit wide transmit and receive buffers, each of which has four stages, for a total of 32 bytes. In other words, the 32 bytes are mapped to the 4-byte address space for the SPDR register. Furthermore, the unit of access for the SPDR register is selected by the SPDCR.SPLW bit and the SPDCR.SPBYT bit.
Data for transmission should be flush with the LSB end of the register. Received data are stored flush with the LSB end.
Operations involved in writing to and reading from the SPDR register are described below.

(a) Writing

Data written to the SPDR register are written to a transmit buffer (SPTXn). This is not influenced by the value of the SPDCR.SPRDTD bit unlike when reading from the SPDR register.
The transmit buffer includes a transmit buffer write pointer which is automatically updated to indicate the next stage each time data are written to the SPDR register.
Figure 37.3 shows the configuration of the bus interface with the transmit buffer in the case of writing to the SPDR register.

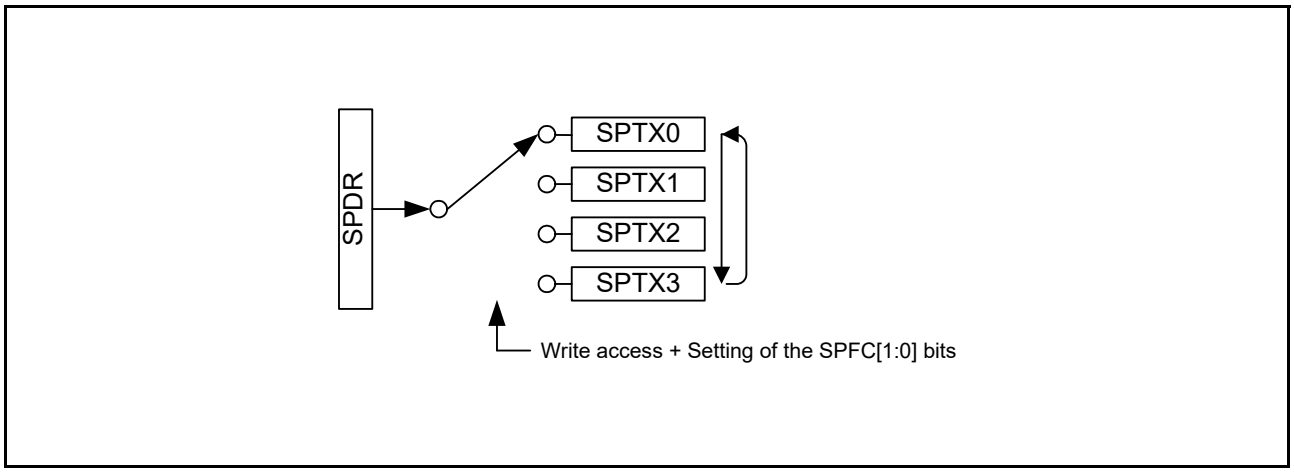


Figure 37.3 Configuration of the SPDR Register (Writing)

The sequence for switching the transmit buffer write pointer differs with the setting of the number of frames specification bits in the RSPI data control register (SPDCR.SPFC[1:0]).

- Settings of the SPFC[1:0] bits and sequence of switching the pointer among SPTX0 to SPTX3.
  When the SPFC[1:0] bits are 00b: SPTX0 → SPTX0 → SPTX0 → …
  When the SPFC[1:0] bits are 01b: SPTX0 → SPTX1 → SPTX0 → SPTX1 → …
  When the SPFC[1:0] bits are 10b: SPTX0 → SPTX1 → SPTX2 → SPTX0 → SPTX1 → …
  When the SPFC[1:0] bits are 11b: SPTX0 → SPTX1 → SPTX2 → SPTX3 → SPTX0 → SPTX1 → …

When 1 is written to the RSPI function enable bit in the RSPI control register (SPCR.SPE) while the bit's current value is 0, SPTX0 will be the destination the next time writing proceeds.
When writing to the transmit buffer (SPTXn) after generation of the transmit buffer empty interrupt (after the SPSR.SPTEF flag becomes 1), write the number of frames set by the number of frames specification bits (SPFC[1:0]) in the RSPI data control register (SPDCR). Even if the number of frames is written to the transmit buffer (SPTXn), the value of the buffer is not updated after completion of the writing and before generation of the next transmit buffer empty interrupt (while the SPSR.SPTEF flag is 0).

(b) Reading

The SPDR register can be read to read the value of a receive buffer (SPRXn) or a transmit buffer (SPTXn). The setting of the RSPI receive/transmit data select bit in the RSPI data control register (SPDCR.SPRDTD) selects whether reading is of the receive or transmit buffer.

The sequence of reading the SPDR register is controlled by independent pointers, receive buffer read pointer and transmit buffer read pointer.

Figure 37.4 shows the configuration of the bus interface with the receive and transmit buffers in the case of reading from the SPDR register.

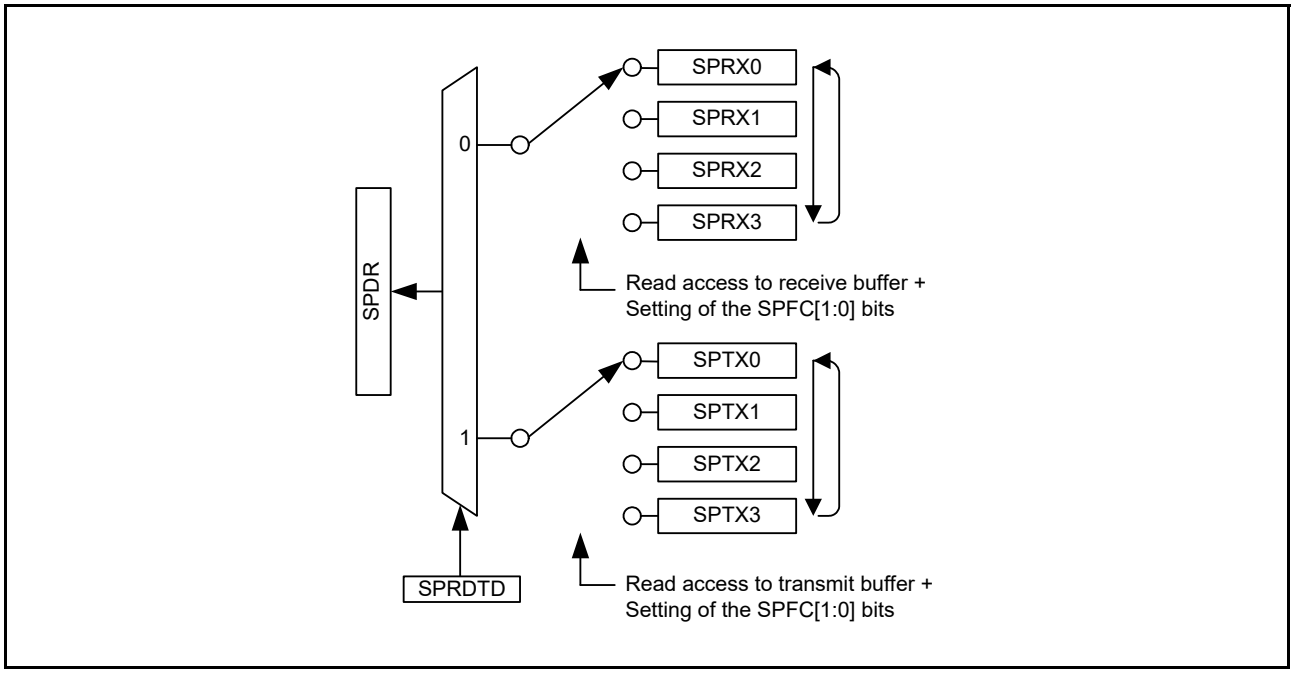


Figure 37.4 Configuration of the SPDR Register (Reading)

Reading the receive buffer switches the receive buffer read pointer to the next buffer automatically.

The sequence of switching the receive buffer read pointer is the same as that for the transmit buffer write pointer. However, when 1 is written to the RSPI function enable bit in the RSPI control register (SPCR.SPE) while the bit's current value is 0, SPRX0 will be indicated by the buffer read pointer the next time reading proceeds.

The transmit buffer read pointer is updated when writing to the SPDR register, and not updated when reading from the transmit buffer. When reading from the transmit buffer, the value most recently written to the SPDR register is read. However, after generation of the transmit buffer empty interrupt, the values read from the transmit buffer are all 0 in the interval after completion of writing the number of frames of data specified in the number of frames specification bits (SPDCR.SPFC[1:0]) and before generation of the next buffer empty interrupt (while the SPSR.SPTEF flag is 0).

37.2.6 RSPI Sequence Control Register (SPSCR)

Address(es): RSPI0.SPSCR 000D 0108h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SPSLN[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SPSLN[2:0]	RSPI Sequence Length Specification	b2 b0 Sequence Length — Referenced SPCMD0 to SPCMD7 registers (No.) 0 0 0: 1 — 0→0→… 0 0 1: 2 — 0→1→0→… 0 1 0: 3 — 0→1→2→0→… 0 1 1: 4 — 0→1→2→3→0→… 1 0 0: 5 — 0→1→2→3→4→0→… 1 0 1: 6 — 0→1→2→3→4→5→0→… 1 1 0: 7 — 0→1→2→3→4→5→6→0→… 1 1 1: 8 — 0→1→2→3→4→5→6→7→0→… The order in which the SPCMD0 to SPCMD7 registers are to be referenced is changed according to the sequence length that is set in these bits. The relationship among the setting of these bits, sequence length, and the SPCMD0 to SPCMD7 registers referenced by the RSPI is shown above. However, the RSPI in slave mode references the SPCMD0 register.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SPSCR register sets the sequence length when the RSPI operates in master mode. When changing the SPSCR.SPSLN[2:0] bits while both the SPCR.MSTR and SPCR.SPE bits are 1, the bits should be changed while the SPSR.IDLNF flag is 0.

SPSLN[2:0] Bits (RSPI Sequence Length Specification)

The SPSLN[2:0] bits specify a sequence length when the RSPI in master mode performs sequential operations. The RSPI in master mode changes the SPCMD0 to SPCMD7 registers to be referenced and the order in which they are referenced according to the sequence length that is set in the SPSLN[2:0] bits.
In slave mode, the SPCMD0 register is referred.

37.2.7 RSPI Sequence Status Register (SPSSR)

Address(es): RSPI0.SPSSR 000D 0109h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	SPECM[2:0]			—	SPCP[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SPCP[2:0]	RSPI Command Pointer	b2 b0 0 0 0: SPCMD0 0 0 1: SPCMD1 0 1 0: SPCMD2 0 1 1: SPCMD3 1 0 0: SPCMD4 1 0 1: SPCMD5 1 1 0: SPCMD6 1 1 1: SPCMD7	R
b3	—	Reserved	This bit is read as 0.	R
b6 to b4	SPECM[2:0]	RSPI Error Command	b6 b4 0 0 0: SPCMD0 0 0 1: SPCMD1 0 1 0: SPCMD2 0 1 1: SPCMD3 1 0 0: SPCMD4 1 0 1: SPCMD5 1 1 0: SPCMD6 1 1 1: SPCMD7	R
b7	—	Reserved	This bit is read as 0.	R

The SPSSR register indicates the sequence control status when the RSPI operates in master mode.
Any writing to the SPSSR register is ignored.

SPCP[2:0] Bits (RSPI Command Pointer)

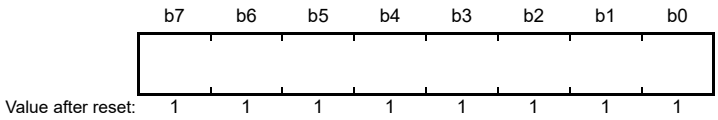
The SPCP[2:0] bits indicate the SPCMDm register that is currently pointed to by the pointer during sequence control by the RSPI.
For the RSPI's sequence control, refer to section 37.3.12.1, Master Mode Operation.

SPECM[2:0] Bits (RSPI Error Command)

The SPECM[2:0] bits indicate the SPCMDm register that is specified by the SPCP[2:0] bits when an error is detected during sequence control by the RSPI. The RSPI updates the SPECM[2:0] bits only when an error is detected. If both the SPSR.OVRF and SPSR.MODF flags are 0 and there is no error, the values of the SPECM[2:0] bits have no meaning.
For the RSPI's error detection function, refer to section 37.3.10, Error Detection. For the RSPI's sequence control, refer to section 37.3.12.1, Master Mode Operation.

37.2.8 RSPI Bit Rate Register (SPBR)

Address(es): RSPI0.SPBR 000D 010Ah



The SPBR register sets the bit rate in master mode. Do not change the SPBR register while both the SPCR.MSTR and SPCR.SPE bits are 1.

When the RSPI is used in slave mode, the bit rate depends on the bit rate of the input clock (bit rate satisfying the electrical characteristics should be used) regardless of the settings of the SPBR register and the SPCMDm.BRDV[1:0] bits (bit rate division setting bits).
The bit rate is determined by combinations of the SPBR register setting and the SPCMDm.BRDV[1:0] bit setting. The equation for calculating the bit rate is given below. In the equation, n denotes the SPBR register setting (0, 1, 2, …, 255), and N denotes the BRDV[1:0] bit setting (0, 1, 2, 3).

$$\text{Bit rate} = \frac{f(PCLK)}{2 \times (n+1) \times 2^{N}}$$

Table 37.3 lists examples of the relationship among the SPBR register settings, the BRDV[1:0] bit settings, and bit rates.
Use the bit rate that meets electrical characteristics based on the AC specifications of the target device.

Table 37.3 Relationship among SPBR Settings, BRDV[1:0] Settings, and Bit Rates

SPBR (n)	BRDV[1:0] Bits (N)	Division Ratio	Bit Rate							
			PCLK = 32 MHz	PCLK = 36 MHz	PCLK = 40 MHz	PCLK = 50 MHz	PCLK = 60 MHz	PCLK = 80 MHz	PCLK = 100 MHz	PCLK = 120 MHz
0	0	2	16.0 Mbps	18.0 Mbps	20.0 Mbps	25.0 Mbps	30.0 Mbps	40.0 Mbps	—	—
1	0	4	8.00 Mbps	9.00 Mbps	10.0 Mbps	12.5 Mbps	15.0 Mbps	20.0 Mbps	25.0 Mbps	30.0 Mbps
2	0	6	5.33 Mbps	6.00 Mbps	6.67 Mbps	8.33 Mbps	10.0 Mbps	13.3 Mbps	16.7 Mbps	20.0 Mbps
3	0	8	4.00 Mbps	4.50 Mbps	5.00 Mbps	6.25 Mbps	7.50 Mbps	10.0 Mbps	12.5 Mbps	15.0 Mbps
4	0	10	3.20 Mbps	3.60 Mbps	4.00 Mbps	5.00 Mbps	6.00 Mbps	8.00 Mbps	10.0 Mbps	12.0 Mbps
5	0	12	2.67 Mbps	3.00 Mbps	3.33 Mbps	4.16 Mbps	5.00 Mbps	6.67 Mbps	8.33 Mbps	10.0 Mbps
5	1	24	1.33 Mbps	1.50 Mbps	1.67 Mbps	2.08 Mbps	2.50 Mbps	3.33 Mbps	4.17 Mbps	5.00 Mbps
5	2	48	667 kbps	750 kbps	833 kbps	1.04 Mbps	1.25 Mbps	1.67 Mbps	2.08 Mbps	2.50 Mbps
5	3	96	333 kbps	375 kbps	417 kbps	521 kbps	625 kbps	833 kbps	1.04 Mbps	1.25 Mbps
255	3	4096	7.81 kbps	8.80 kbps	9.78 kbps	12.2 kbps	14.6 kbps	19.5 kbps	24.4 kbps	29.3 kbps

37.2.9 RSPI Data Control Register (SPDCR)

Address(es): RSPI0.SPDCR 000D 010Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	SPBYT	SPLW	SPRDTD	—	—	SPFC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	SPFC[1:0]	Number of Frames Specification	b1 b0 0 0: 1 frame 0 1: 2 frames 1 0: 3 frames 1 1: 4 frames	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	SPRDTD	RSPI Receive/Transmit Data Select	0: The SPDR values are read from the receive buffer 1: The SPDR values are read from the transmit buffer (but only if the transmit buffer is empty)	R/W
b5	SPLW	RSPI Longword Access/ Word Access Specification*1	0: The SPDR register is accessed in words 1: The SPDR register is accessed in longwords	R/W
b6	SPBYT	RSPI Byte Access Specification	0: The SPDR register is accessed in words or longwords (the SPLW bit is enabled) 1: The SPDR register is accessed in bytes (the SPLW bit is disabled)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. Set the SPBYT bit to 0, when accessing the SPDR register in words or longwords.

Up to four frames can be transmitted or received in one round of transmission or reception activation. The amount of data in each transfer is controlled by the combination of the SPCMDm.SPB[3:0] bits, the SPSCR.SPSLN[2:0] bits, and the SPDCR.SPFC[1:0] bits.
When changing the SPDCR.SPFC[1:0] bits while the SPCR.SPE bit is 1, the bits should be changed while the SPSR.IDLNF flag is 0.

SPFC[1:0] Bits (Number of Frames Specification)

The SPFC[1:0] bits specify the number of frames that can be stored in the SPDR register (per transfer activation). Up to four frames can be transmitted or received in one round of transmission or reception, and the amount of data is determined by the combination of the SPSCR.SPSLN[2:0] bits, and the SPDCR.SPFC[1:0] bits. Furthermore, the setting of the SPFC[1:0] bits adjusts the number of frames for generation of receive buffer full interrupt, and start of transmission or generation of transmit buffer empty interrupts.
When the number of frames of transmit data specified by SPFC[1:0] bits is written to the SPDR register, the SPSR.SPTEF flag becomes 0 and transmission starts. Then, when the specified number of frames of transmit data has been transferred to the shift register, the SPTEF flag becomes 1 and the RSPI transmit buffer empty interrupt is generated.
When the number of frames specified by the SPFC[1:0] bits are received, the SPSR.SPRF flag becomes 1 and the receive buffer full interrupt is generated.
Table 37.4 lists the frame configurations that can be stored in the SPDR register and examples of combinations of settings for transmission and reception. Do not select the combinations of settings other than those shown in the examples.

Table 37.4 Settable Combinations of SPSLN[2:0] Bits and SPFC[1:0] Bits

Setting	SPSLN[2:0]	SPFC[1:0]	Number of Frames in a Single Sequence	Number of Frames at which Transmit Buffer or Receive Buffer Status Becomes “Has Valid Data”
1-1	000b	00b	1	1
1-2	000b	01b	2	2
1-3	000b	10b	3	3
1-4	000b	11b	4	4
2-1	001b	01b	2	2
2-2	001b	11b	4	4
3	010b	10b	3	3
4	011b	11b	4	4
5	100b	00b	5	1
6	101b	00b	6	1
7	110b	00b	7	1
8	111b	00b	8	1

SPRDTD Bit (RSPI Receive/Transmit Data Select)

The SPRDTD bit selects whether the SPDR register reads values from the receive buffer or from the transmit buffer.
If reading is from the transmit buffer, the value written to the SPDR register immediately beforehand is read.
When reading the transmit buffer, do so before writing of the number of frames set in the SPFC[1:0] bits is finished and after generation of the transmit buffer empty interrupt (While the SPSR.SPTEF flag is 1).
For details, refer to section 37.2.5, RSPI Data Register (SPDR).

SPLW Bit (RSPI Longword Access/Word Access Specification)

The SPLW bit specifies the access width for the SPDR register. This bit setting is enabled when the SPBYT bit is 0.
Access to the SPDR register in words when the SPLW bit is 0 and in longwords when the SPLW bit is 1.
Also, when the SPLW bit is 0, set the SPCMDm.SPB[3:0] bits (RSPI data length setting bits) to 8 to 16 bits. Do not select 20, 24, or 32 bits.

SPBYT Bit (RSPI Byte Access Specification)

The SPBYT bit specifies the access width for the SPDR register. Access to the SPDR register according to the SPLW bit setting when the SPBYT bit is 0. Access to the SPDR register in bytes when the SPBYT bit is 1.
When the SPBYT bit is 1, set the SPCMDm.SPB[3:0] bits (RSPI data length setting bits) to 8 bits. Do not select 9 to 16, 20,24, or 32 bits.

37.2.10 RSPI Clock Delay Register (SPCKD)

Address(es): RSPI0.SPCKD 000D 010Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SCKDL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SCKDL[2:0]	RSPCK Delay Setting	b2 b0 0 0 0: 1 RSPCK 0 0 1: 2 RSPCK 0 1 0: 3 RSPCK 0 1 1: 4 RSPCK 1 0 0: 5 RSPCK 1 0 1: 6 RSPCK 1 1 0: 7 RSPCK 1 1 1: 8 RSPCK	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SPCKD register sets a period from the beginning of SSLAi signal assertion to RSPCK oscillation (RSPCK delay) when the SPCMDm.SCKDEN bit is 1. Do not change the SPCKD register while both the SPCR.MSTR and SPCR.SPE bits are 1.

SCKDL[2:0] Bits (RSPCK Delay Setting)

The SCKDL[2:0] bits set an RSPCK delay value when the SPCMDm.SCKDEN bit is 1.
When using the RSPI in slave mode, set the SCKDL[2:0] bits to 000b.

37.2.11 RSPI Slave Select Negation Delay Register (SSLND)

Address(es): RSPI0.SSLND 000D 010Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SLNDL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SLNDL[2:0]	SSL Negation Delay Setting	b2 b0 0 0 0: 1 RSPCK 0 0 1: 2 RSPCK 0 1 0: 3 RSPCK 0 1 1: 4 RSPCK 1 0 0: 5 RSPCK 1 0 1: 6 RSPCK 1 1 0: 7 RSPCK 1 1 1: 8 RSPCK	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SSLND register sets a period (SSL negation delay) from the transmission of a final RSPCK edge to the negation of the SSLAi signal during a serial transfer by the RSPI in master mode. Do not change the SSLND register while both the SPCR.MSTR and SPCR.SPE bits are 1.

SLNDL[2:0] Bits (SSL Negation Delay Setting)

The SLNDL[2:0] bits set an SSL negation delay value when the SPCMDm.SLNDEN bit is 1.
When using the RSPI in slave mode, set the SLNDL[2:0] bits to 000b.

37.2.12 RSPI Next-Access Delay Register (SPND)

Address(es): RSPI0.SPND 000D 010Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SPNDL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SPNDL[2:0]	RSPI Next-Access Delay Setting	b2 b0 0 0 0: 1 RSPCK + 2 PCLK 0 0 1: 2 RSPCK + 2 PCLK 0 1 0: 3 RSPCK + 2 PCLK 0 1 1: 4 RSPCK + 2 PCLK 1 0 0: 5 RSPCK + 2 PCLK 1 0 1: 6 RSPCK + 2 PCLK 1 1 0: 7 RSPCK + 2 PCLK 1 1 1: 8 RSPCK + 2 PCLK	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SPND sets a non-active period (next-access delay) of the SSLAi signal after termination of a serial transfer when the SPCMDm.SPNDEN bit is 1. Do not change the SPND register while both the SPCR.MSTR and SPCR.SPE bits are 1.

SPNDL[2:0] Bits (RSPI Next-Access Delay Setting)

The SPNDL[2:0] bits set a next-access delay when the SPCMDm.SPNDEN bit is 1.
When using the RSPI in slave mode, set the SPNDL[2:0] bits to 000b.

37.2.13 RSPI Control Register 2 (SPCR2)

Address(es): RSPI0.SPCR2 000D 010Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	SCKASE	PTE	SPIIE	SPOE	SPPE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SPPE	Parity Enable*1	0: Does not add the parity bit to transmit data and does not check the parity bit of receive data 1: Adds the parity bit to transmit data and checks the parity bit of receive data	R/W
b1	SPOE	Parity Mode*1	0: Selects even parity for use in transmission and reception 1: Selects odd parity for use in transmission and reception	R/W
b2	SPIIE	Idle Interrupt Enable	0: Disables the generation of idle interrupt requests 1: Enables the generation of idle interrupt requests	R/W
b3	PTE	Parity Self-Diagnosis	0: Disables the self-diagnosis function of the parity circuit 1: Enables the self-diagnosis function of the parity circuit	R/W
b4	SCKASE	RSPCK Auto-Stop Function Enable*1	0: Disables the RSPCK auto-stop function 1: Enables the RSPCK auto-stop function	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Do not change the SPPE, SPOE, and SCKASE bits while the SPCR.SPE bit is 1.

SPPE Bit (Parity Enable)

The SPPE bit enables or disables the parity function.

SPOE Bit (Parity Mode)

The SPOE bit specifies odd or even parity.

When even parity is set, parity bit addition is performed so that the total number of 1-bits in the transmit/receive character plus the parity bit is even. Similarly, when odd parity is set, parity bit addition is performed so that the total number of 1-bits in the transmit/receive character plus the parity bit is odd.

The SPOE bit is valid only when the SPPE bit is 1.

SPIIE Bit (Idle Interrupt Enable)

The SPIIE bit enables or disables the generation of idle interrupt requests when the RSPI being in the idle state is detected and the SPSR.IDLNF flag is set to 0.

PTE Bit (Parity Self-Diagnosis)

The PTE bit enables the self-diagnosis function of the parity circuit in order to check whether the parity function is operating correctly.

SCKASE Bit (RSPCK Auto-Stop Function Enable)

The SCKASE bit enables or disables the RSPCK auto-stop function. When this function is enabled, the RSPCK clock is stopped before an overrun error occurs when data is received in master mode. For details, refer to section 37.3.10.1, Overrun Error.

37.2.14 RSPI Command Register m (SPCMDm) (m = 0 to 7)

Address(es): RSPI0.SPCMD0 000D 0110h, RSPI0.SPCMD1 000D 0112h, RSPI0.SPCMD2 000D 0114h, RSPI0.SPCMD3 000D 0116h, RSPI0.SPCMD4 000D 0118h, RSPI0.SPCMD5 000D 011Ah, RSPI0.SPCMD6 000D 011Ch, RSPI0.SPCMD7 000D 011Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]				SSLKP	SSLA[2:0]			BRDV[1:0]		CPOL	CPHA
Value after reset:	0	0	0	0	0	1	1	1	0	0	0	0	1	1	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	CPHA	RSPCK Phase Setting	0: Data sampling on odd edge, data variation on even edge 1: Data variation on odd edge, data sampling on even edge	R/W
b1	CPOL	RSPCK Polarity Setting	0: RSPCK is low when idle 1: RSPCK is high when idle	R/W
b3, b2	BRDV[1:0]	Bit Rate Division Setting	b3 b2 0 0: These bits select the base bit rate 0 1: These bits select the base bit rate divided by 2 1 0: These bits select the base bit rate divided by 4 1 1: These bits select the base bit rate divided by 8	R/W
b6 to b4	SSLA[2:0]	SSL Signal Assertion Setting	b6 b4 0 0 0: SSL0 0 0 1: SSL1 0 1 0: SSL2 0 1 1: SSL3 1 x x: Setting prohibited	R/W
b7	SSLKP	SSL Signal Level Keeping	0: Negates all SSL signals upon completion of transfer 1: Keeps the SSL signal level from the end of transfer until the beginning of the next access (burst transfer)	R/W
b11 to b8	SPB[3:0]	RSPI Data Length Setting	b11 b8 0100 to 0111: 8 bits 1 0 0 0: 9 bits 1 0 0 1: 10 bits 1 0 1 0: 11 bits 1 0 1 1: 12 bits 1 1 0 0: 13 bits 1 1 0 1: 14 bits 1 1 1 0: 15 bits 1 1 1 1: 16 bits 0 0 0 0: 20 bits 0 0 0 1: 24 bits 0010, 0011: 32 bits	R/W
b12	LSBF	RSPI LSB First	0: MSB first 1: LSB first	R/W
b13	SPNDEN	RSPI Next-Access Delay Enable	0: A next-access delay of 1 RSPCK + 2 PCLK 1: A next-access delay is equal to the setting of the RSPI next-access delay register (SPND)	R/W
b14	SLNDEN	SSL Negation Delay Setting Enable	0: An SSL negation delay of 1 RSPCK 1: An SSL negation delay is equal to the setting of the RSPI slave select negation delay register (SSLND)	R/W
b15	SCKDEN	RSPCK Delay Setting Enable	0: An RSPCK delay of 1 RSPCK 1: An RSPCK delay is equal to the setting of the RSPI clock delay register (SPCKD)	R/W

x: Don’t care

The SPCMDm register is used to set a transfer format for the RSPI in master mode. Each channel has eight RSPI command registers (SPCMD0 to SPCMD7). Some of the bits in the SPCMD0 register is used to set a transfer mode for the RSPI in slave mode. The RSPI in master mode sequentially references the SPCMDm register according to the settings in the SPSCR.SPSLN[2:0] bits, and executes the serial transfer that is set in the referenced SPCMDm register. SPCMDm register should be set while the transmit buffer is empty (data for the next transfer is not set) and before setting of the data that is to be transmitted when that SPCMDm register is referenced.
An SPCMDm register that is referenced by the RSPI in master mode can be checked by means of the SPSSR.SPCP[2:0] bits. Do not change the SPCMDm register while the SPCR.MSTR bit is 0 and the SPCR.SPE bit is 1.

CPHA Bit (RSPCK Phase Setting)

The CPHA bit sets an RSPCK phase of the RSPI in master mode or slave mode. Data communications between RSPI modules require the same RSPCK phase setting between the modules.

CPOL Bit (RSPCK Polarity Setting)

The CPOL bit sets an RSPCK polarity of the RSPI in master mode or slave mode. Data communications between RSPI modules require the same RSPCK polarity setting between the modules.

BRDV[1:0] Bits (Bit Rate Division Setting)

The BRDV[1:0] bits are used to determine the bit rate. A bit rate is determined by combinations of the settings in the BRDV[1:0] bits and the SPBR register (refer to section 37.2.8, RSPI Bit Rate Register (SPBR)). The settings in the SPBR register determine the base bit rate. The settings in the BRDV[1:0] bits are used to select a bit rate which is obtained by dividing the base bit rate by 1, 2, 4, or 8. In the SPCMDm register, different BRDV[1:0] bit settings can be specified. This enables execution of serial transfers at a different bit rate for each command.

SSLA[2:0] Bits (SSL Signal Assertion Setting)

The SSLA[2:0] bits control the SSLAi signal assertion when the RSPI performs serial transfers in master mode.
Setting the SSLA[2:0] bits controls the assertion for the SSLAi signal. When an SSLAi signal is asserted, its polarity is determined by the set value in the corresponding SSLP. When the SSLA[2:0] bits are set to 000b in multi-master mode, serial transfers are performed with all the SSL signals in the negated state (as the SSLA0 pin acts as input).
When using the RSPI in slave mode, set the SSLA[2:0] bits to 000b.

SSLKP Bit (SSL Signal Level Keeping)

When the RSPI in master mode performs a serial transfer, the SSLKP bit specifies whether the SSLAi 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.
Setting the SSLKP bit to 1 enables a burst transfer. For details, refer to section 37.3.12.1, Master Mode Operation (4) Burst Transfer.
When using the RSPI in slave mode, the SSLKP bit should be set to 0.

SPB[3:0] Bits (RSPI Data Length Setting)

The SPB[3:0] bits set a transfer data length for the RSPI in master mode or slave mode. When the SPDCR.SPBYT is 1, set the SPB[3:0] bits to 0100b (8 bits). When the SPBYT bit is 0 and the SPDCR.SPLW bit is 0, set the SPB[3:0] bits to 0100b (8 bits) to 1111b (16 bits).

LSBF Bit (RSPI LSB First)

The LSBF bit sets the data format of the RSPI in master mode or slave mode to MSB first or LSB first.

SPNDEN Bit (RSPI Next-Access Delay Enable)

The SPNDEN bit sets the period from the time the RSPI in master mode terminates a serial transfer and sets the SSLAi signal inactive until the RSPI enables the SSLAi 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 + 2 PCLK. If the SPNDEN bit is 1, the RSPI inserts a next-access delay in compliance with the SPND setting.

When using the RSPI in slave mode, the SPNDEN bit should be set to 0.

SLNDEN Bit (SSL Negation Delay Setting Enable)

The SLNDEN bit sets the period from the time the RSPI in master mode stops RSPCK oscillation until the RSPI sets the SSLAi signal inactive (SSL negation delay). 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 the SSLND setting.

When using the RSPI in slave mode, the SLNDEN bit should be set to 0.

SCKDEN Bit (RSPCK Delay Setting Enable)

The SCKDEN bit sets the period from the point when the RSPI in master mode activates the SSLAi signal until the RSPCK starts oscillation (RSPI clock 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 the SPCKD setting.

When using the RSPI in slave mode, the SCKDEN bit should be set to 0.

37.2.15 RSPI Data Control Register 2 (SPDCR2)

Address(es): RSPI0.SPDCR2 000D 0120h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	DINV	BYSW
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BYSW	Byte Swap	0: Byte swapping of SPDR data disabled 1: Byte swapping of SPDR data enabled	R/W
b1	DINV	Transfer Data Invert	0: Data bits in the transmit buffer are transferred to the shift register as they are. Data bits in the shift register are transferred to the receive buffer as they are. 1: Data bits in the transmit buffer are transferred to the shift register with inverting. Data bits in the shift register are transferred to the receive buffer with inverting.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to enable or disable byte swapping and logic inverting of transmit and receive data.
The SPCR.SPE bit should be 0 when rewriting this register.

BYSW Bit (Byte Swap)

On transmit, this bit specifies that data bytes written in the SPDR register will be swapped before being transmitted. On receive, this bit specifies that received bytes will be swapped before the data is transferred to the SPDR register. This bit setting is enabled when the SPDCR.SPBYT bit is 0.
When using byte swap, set the SPCMD.SPB[3:0] bits to 1111b (16 bits), 0010b (32 bits), or 0011b (32 bits). Also, set the SPCR2.SPPE bit to 0 (parity bit not added). For details, refer to sections 37.3.4.3 Byte Swap Transmission and 37.3.4.4 Byte Swap Reception.

DINV Bit (Transfer Data Invert)

This bit is used to invert the logic levels of the data bits when the data is transferred from the transmit buffer to the shift register and from the shift register to the receive buffer.
The parity bit is added for the data in the transmit shift register and the parity is checked for the data in the receive shift register.

37.2.16 RSPI Control Register 3 (SPCR3)

Address(es): RSPI0.SPCR3 000D 0121h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	SPCIE	—	—	SCKDDIS	RXMD
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RXMD	Receive Operating Mode Setting	0: Full-duplex or transmit-only simplex communications (enables the transmitter) 1: Receive-only simplex communications (disables the transmitter)	R/W
b1	SCKDDIS	RSPCK Delay Between Data Byte Disable	0: Inserts delays between data bytes during burst transfer 1: Does not insert delays between data bytes during burst transfer	R/W
b3 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b4	SPCIE	Communication End Interrupt Enable	0: Disables the generation of communication end interrupt requests 1: Enables the generation of communication end interrupt requests	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Do not change the RXMD and SCKDDIS bits while the SPCR.SPE bit is 1.

This register is used to set the operating mode of the RSPI.

RXMD Bit (Receive Operating Mode Setting)

This bit is used to stop the transmitter and operate only the receiver. This bit is valid only in slave mode.
When this bit is 1, the setting of the SPCR.TXMD bit is ignored.

SCKDDIS Bit (RSPCK Delay Between Data Byte Disable)

This bit is used to select whether the RSPCK delay between data bytes is to be inserted or not during burst transfer. This bit is valid only when the SPCR.MSTR bit is 1 (master mode) and the SPCMDm.SSLKP bit is 1. Set this bit to 0 in slave mode.

SPCIE Bit (Communication End Interrupt Enable)

This bit is used to enable or disable the generation of the communication end interrupt request.

37.3 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.

37.3.1 Overview of RSPI Operations

The RSPI is capable of synchronous serial transfers in slave mode (SPI operation), single-master mode (SPI operation), multi-master mode (SPI operation), slave mode (clock synchronous operation), and master mode (clock synchronous operation). A particular mode of the RSPI can be selected by using the MSTR, MODFEN, and SPMS bits in SPCR. Table 37.5 lists the relationship between RSPI modes and SPCR settings, and a description of each mode.

Table 37.5 Relationship between RSPI Modes and SPCR Settings and Description of Each Mode

	SPI Operation			Clock Synchronous Operation	
Mode	**Slave**	**Single-Master**	**Multi-Master**	**Slave**	**Master**
MSTR bit setting	0	1	1	0	1
MODFEN bit setting	0 or 1	0	1	0	0
SPMS bit setting	0	0	0	1	1
RSPCKA signal	Input	Output	Output/Hi-Z*1	Input	Output
MOSIA signal	Input	Output	Output/Hi-Z*1	Input	Output
MISOA signal	Output/Hi-Z*2	Input	Input	Output	Input
SSLA0 signal	Input	Output	Input	Hi-Z*3	Hi-Z*3
SSLA1 to SSLA3 signals	Hi-Z*3	Output	Output/Hi-Z*1	Hi-Z*3	Hi-Z*3
SSL polarity change function	Supported	Supported	Supported	—	—
Transfer rate	Up to PCLK/4	Up to PCLK/2	Up to PCLK/2	Up to PCLK/4	Up to PCLK/2
Clock source	RSPCK input	On-chip baud rate generator	On-chip baud rate generator	RSPCK input	On-chip baud rate generator
Clock polarity	Two				
Clock phase	Two	Two	Two	One (CPHA = 1)	Two
First transfer bit	MSB/LSB				
Transfer data length	8 to 16, 20, 24, 32 bits				
Burst transfer	Possible (CPHA = 1)	Possible (CPHA = 0,1)	Possible (CPHA = 0,1)	—	—
RSPCK delay control	Not supported	Supported	Supported	Not supported	Supported
SSL negation delay control	Not supported	Supported	Supported	Not supported	Supported
Next-access delay control	Not supported	Supported	Supported	Not supported	Supported
Transfer activation method	SSL input active or RSPCK oscillation	Transmit buffer is written to when a transmit buffer empty interrupt request is generated or the SPTEF flag is 1	Transmit buffer is written to when a transmit buffer empty interrupt request is generated or the SPTEF flag is 1	RSPCK oscillation	Transmit buffer is written to when a transmit buffer empty interrupt request is generated or the SPTEF flag is 1
Sequence control	Not supported	Supported	Supported	Not supported	Supported
Transmit buffer empty detection	Supported*7	Supported	Supported	Supported*7	Supported
Receive buffer full detection	Supported*4				
Overrun error detection	Supported*4	Supported*4, *6	Supported*4, *6	Supported*4	Supported*4, *6
Underrun error detection	Supported*7	Not supported	Not supported	Supported*7	Not supported
Parity error detection	Supported*4, *5				
Mode fault error detection	Supported (MODFEN = 1)	Not supported	Supported	Not supported	Not supported

Note 1. When SSLA0 is asserted by another master device, the pin becomes Hi-Z.
Note 2. When SSLA0 is negated or the SPCR.SPE bit is 0, the pin becomes Hi-Z.
Note 3. This function is not supported in this mode.
Note 4. When the SPCR.TXMD bit is 1, the detections of receiver buffer full, overrun error, and parity error are not performed.
Note 5. When the SPCR2.SPPE bit is 0, the detection of parity error is not performed.

Note 6. When the SPCR2.SCKASE bit is 1, the detection of overrun error is not performed.
Note 7. When the SPCR3.RXMD bit is 1, the detections of transmit buffer empty and underrun error are not performed.

37.3.2 Controlling RSPI Pins

The RSPI in single-master mode (SPI operation) or multi-master mode (SPI operation) 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 listed in Table 37.6.

Table 37.6 MOSI Signal Value Determination during SSL Negation Period

MOIFE Bit	MOIFV Bit	MOSIA Signal Value during SSL Negation Period
0	0, 1	Final data from previous transfer
1	0	Low
1	1	High

37.3.3 RSPI System Configuration Examples

37.3.3.1 Single Master/Single Slave (with This MCU Acting as Master)

Figure 37.5 shows a single-master/single-slave RSPI system configuration example when this MCU is used as a master. In the single-master/single-slave configuration, the SSLA0 to SSLA3 output of this MCU (master) are not used. The SSL input of the SPI slave is fixed to the low level, and the SPI slave is maintained in a select state.*1
This MCU (master) drives the RSPCKA and MOSIA. The SPI slave drives the MISO.

Note 1. In the transfer format corresponding to the case where the SPCMDm.CPHA bit 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 SSLAi output of this MCU should be connected to the SSL input of the slave device.

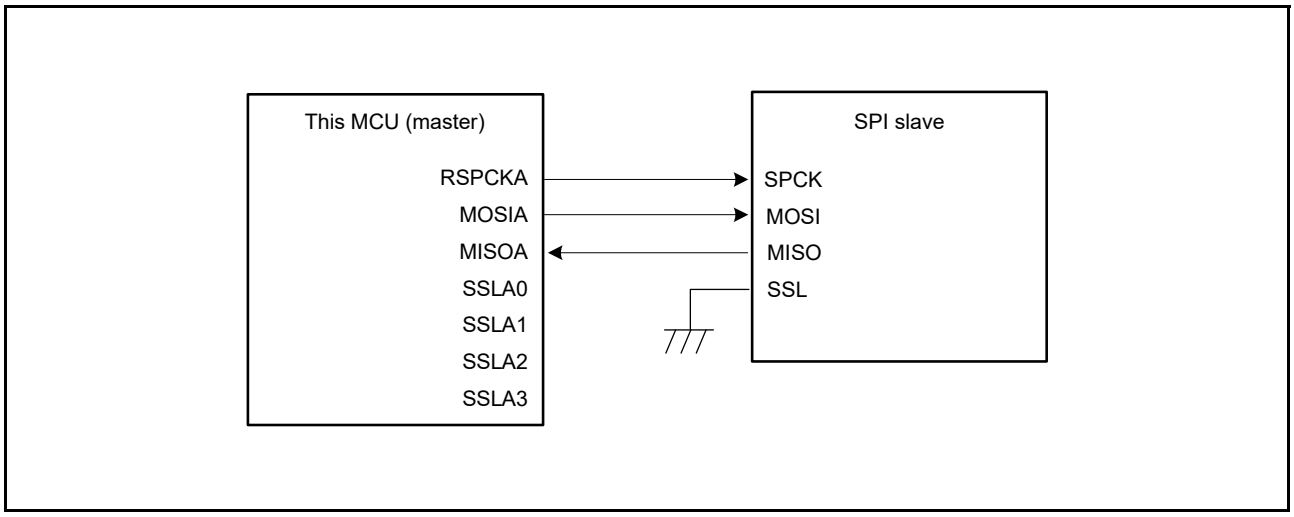


Figure 37.5 Single-Master/Single-Slave Configuration Example (This MCU = Master)

37.3.3.2 Single Master/Single Slave (with This MCU Acting as Slave)

Figure 37.6 shows a single-master/single-slave RSPI system configuration example when this MCU is used as a slave. When this MCU is to operate as a slave, the SSLA0 pin is used as SSL input. The SPI master drives the RSPCK and MOSI. This MCU (slave) drives the MISOA.*1

In the single-slave configuration in which the SPCMDm.CPHA bit is set to 1, the SSLA0 input of this MCU (slave) is fixed to the low level, this MCU (slave) is maintained in a select state, and in this manner it is possible to execute serial transfer (Figure 37.7). In this configuration, the communication end interrupt and the communication end event are not generated.

Note 1. When SSLA0 is at the non-active level, the pin state is Hi-Z.

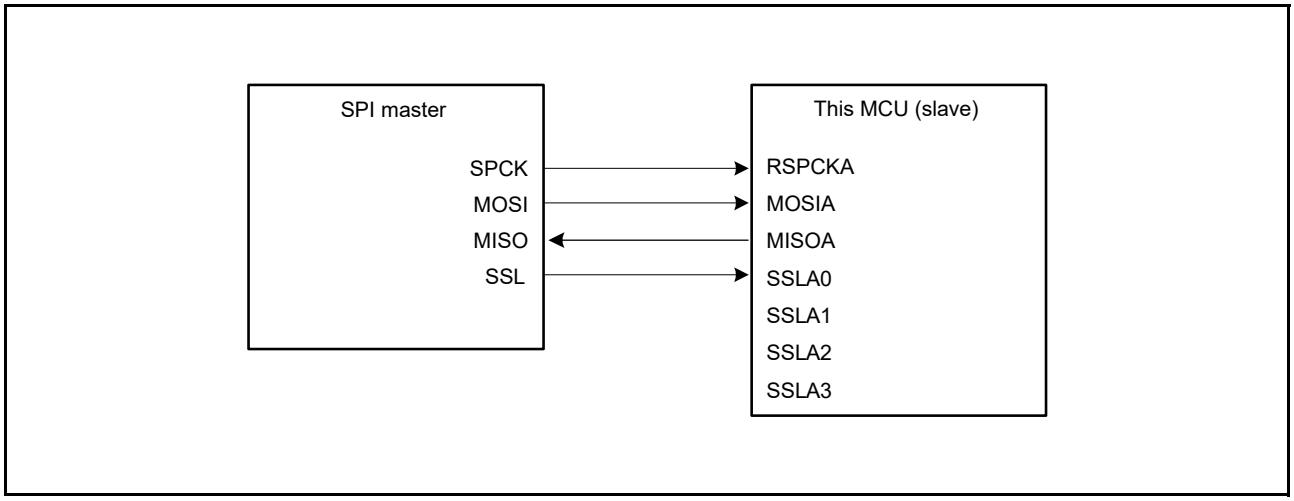


Figure 37.6 Single-Master/Single-Slave Configuration Example (This MCU = Slave, CPHA = 0)

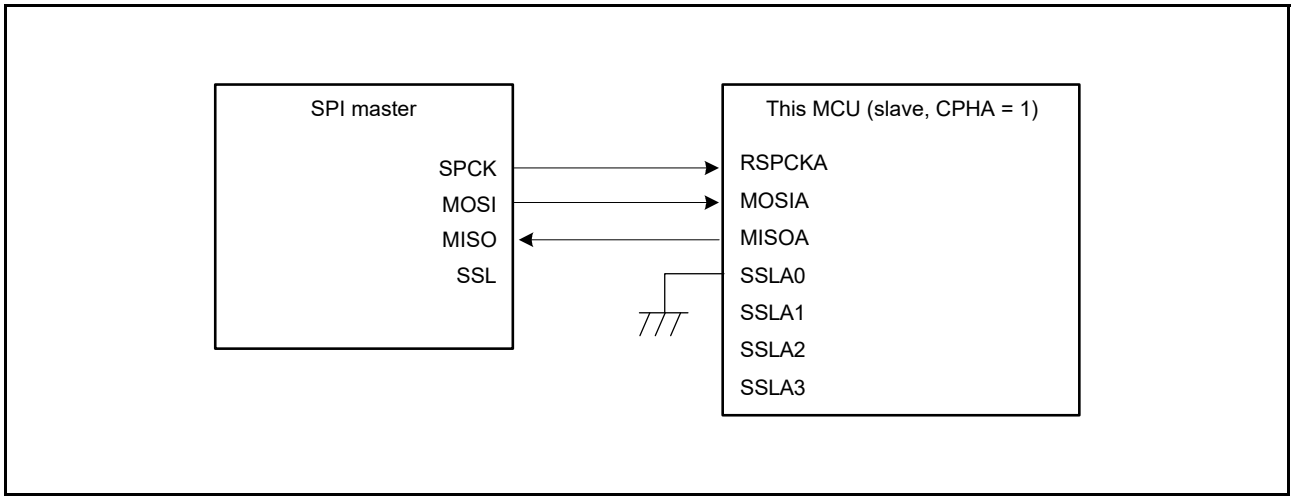


Figure 37.7 Single-Master/Single-Slave Configuration Example (This MCU = Slave, CPHA = 1)

37.3.3.3 Single Master/Multi-Slave (with This MCU Acting as Master)

Figure 37.8 shows a single-master/multi-slave RSPI system configuration example when this MCU is used as a master. In the example of Figure 37.8, the RSPI system is comprised of this MCU (master) and four slaves (SPI slave 0 to SPI slave 3).

The RSPCKA and MOSIA outputs of this MCU (master) are connected to the RSPCK and MOSI inputs of SPI slave 0 to SPI slave 3. The MISO outputs of SPI slave 0 to SPI slave 3 are all connected to the MISOA input of this MCU (master). SSLA0 to SSLA3 outputs of this MCU (master) are connected to the SSL inputs of SPI slave 0 to SPI slave 3, respectively.

This MCU (master) drives RSPCKA, MOSIA, and SSLA0 to SSLA3. Of the SPI slave 0 to SPI slave 3, the slave that receives low-level input into the SSL input drives MISO.

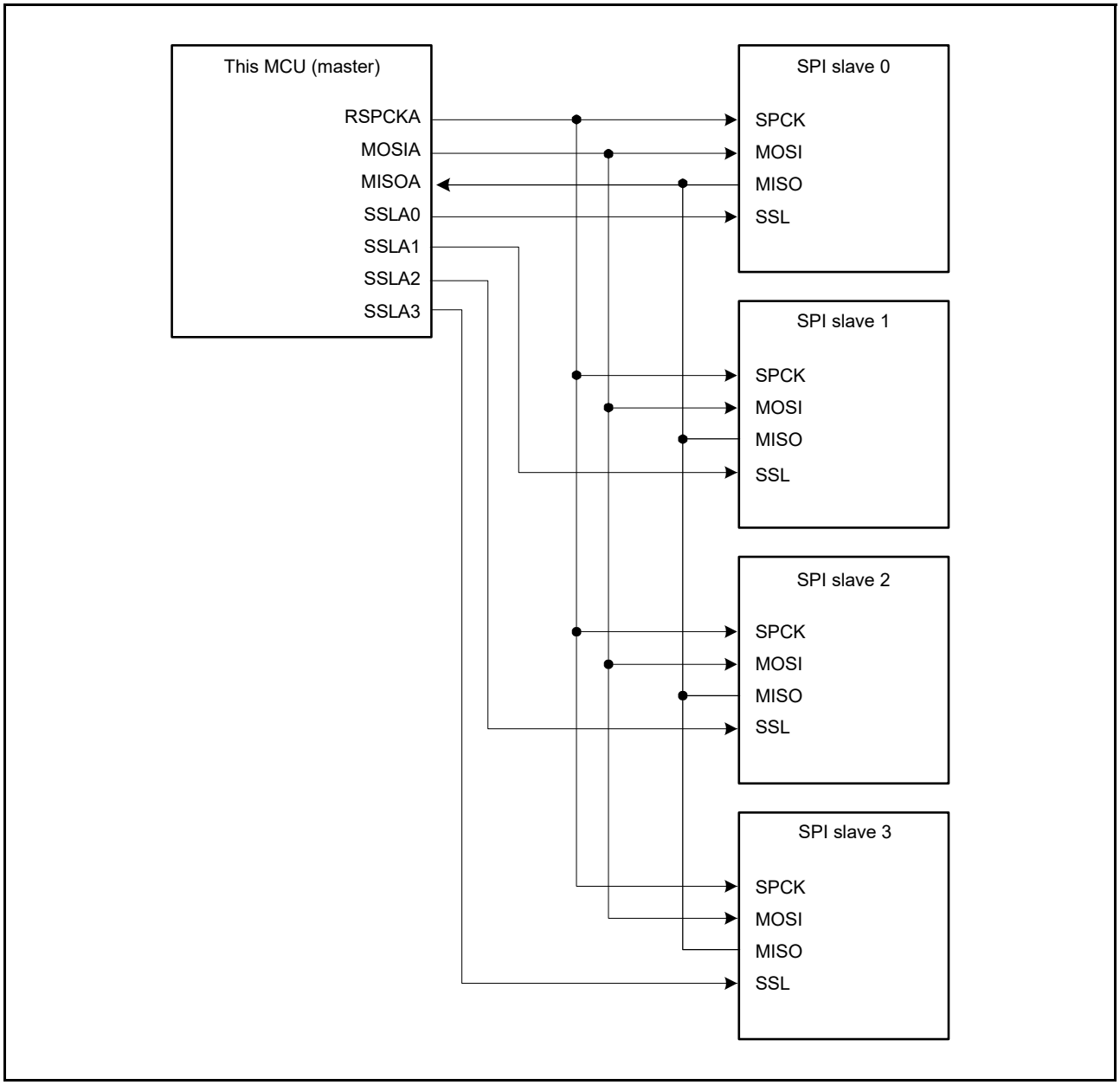


Figure 37.8 Single-Master/Multi-Slave Configuration Example (This MCU = Master)

37.3.3.4 Single Master/Multi-Slave (with This MCU Acting as Slave)

Figure 37.9 shows a single-master/multi-slave RSPI system configuration example when this MCU is used as a slave. In the example of Figure 37.9, the RSPI system is comprised of an SPI master and two MCUs (slave X and slave Y).
The SPCK and MOSI outputs of the SPI master are connected to the RSPCKA and MOSIA inputs of the MCUs (slave X and slave Y). The MISOA outputs of the MCUs (slave X and slave Y) are all connected to the MISO input of the SPI master. SSLX and SSLY outputs of the SPI master are connected to the SSLA0 inputs of the MCUs (slave X and slave Y), respectively.
The SPI master drives SPCK, MOSI, SSLX, and SSLY. Of the MCUs (slave X and slave Y), the slave that receives low-level input into the SSLA0 input drives MISOA.

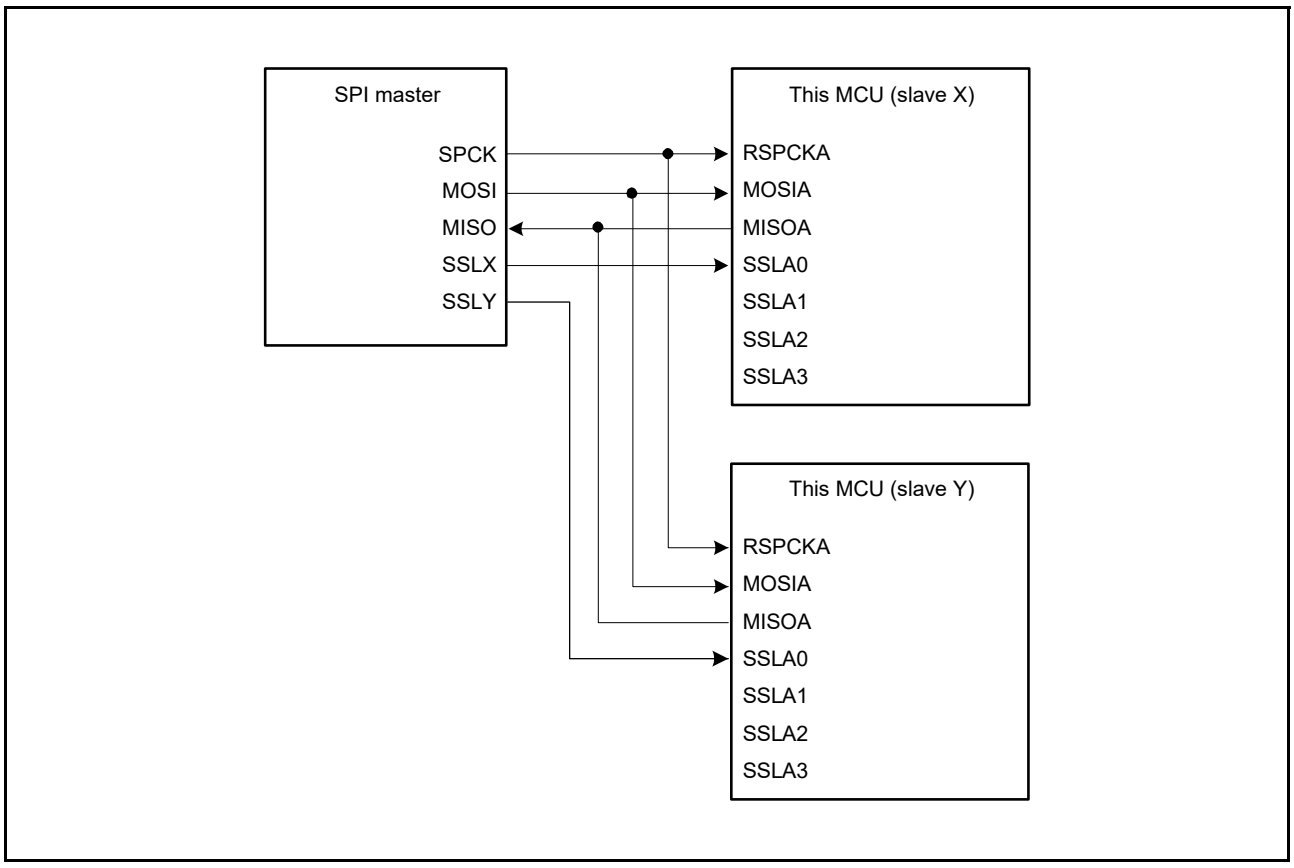


Figure 37.9 Single-Master/Multi-Slave Configuration Example (This MCU = Slave)

37.3.3.5 Multi-Master/Multi-Slave (with This MCU Acting as Master)

Figure 37.10 shows a multi-master/multi-slave RSPI system configuration example when this MCU is used as a master. In the example of Figure 37.10, the RSPI system is comprised of two MCUs (master X and master Y) and two SPI slaves (SPI slave 1 and SPI slave 2).

The RSPCKA and MOSIA outputs of the MCUs (master X and master Y) are connected to the RSPCK and MOSI inputs of SPI slaves 1 and 2. The MISO outputs of SPI slaves 1 and 2 are connected to the MISOA inputs of the MCUs (master X and master Y). Any generic port Y output from this MCU (master X) is connected to the SSLA0 input of this MCU (master Y). Any generic port X output of this MCU (master Y) is connected to the SSLA0 input of this MCU (master X). The SSLA1 and SSLA2 outputs of the MCUs (master X and master Y) are connected to the SSL inputs of the SPI slaves 1 and 2. In this configuration example, since the system can be comprised solely of SSLA0 input, and SSLA1 and SSLA2 outputs for slave connections, the SSLA3 output of this MCU is not required.

This MCU drives RSPCKA, MOSIA, SSLA1, and SSLA2 when the SSLA0 input level is high. When the SSLA0 input level is low, this MCU detects a mode fault error, sets RSPCKA, MOSIA, SSLA1, and SSLA2 to Hi-Z, and releases the RSPI bus right to the other master. Of the SPI slaves 1 and 2, the slave that receives low-level input into the SSL input drives MISO.

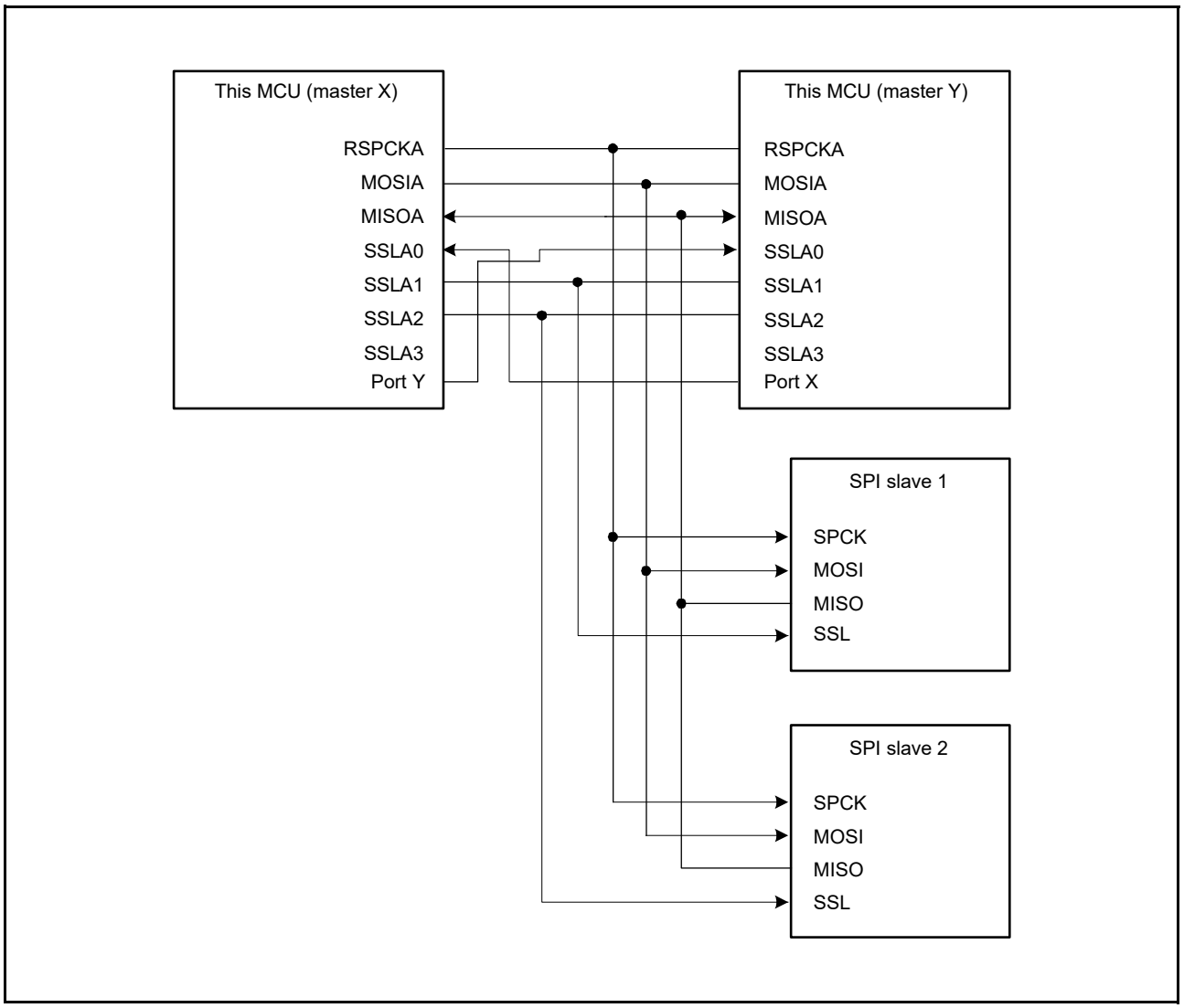


Figure 37.10 Multi-Master/Multi-Slave Configuration Example (This MCU = Master)

37.3.3.6 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (with This MCU Acting as Master)

Figure 37.11 shows a master (clock synchronous operation)/slave (clock synchronous operation) RSPI system configuration example when this MCU is used as a master. In the master (clock synchronous operation)/slave (clock synchronous operation) configuration, SSLA0 to SSLA3 of this MCU (master) are not used.
This MCU (master) drives the RSPCKA and MOSIA. The SPI slave drives the MISO.

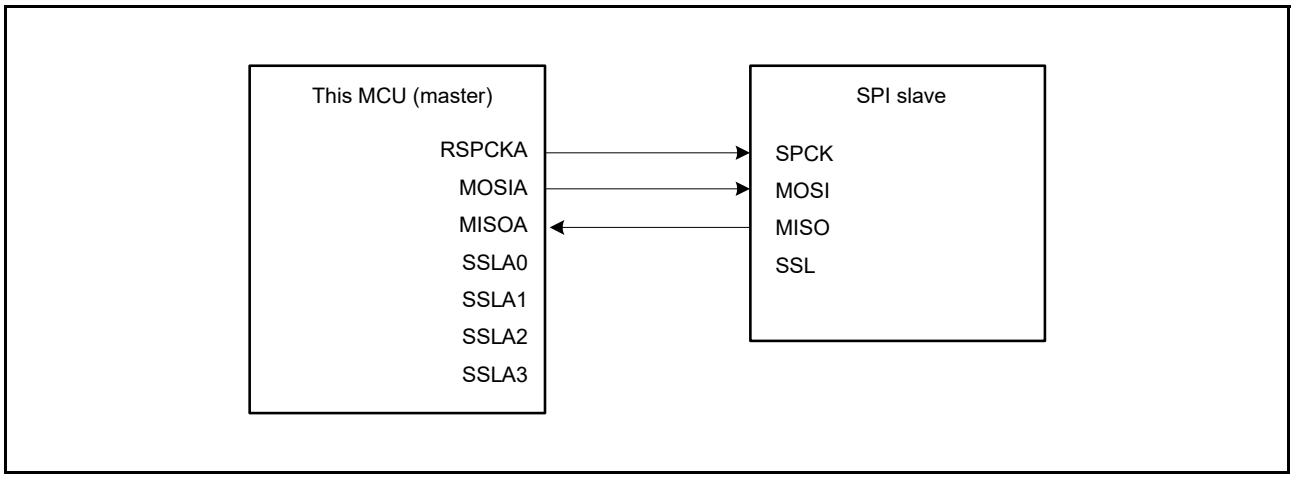


Figure 37.11 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) Configuration Example (This MCU = Master)

37.3.3.7 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (with This MCU Acting as Slave)

Figure 37.12 shows a master (clock synchronous operation)/slave (clock synchronous operation) RSPI system configuration example when this MCU is used as a slave. When this MCU is to operate as a slave (clock synchronous operation), this MCU (slave) drives the MISOA and the SPI master drives the SPCK and MOSI. In addition, SSLA0 to SSLA3 of this MCU (slave) are not used.
Only in the single-slave configuration in which the SPCMDm.CPHA bit is set to 1, this MCU (slave) can execute serial transfer.

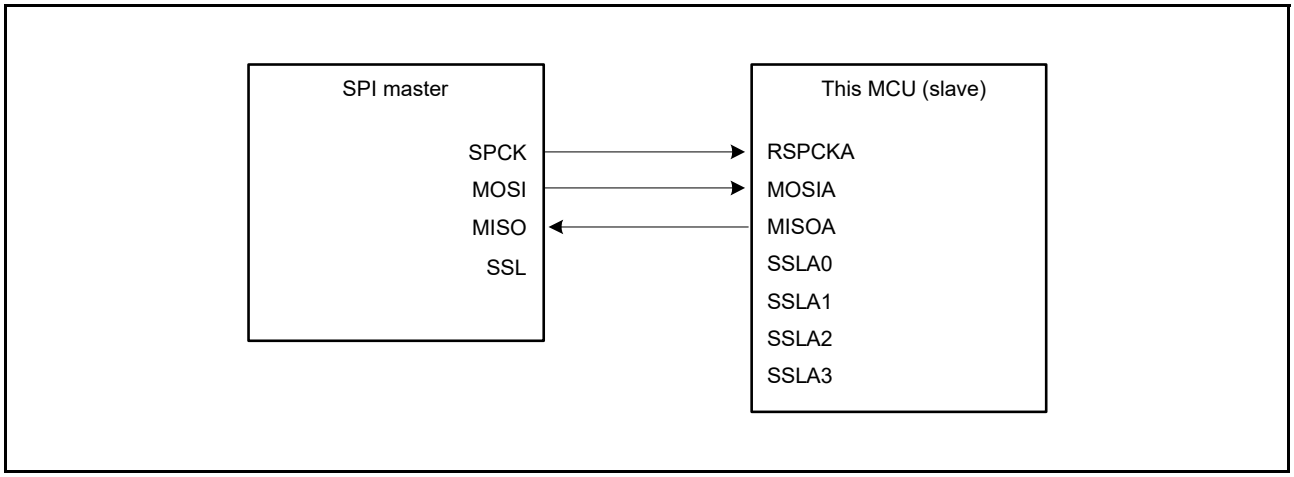


Figure 37.12 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) Configuration Example (This MCU = Slave, CPHA = 1)

37.3.4 Data Format

The RSPI's data format depends on the settings in RSPI command register m (SPCMDm) (m = 0 to 7) and the parity enable bit in RSPI control register 2 (SPCR2.SPPE) and RSPI data control register 2 (SPDCR2). Regardless of whether the MSB or LSB is first, the RSPI treats the range from the LSB bit in the RSPI data register (SPDR) to the selected data length as transfer data.
The format of one frame of data before or after transfer is shown below.

(a) With Parity Disabled

When parity is disabled, transmission or reception of data proceeds with the length in bits selected in the RSPI data length setting bits in RSPI command register m (SPCMDm.SPB[3:0]).

(b) With Parity Enabled

When parity is enabled, transmission or reception of data proceeds with the length in bits selected in the RSPI data length setting bits in RSPI command register m (SPCMDm.SPB[3:0]). In this case, however, the last bit is a parity bit.

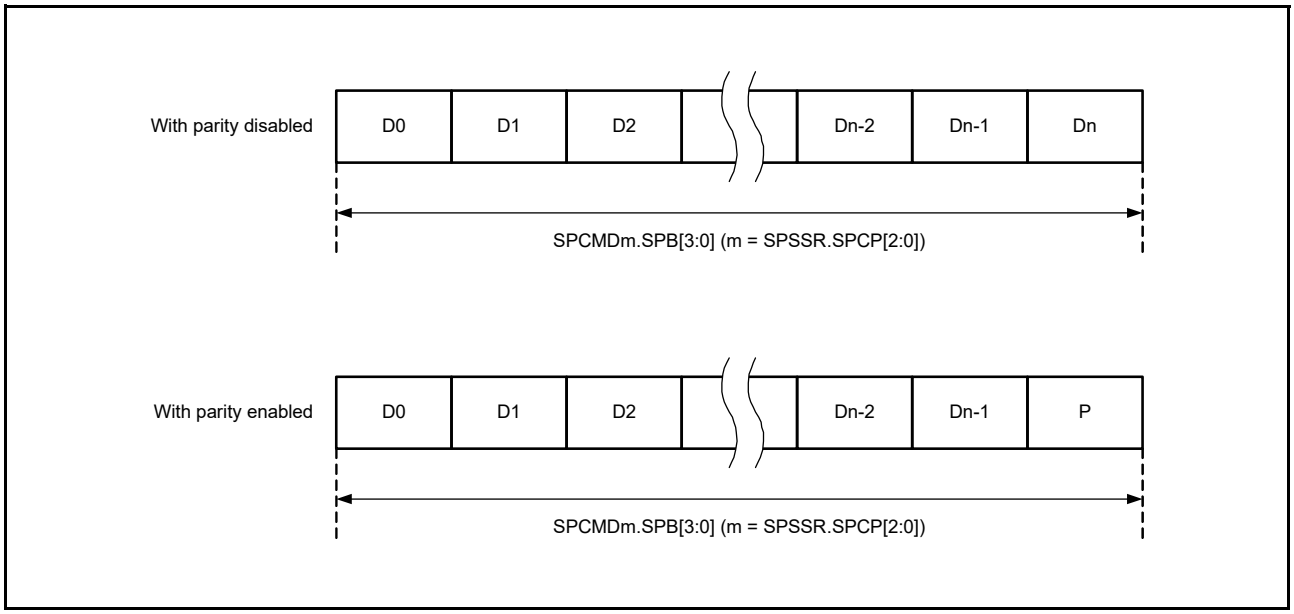


Figure 37.13 Outline of the Data Format (with Parity Disabled/Enabled)

37.3.4.1 When Parity is Disabled (SPCR2.SPPE = 0)

When parity is disabled, data for transmission are copied to the shift register with no prior processing. A description of the connection between the RSPI data register (SPDR) and the shift register in terms of the combination of MSB or LSB first and data length is given below.

(1) MSB First Transfer (32-Bit Data)

Figure 37.14 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity disabled, an RSPI data length of 32 bits, and MSB first selected.
In transmission, bits T31 to T00 from the current stage of the transmit buffer are copied to the shift register. Data for transmission are shifted out from the shift register in order from T31, through T30, and so on to T00.
In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R31 to R00 have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer.

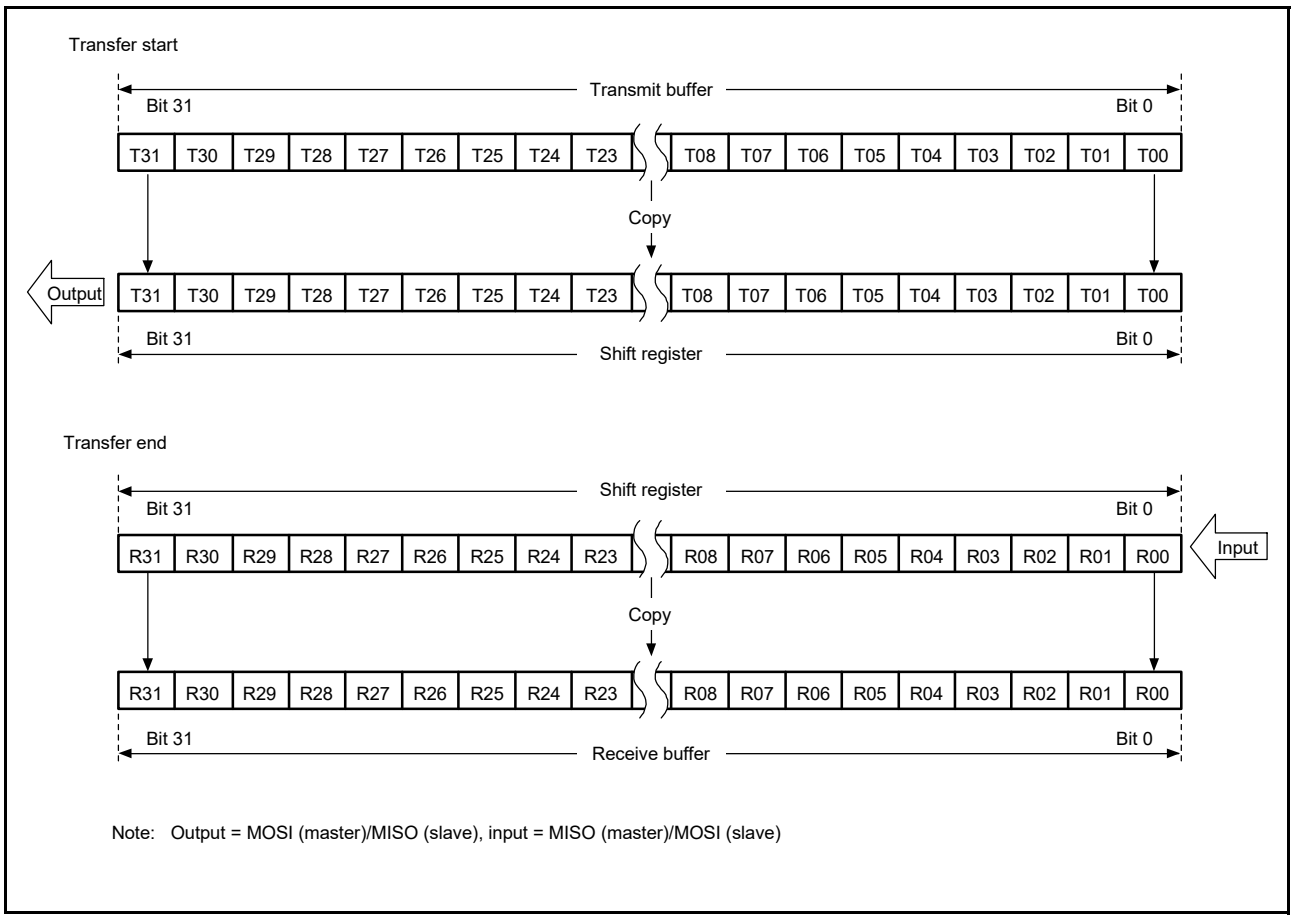


Figure 37.14 MSB First Transfer (32-Bit Data, Parity Disabled)

(2) MSB First Transfer (24-Bit Data)

Figure 37.15 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity disabled, 24 bits as the RSPI data length for an example that is not 32 bits, and MSB first selected.
In transmission, the lower 24 bits (T23 to T00) from the current stage of the transmit buffer are copied to the shift register. Data for transmission are shifted out from the shift register in order from T23, through T22, and so on to T00.
In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R23 to R00 have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. At this time, if the SPCR3.RXMD bit is 0, the upper 8 bits of the transmit buffer are stored in the upper 8 bits of the receive buffer. Writing 0 to bits T31 to T24 at the time of transmission leads to 0 being set in the upper 8 bits of the receive buffer. If the SPCR3.RXMD bit is 1, 00h is stored in the upper 8 bits of the receive buffer.

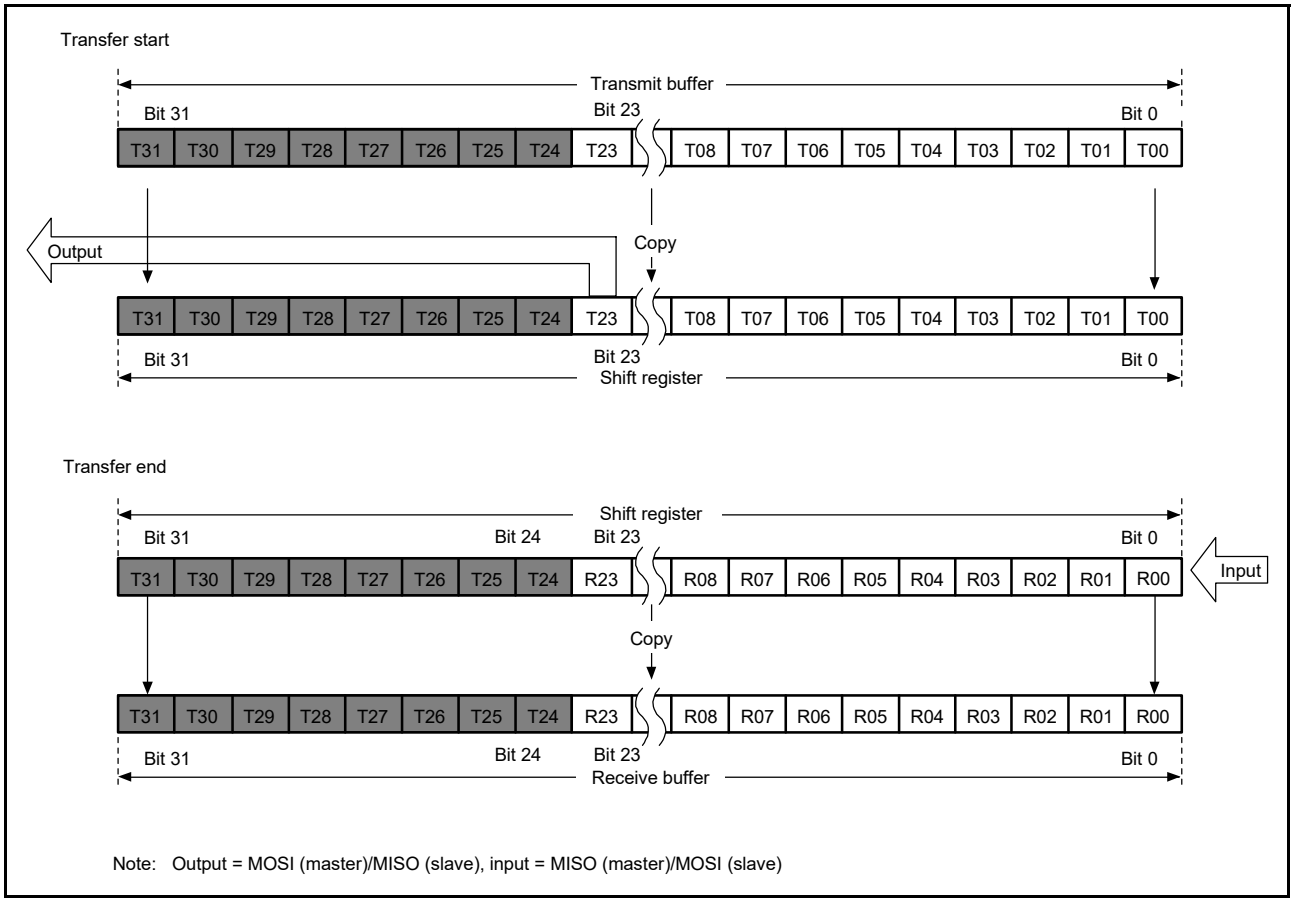


Figure 37.15 MSB First Transfer (24-Bit Data, Parity Disabled)

(3) LSB First Transfer (32-Bit Data)

Figure 37.16 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity disabled, an RSPI data length of 32 bits, and LSB first selected.

In transmission, bits T31 to T00 from the current stage of the transmit buffer are reordered bit by bit to obtain the order T00 to T31 for copying to the shift register. Data for transmission are shifted out from the shift register in order from T00, through T01, and so on to T31.

In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R00 to R31 have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer.

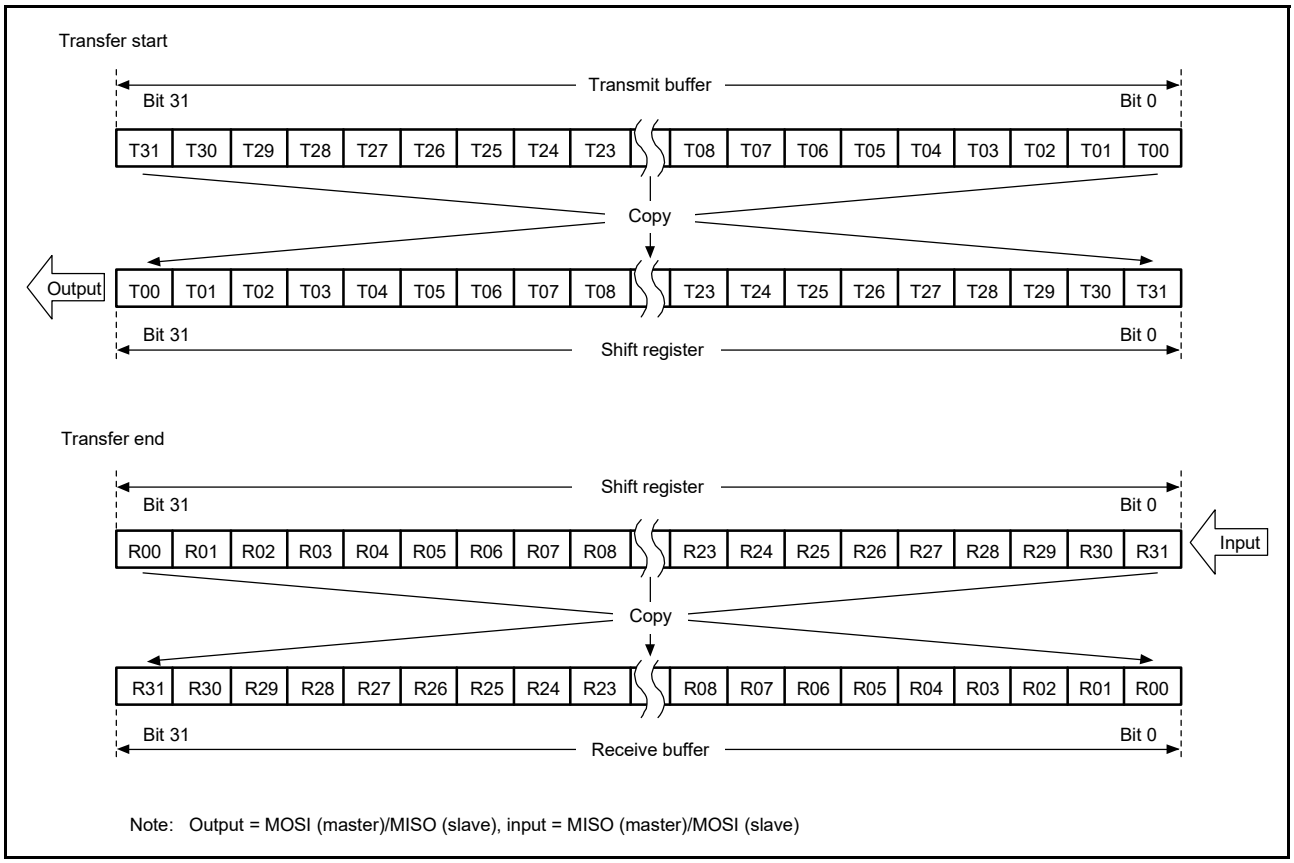


Figure 37.16 LSB First Transfer (32-Bit Data, Parity Disabled)

(4) LSB First Transfer (24-Bit Data)

Figure 37.17 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity disabled, 24 bits as the RSPI data length for an example that is not 32 bits, and LSB first selected.

In transmission, the lower 24 bits (T23 to T00) from the current stage of the transmit buffer are reordered bit by bit to obtain the order T00 to T23 for copying to the shift register. Data for transmission are shifted out from the shift register in order from T00, through T01, and so on to T23.

In reception, received data are shifted in bit by bit through bit 8 of the shift register. When bits R00 to R23 have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. At this time, if the SPCR3.RXMD bit is 0, the upper 8 bits of the transmit buffer are stored in the upper 8 bits of the receive buffer. Writing 0 to bits T31 to T24 at the time of transmission leads to 0 being set in the upper 8 bits of the receive buffer. If the SPCR3.RXMD bit is 1, 00h is stored in the upper 8 bits of the receive buffer.

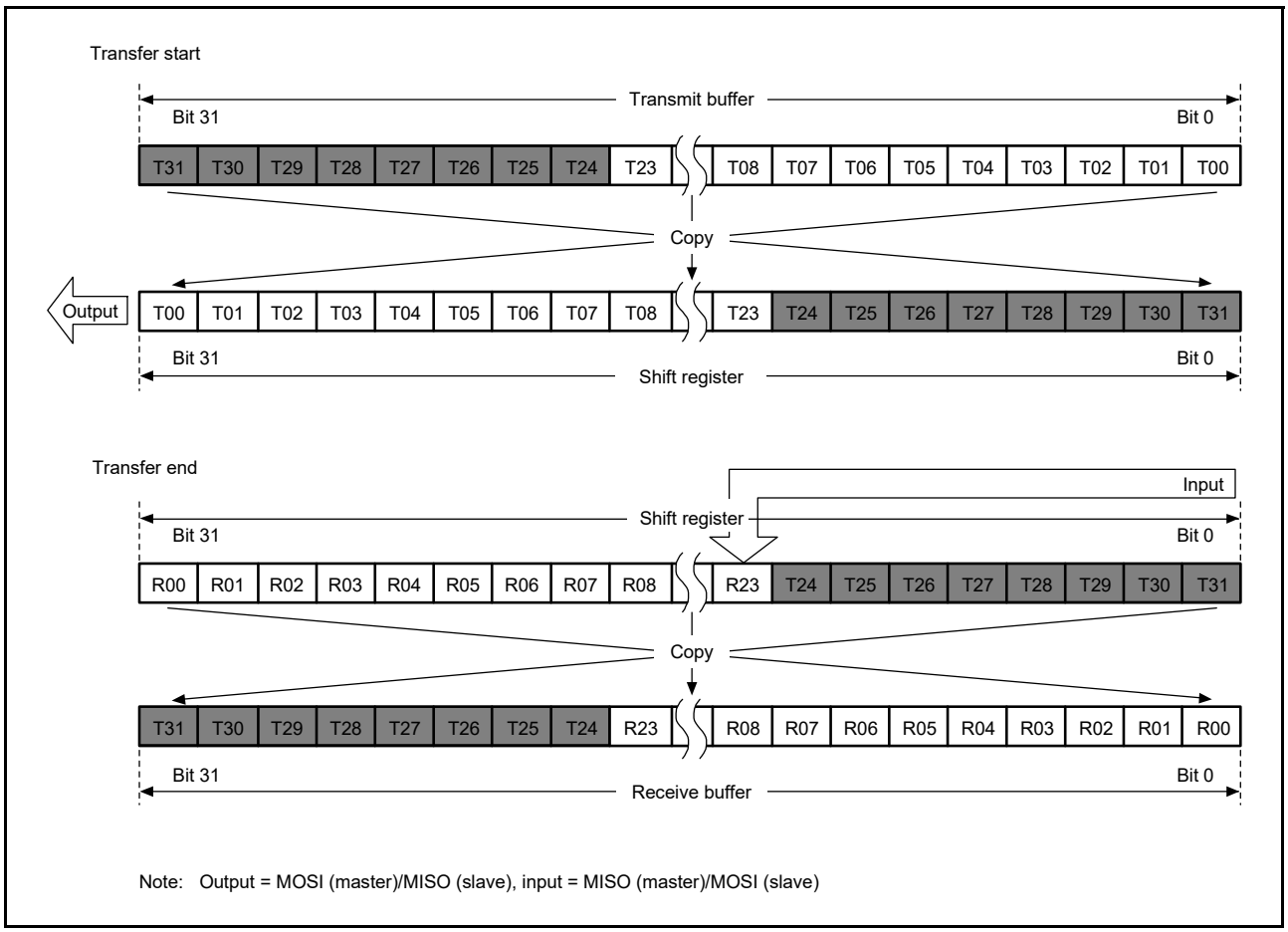


Figure 37.17 LSB First Transfer (24-Bit Data, Parity Disabled)

37.3.4.2 When Parity is Enabled (SPCR2.SPPE = 1)

When parity is enabled, the lowest-order bit of the data for transmission becomes a parity bit. Hardware calculates the value of the parity bit.

(1) MSB First Transfer (32-Bit Data)

Figure 37.18 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity enabled, an RSPI data length of 32 bits, and MSB first selected.
In transmission, the value of the parity bit (P) is calculated from bits T31 to T01. This replaces the final bit, T00, and the whole is copied to the shift register. Data are transmitted in the order T31, T30, …, T01, and P.
In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R31 to P have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R31 to P are checked by judging the parity.

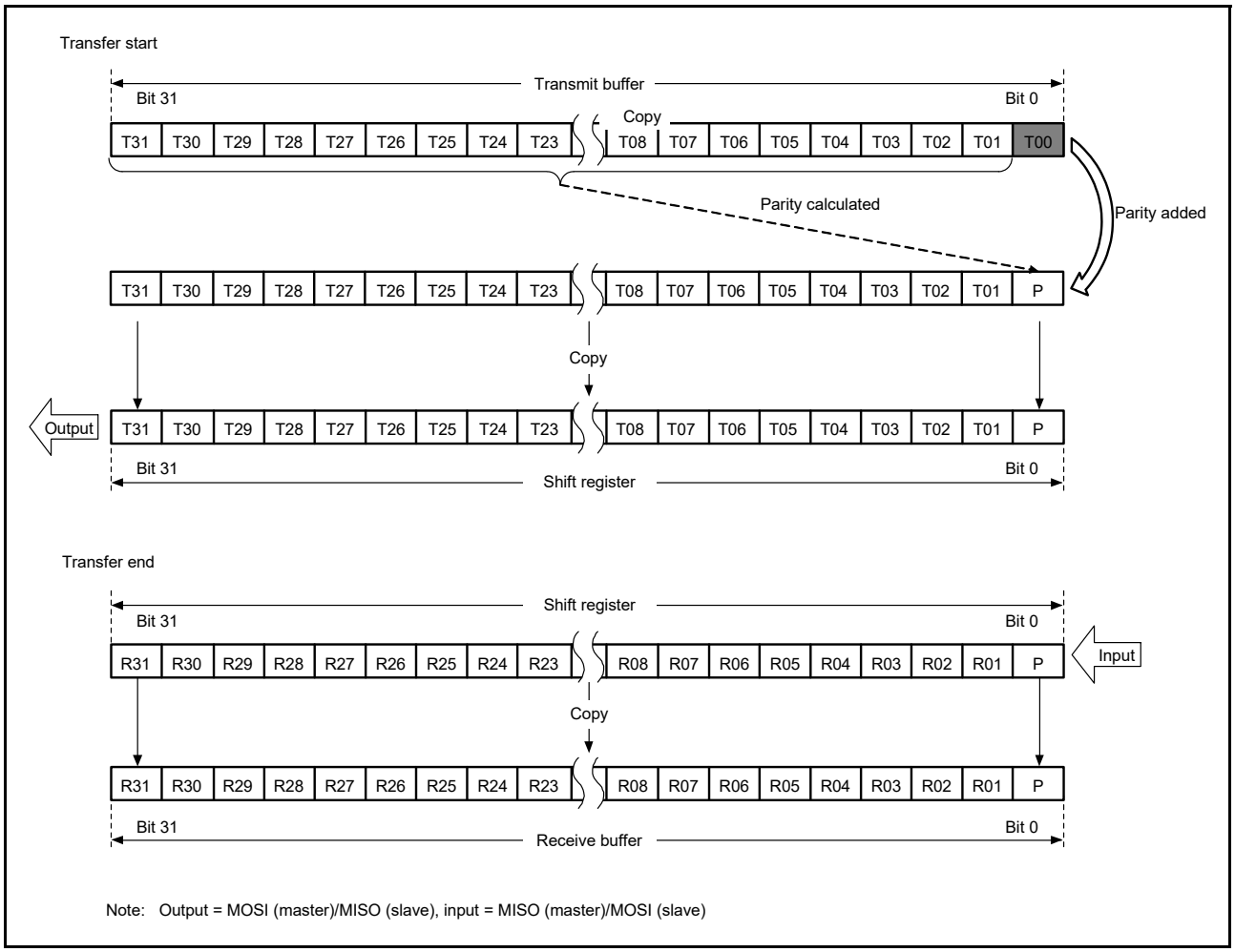


Figure 37.18 MSB First Transfer (32-Bit Data, Parity Enabled)

(2) MSB First Transfer (24-Bit Data)

Figure 37.19 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity enabled, 24 bits as the RSPI data length for an example that is not 32 bits, and MSB first selected.
In transmission, the value of the parity bit (P) is calculated from bits T23 to T01. This replaces the final bit, T00, and the whole is copied to the shift register. Data are transmitted in the order T23, T22, …, T01, and P.
In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R23 to P have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R23 to P are checked by judging the parity. At this time, if the SPCR3.RXMD bit is 0, the upper 8 bits of the transmit buffer are stored in the upper 8 bits of the receive buffer. Writing 0 to bits T31 to T24 at the time of transmission leads to 0 being set in the upper 8 bits of the receive buffer. If the SPCR3.RXMD bit is 1, 00h is stored in the upper 8 bits of the receive buffer.

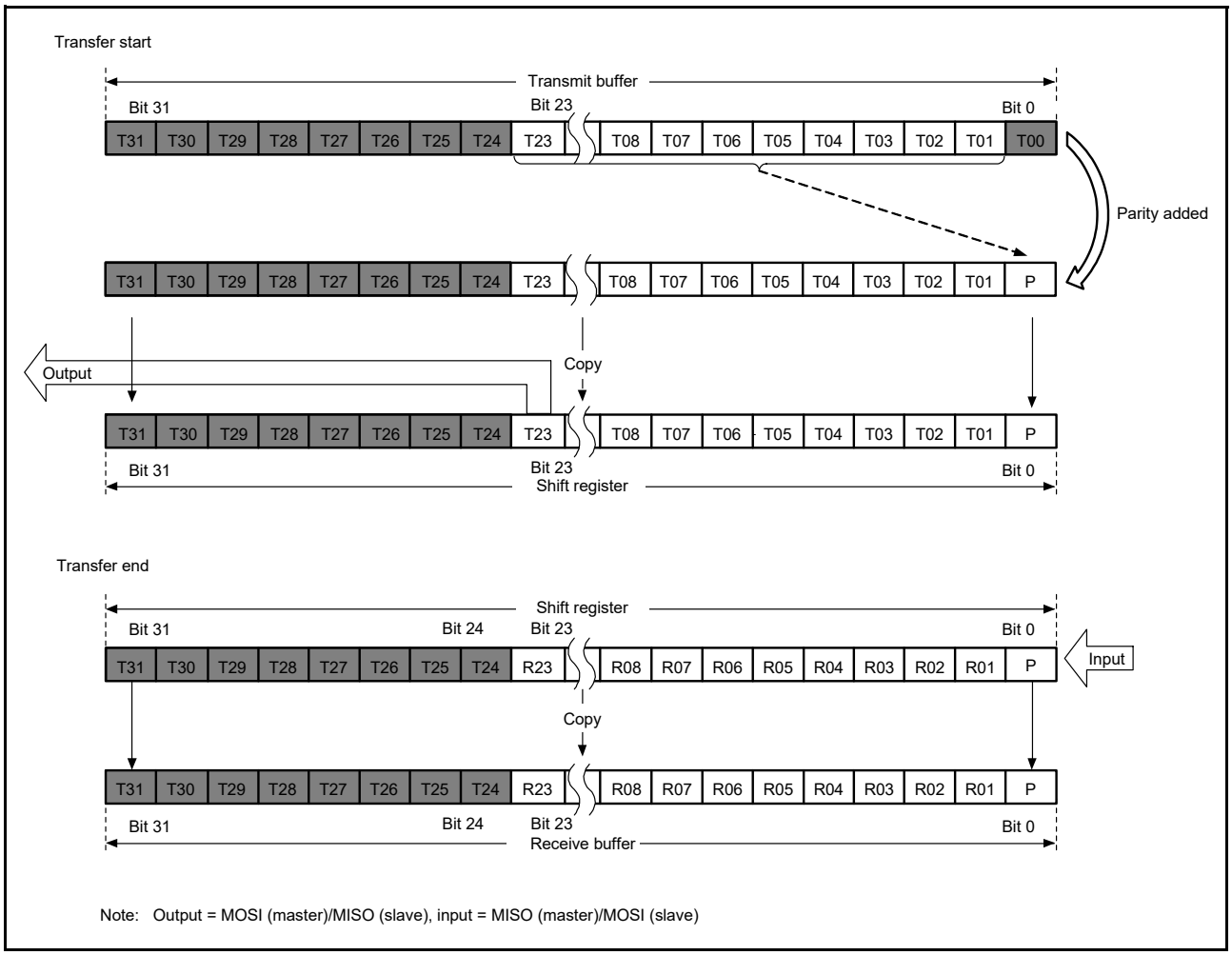


Figure 37.19 MSB First Transfer (24-Bit Data, Parity Enabled)

(3) LSB First Transfer (32-Bit Data)

Figure 37.20 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity enabled, an RSPI data length of 32 bits, and LSB first selected.

In transmission, the value of the parity bit (P) is calculated from bits T30 to T00. This replaces the final bit, T31, and the whole is copied to the shift register. Data are transmitted in the order T00, T01, …, T30, and P.

In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R00 to P have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R00 to P are checked by judging the parity.

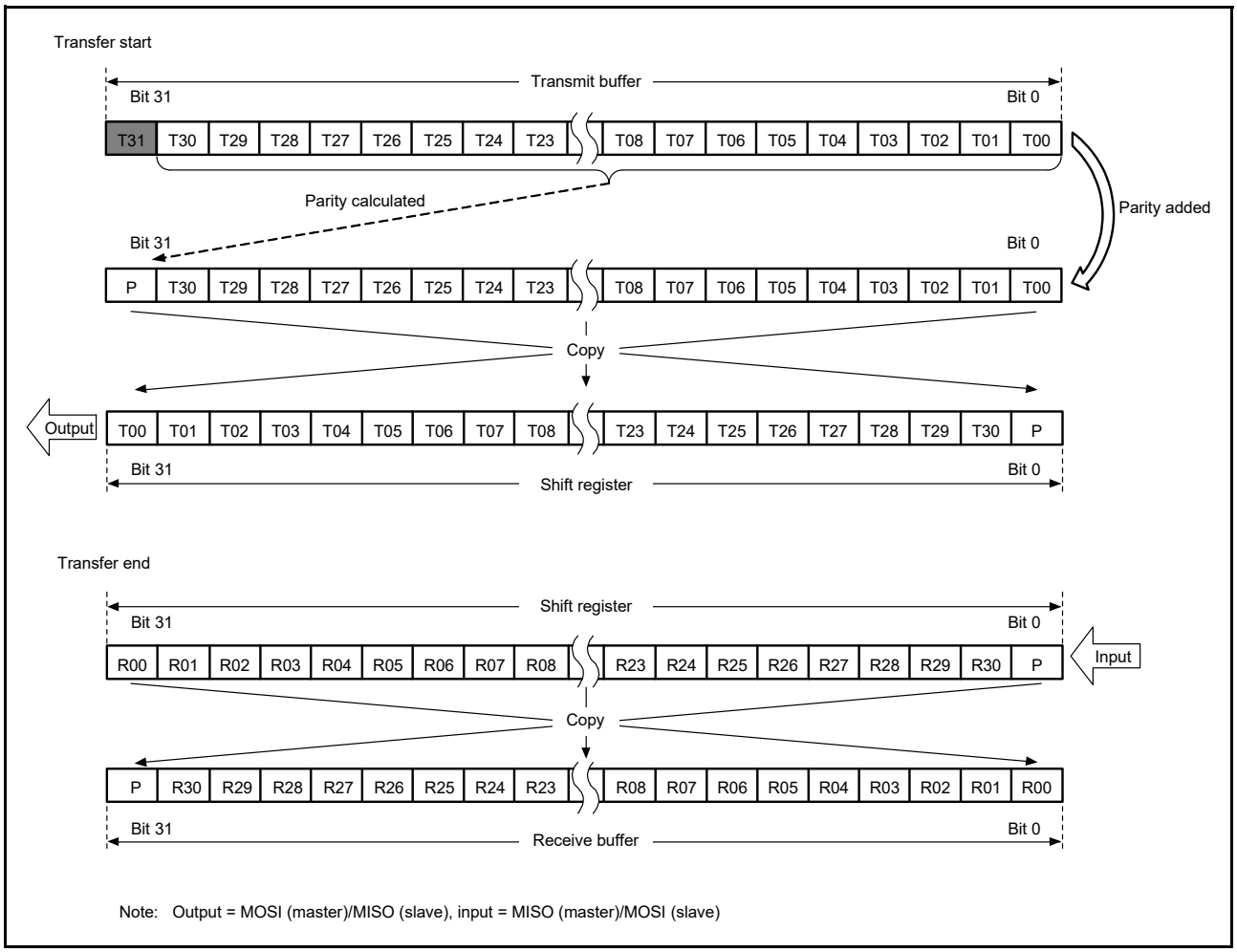


Figure 37.20 LSB First Transfer (32-Bit Data, Parity Enabled)

(4) LSB First Transfer (24-Bit Data)

Figure 37.21 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity enabled, 24 bits as the RSPI data length for an example that is not 32 bits, and LSB first selected.
In transmission, the value of the parity bit (P) is calculated from bits T22 to T00. This replaces the final bit, T23, and the whole is copied to the shift register. Data are transmitted in the order T00, T01, …, T22, and P.
In reception, received data are shifted in bit by bit through bit 8 of the shift register. When bits R00 to P have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R00 to P are checked by judging the parity. At this time, if the SPCR3.RXMD bit is 0, the upper 8 bits of the transmit buffer are stored in the upper 8 bits of the receive buffer. Writing 0 to bits T31 to T24 at the time of transmission leads to 0 being set in the upper 8 bits of the receive buffer. If the SPCR3.RXMD bit is 1, 00h is stored in the upper 8 bits of the receive buffer.

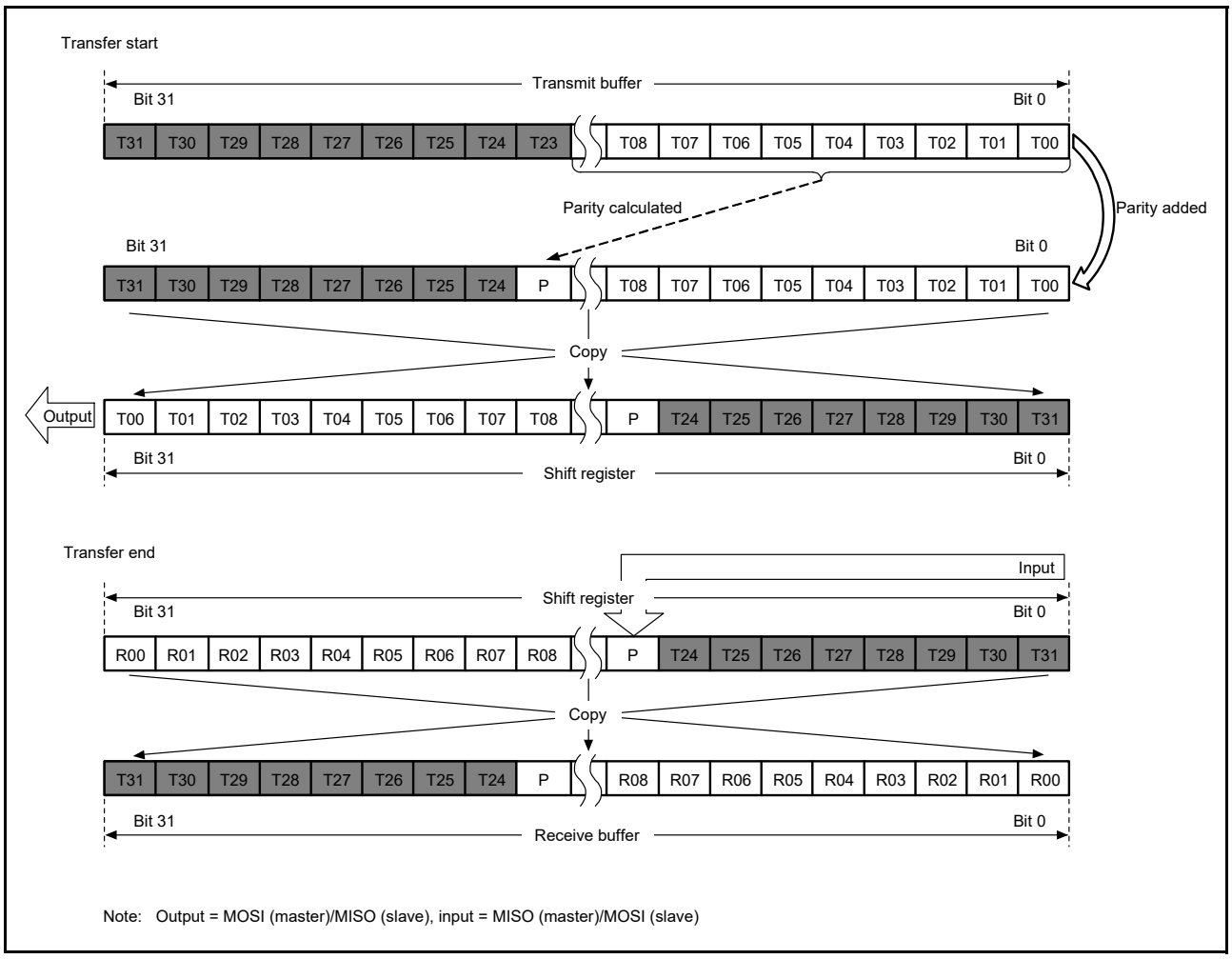


Figure 37.21 LSB First Transfer (24-Bit Data, Parity Enabled)

37.3.4.3 Byte Swap Transmission

When the SPDCR2.BYSW bit is 1 (byte swapping of SPDR data enabled), data bytes written in the transmit buffer (SPDR) will be swapped (in 8 bit units) when the data is transferred to the shift register. Figure 37.22 shows data transfer between the SPDR register and the shift register when data length is 32 bits.

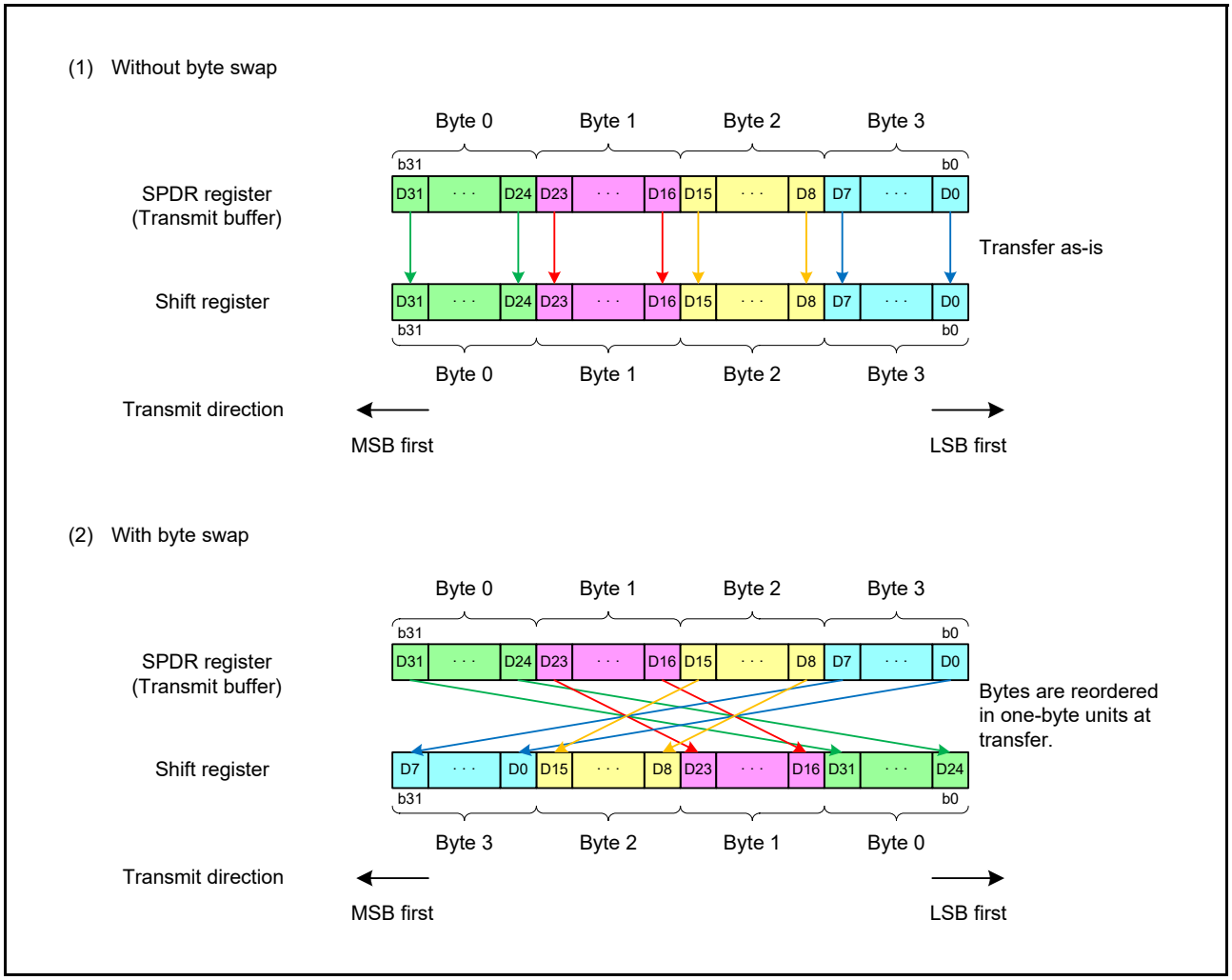


Figure 37.22 Transmit Data Handling in MSB/LSB First with Byte Swapping Enabled/Disabled

37.3.4.4 Byte Swap Reception

When the SPDCR2.BYSW bit is 1 (byte swapping of SPDR data enabled), data bytes in the shift register will be swapped (in 8 bit units) when the data is transferred to the receive buffer (SPDR). Figure 37.23 shows data transfer between the shift register and the SPDR register when data length is 32 bits.

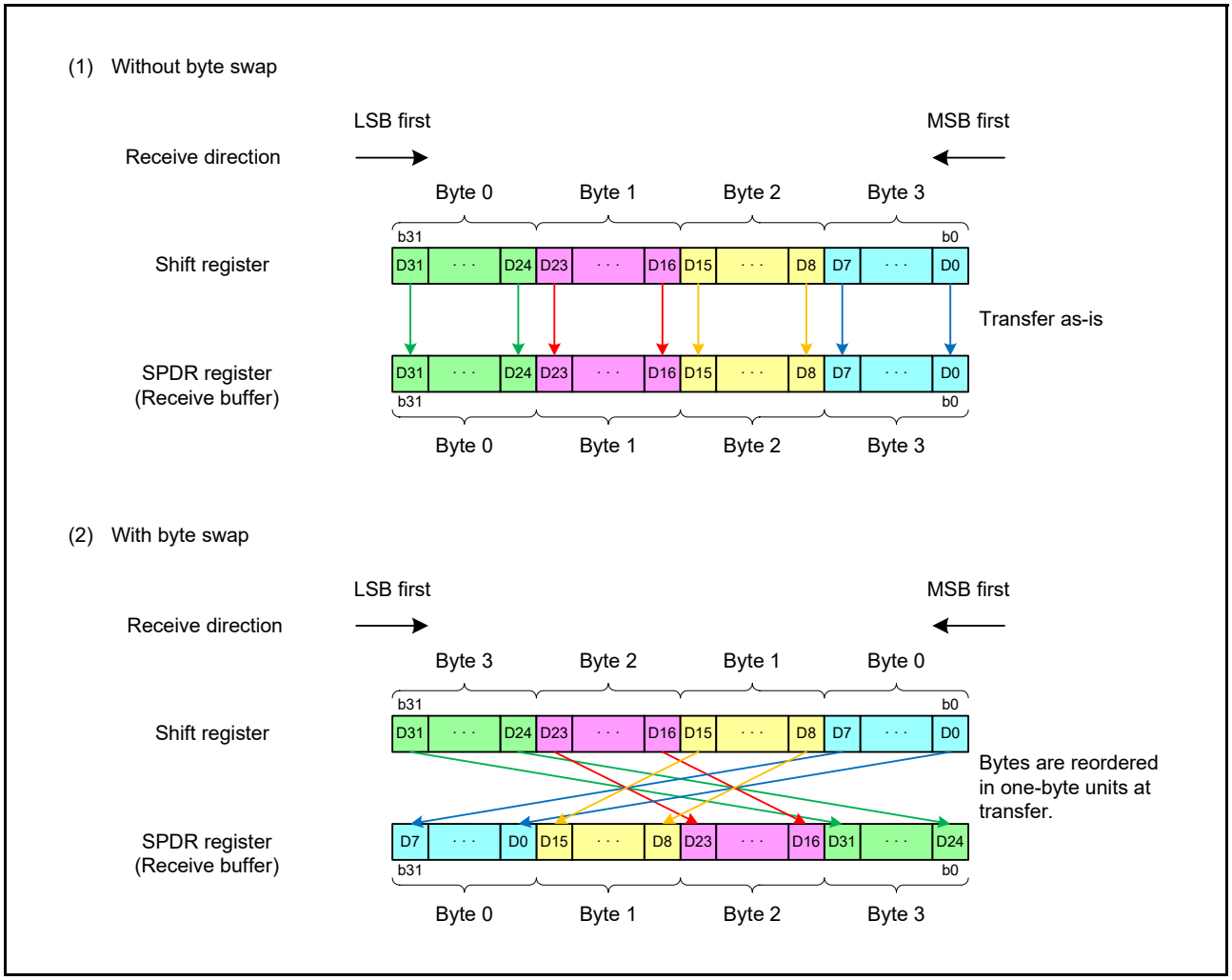


Figure 37.23 Receive Data Handling in MSB/LSB First with Byte Swapping Enabled/Disabled

37.3.5 Transfer Format

37.3.5.1 CPHA = 0

Figure 37.24 shows a sample transfer format for the serial transfer of 8-bit data when the SPCMDm.CPHA bit is 0. Note that clock synchronous operation (the SPCR.SPMS bit is 1) should not performed when the RSPI operates in slave mode (SPCR.MSTR = 0) and the CPHA bit is 0. In Figure 37.24, RSPCKA (CPOL = 0) indicates the RSPCKA signal waveform when the SPCMDm.CPOL bit is 0; RSPCKA (CPOL = 1) indicates the RSPCKA 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 I/O directions of the signals depend on the RSPI settings. For details, refer to section 37.3.2, Controlling RSPI Pins.

When the SPCMDm.CPHA bit is 0, the driving of valid data to the MOSIA and MISOA signals commences at an SSLAi signal assertion timing. The first RSPCKA signal change timing that occurs after the SSLAi signal assertion becomes the first transfer data fetch timing. After this timing, data is sampled at every 1 RSPCK cycle. The change timing for MOSIA and MISOA signals is 1/2 RSPCK cycles after the transfer data fetch timing. The CPOL bit setting does not affect the RSPCKA signal operation timing; it only affects the signal polarity.

t1 denotes a period from an SSLAi signal assertion to RSPCKA oscillation (RSPCK delay). t2 denotes a period from the termination of RSPCKA oscillation to an SSLAi signal negation (SSL negation delay). t3 denotes a period in which SSLAi 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 MCU is in master mode, refer to section 37.3.12.1, Master Mode Operation.

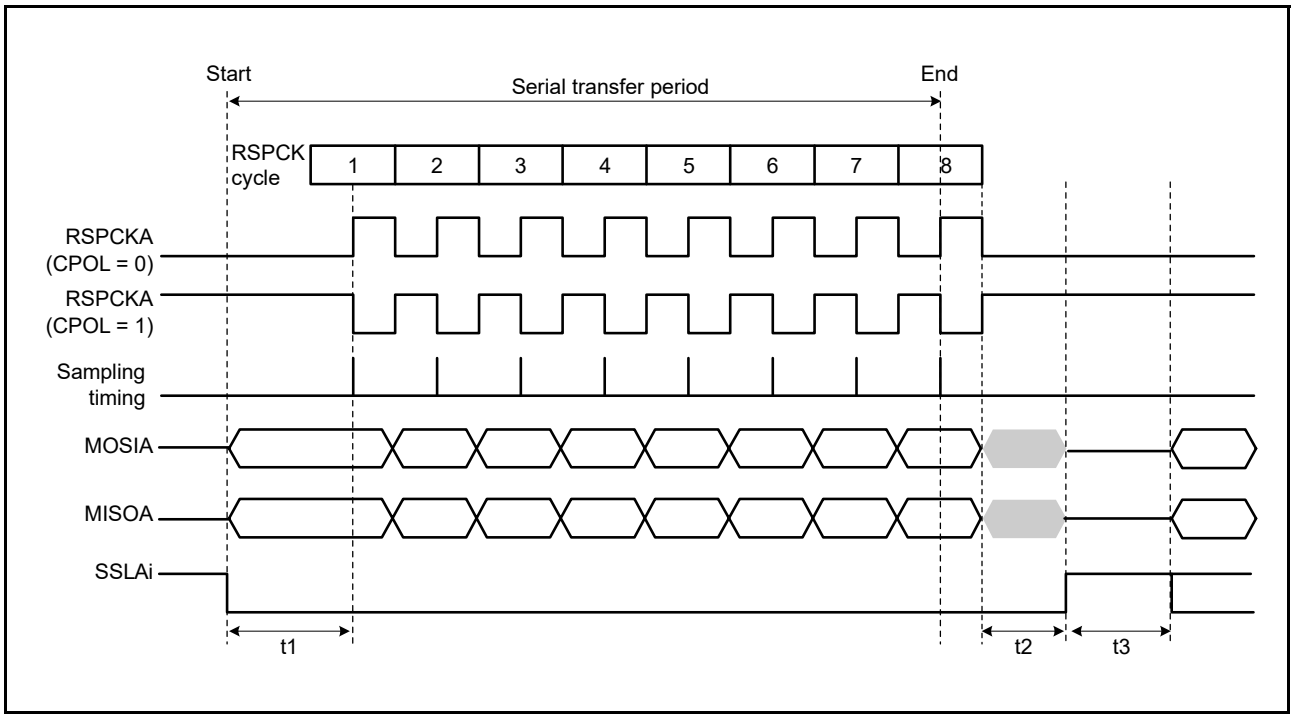


Figure 37.24 RSPI Transfer Format (CPHA = 0)

37.3.5.2 CPHA = 1

Figure 37.25 shows a sample transfer format for the serial transfer of 8-bit data when the SPCMDm.CPHA bit is 1. However, when the SPCR.SPMS bit is 1, the SSLAi signals are not used, and only the three signals RSPCKA, MOSIA, and MISOA handle communications. In Figure 37.25, RSPCK (CPOL = 0) indicates the RSPCKA signal waveform when the SPCMDm.CPOL bit is 0; RSPCKA (CPOL = 1) indicates the RSPCKA 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 I/O directions of the signals depend on the RSPI mode (master or slave). For details, refer to section 37.3.2, Controlling RSPI Pins.

When the SPCMDm.CPHA bit is 1, the driving of invalid data to the MISOA signal commences at an SSLAi signal assertion timing. The output of valid data to the MOSIA and MISOA signals commences at the first RSPCKA signal change timing that occurs after the SSLAi signal assertion. After this timing, data is updated at every 1 RSPCK cycle. The transfer data fetch timing is 1/2 RSPCK cycles after the data update timing. The SPCMDm.CPOL bit setting does not affect the RSPCKA signal operation timing; it only affects 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 MCU is in master mode, refer to section 37.3.12.1, Master Mode Operation.

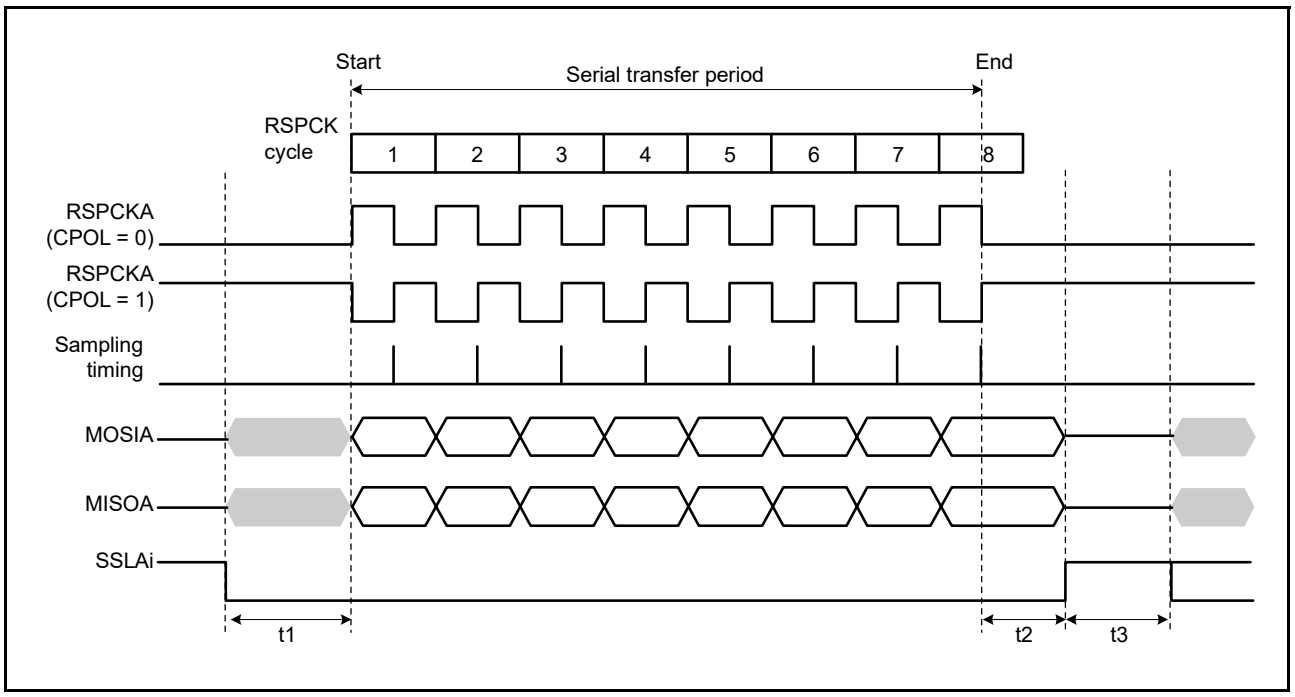


Figure 37.25 RSPI Transfer Format (CPHA = 1)

37.3.6 Communications Operating Mode

Full-duplex communications, transmit-only simplex communications, or receive-only simplex communications can be selected by the SPCR.TXMD and SPCR3.RXMD bits.
'SPDR access' shown in Figure 37.26 and Figure 37.27 indicate the condition of access to the SPDR register, where 'W' denotes a write cycle.

37.3.6.1 Full-Duplex Communications (SPCR.TXMD = 0, SPCR3.RXMD = 0)

Figure 37.26 shows an example of operation when the SPCR.TXMD bit is set to 0 and the SPCR3.RXMD bit is set to 0. In the example in Figure 37.26, the RSPI performs an 8-bit serial transfer in which the SPDCR.SPFC[1:0] bits are 00b, the SPCMDm.CPHA bit is 1, and the SPCMDm.CPOL bit is 0. The numbers given under the RSPCKA waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

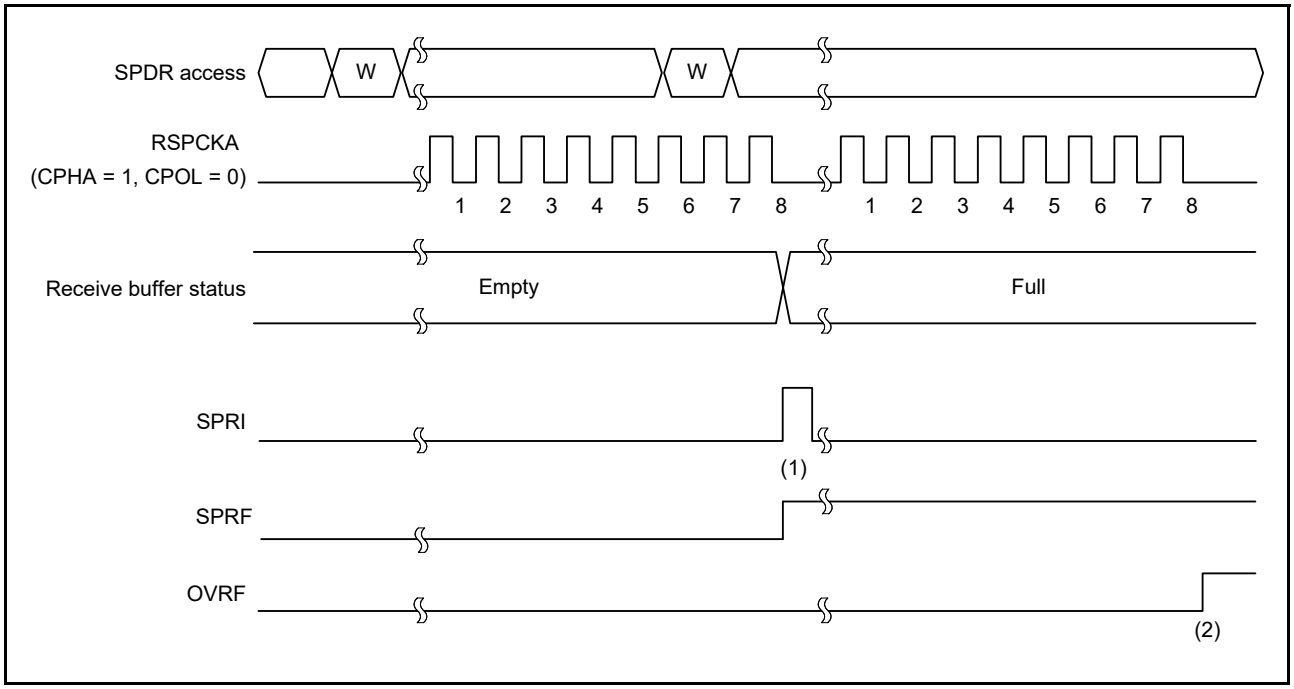


Figure 37.26 Operation Example of SPCR.TXMD = 0 and SPCR3.RXMD = 0

The operation of the flags at timings shown in steps (1) and (2) in the figure is described below.

(1) When a serial transfer ends with the receive buffer of the SPDR register empty, the RSPI generates a receive buffer full interrupt request (SPRI) (sets the SPSR.SPRF flag to 1) and copies the received data in the shift register to the receive buffer.

(2) When a serial transfer ends with the receive buffer of the SPDR register holding data that was received in the previous serial transfer, the RSPI sets the SPSR.OVRF flag to 1 and discards the received data in the shift register.

When full-duplex communications (SPCR.TXMD = 0, SPCR3.RXMD = 0) is selected, reception occurs simultaneously with transmit operations. As such, the SPRF and OVRF flags in the SPSR register become 1 at the timing described in (1) and (2), respectively, according to the state of the receive buffer.

37.3.6.2 Transmit-only Simplex Communications (SPCR.TXMD = 1, SPCR3.RXMD = 0)

Figure 37.27 shows an example of operation when the SPCR.TXMD bit is set to 1 and the SPCR3.RXMD bit is set to 0. In the example in Figure 37.27, the RSPI performs an 8-bit serial transfer in which the SPDCR.SPFC[1:0] bits are 00b, the SPCMDm.CPHA bit is 1, and the SPCMDm.CPOL bit is 0. The numbers given under the RSPCKA waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

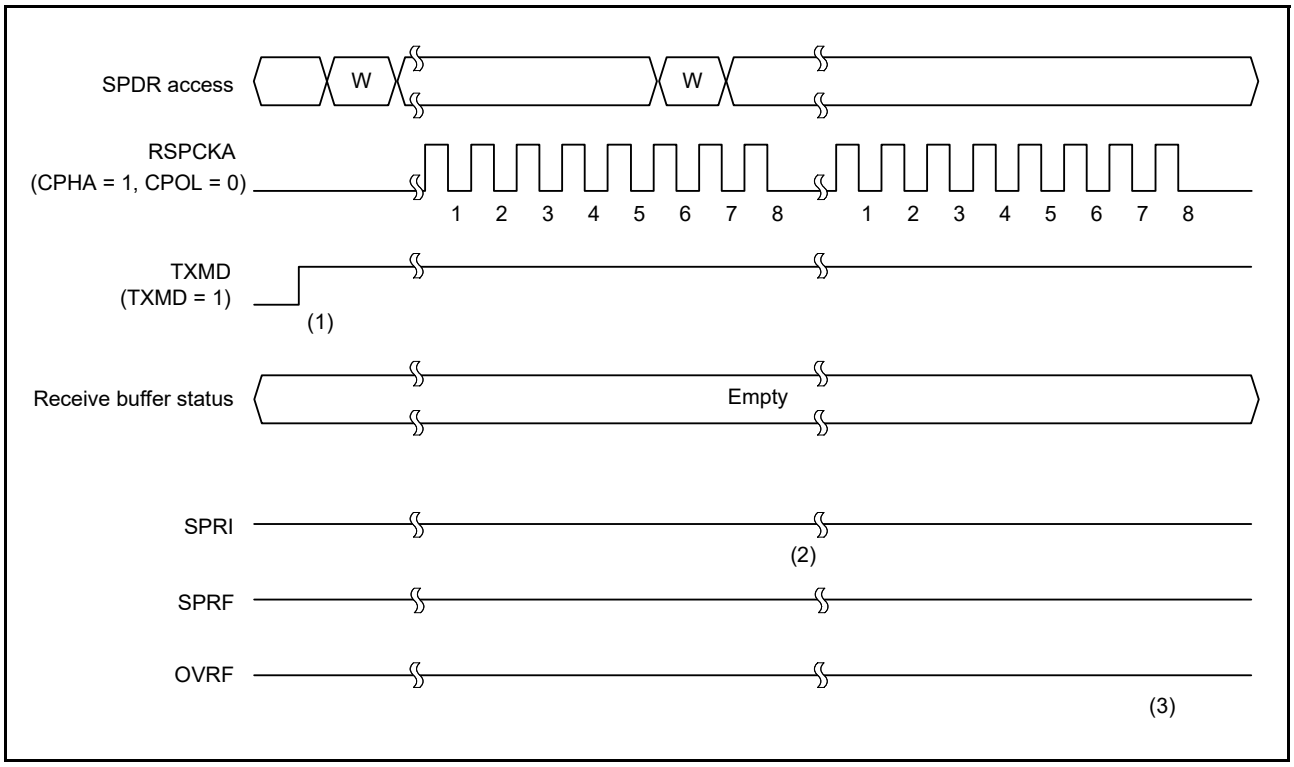


Figure 37.27 Operation Example of SPCR.TXMD = 1 and SPCR3.RXMD = 0

The operation of the flags at timings shown in steps (1) to (3) in the figure is described below.

(1) Make sure there is no data left in the receive buffer and the SPSR.SPRF, OVRF flags are 0 before entering the mode of transmit-only simplex communications (SPCR.TXMD = 1, SPCR3.RXMD = 0).

(2) When a serial transfer ends with the receive buffer of the SPDR register empty, if the mode of transmit-only simplex communications is selected (SPCR.TXMD = 1, SPCR3.RXMD = 0), the SPRF flag remains 0 and the RSPI does not copy the data from the shift register to the receive buffer.

(3) Since the receive buffer of the SPDR register does not hold data that was received in the previous serial transfer, even when a serial transfer ends, the SPSR.OVRF flag retains the value of 0, and the data in the shift register is not copied to the receive buffer.

When performing transmit-only simplex communications (SPCR.TXMD = 1, SPCR3.RXMD = 0), the RSPI transmits data but does not receive data. Therefore, the SPSR.SPRF, OVRF flags remain 0 at the timings of (1) to (3).

37.3.6.3 Receive-only Simplex Communications (SPCR3.RXMD = 0)

The receive-only simplex communications are valid only when the SPCR.MSTR bit is 0 (slave mode).
Figure 37.28 shows an example of operation when the SPCR3.RXMD bit is set to 1. In the example in Figure 37.28, the RSPI performs an 8-bit serial transfer in which the SPDCR.SPFC[1:0] bits are 00b, the SPCMDm.CPHA bit is 1, and the SPCMDm.CPOL bit is 0. The numbers given under the RSPCKA waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

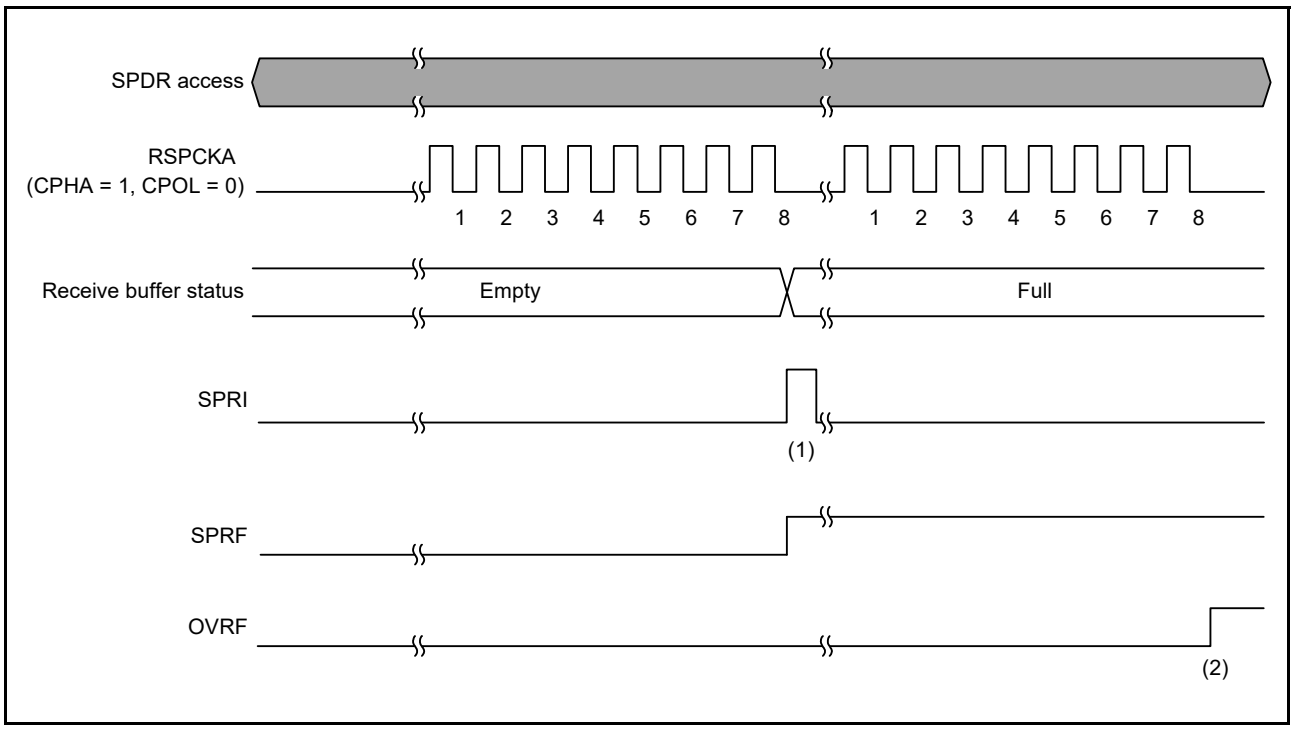


Figure 37.28 Operation Example of SPCR3.RXMD = 1

The operation of the flags at timings shown in steps (1) and (2) in the figure is described below.

(1) When a serial transfer ends with the receive buffer of the SPDR register empty, the RSPI generates a receive buffer full interrupt request (SPRI) (sets the SPSR.SPRF flag to 1) and copies the received data in the shift register to the receive buffer.

(2) When a serial transfer ends with the receive buffer of the SPDR register holding data that was received in the previous serial transfer, the RSPI sets the SPSR.OVRF flag to 1 and discards the received data in the shift register.

37.3.7 Transmit Buffer Empty/Receive Buffer Full Interrupts

Figure 37.29 shows an example of operation of the transmit buffer empty interrupt (SPTI) and the receive buffer full interrupt (SPRI). 'SPDR access' shown in Figure 37.29 indicates the condition of access to the SPDR register, where 'W' denotes a write cycle, and 'R' a read cycle. In the example in Figure 37.29, the RSPI performs an 8-bit serial transfer in which the SPCR.TXMD bit is 0, the SPCR3.RXMD bit is 0, the SPDCR.SPFC[1:0] bits are 00b, the SPCMDm.CPHA bit is 1, and the SPCMDm.CPOL bit is 0. The numbers given under the RSPCKA waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

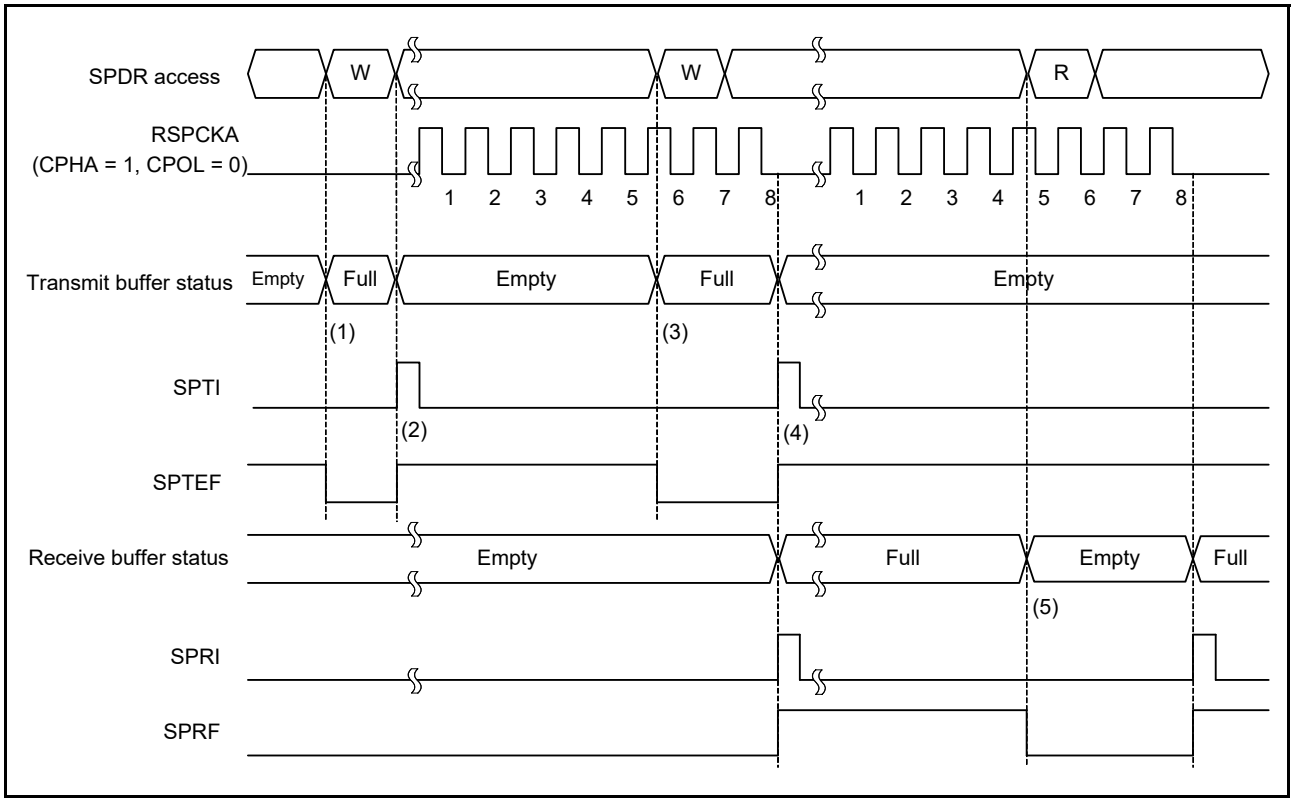


Figure 37.29 Operation Example of SPTI and SPRI Interrupts

The operation of the interrupts at timings shown in steps (1) to (5) in the figure is described below.

(1) When transmit data is written to the SPDR register when the transmit buffer of the SPDR register is empty (data for the next transfer is not set), the RSPI writes data to the transmit buffer and sets the SPSR.SPTEF flag to 0.

(2) If the shift register is empty, the RSPI copies the data from the transmit buffer to the shift register and generates a transmit buffer empty interrupt request (SPTI) and sets the SPSR.SPTEF flag to 1. How a serial transfer is started depends on the mode of the RSPI. For details, refer to section 37.3.12, SPI Operation, and section 37.3.13, Clock Synchronous Operation.

(3) When transmit data is written to the SPDR register in the transmit buffer empty interrupt routine or in the transmit buffer empty detecting process by polling the SPTEF flag, the data is transferred to the transmit buffer and the SPSR.SPTEF flag becomes 0. Because the data being transmitted is stored in the shift register, the RSPI does not copy the data from the transmit buffer to the shift register.

(4) When the serial transfer ends with the receive buffer of the SPDR register being empty, the RSPI copies the receive data from the shift register to the receive buffer, generates a receive buffer full interrupt request (SPRI), and sets the SPSR.SPRF flag to 1. Since the shift register becomes empty upon completion of serial transfer, when the transmit buffer had been full before the serial transfer ended, the RSPI sets the SPSR.SPTEF flag to 1 and copies the data from the transmit 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, thus data transfer from the transmit buffer to the shift register is enabled.

(5) When the SPDR register is read in the receive buffer full interrupt routine or in the receive buffer full detecting process by polling the SPRF flag, the receive data can be read. When the receive data is read, the SPRF flag becomes 0.

If transmit data is written to the SPDR register while the transmit buffer holds data that has not yet been transmitted (the SPTEF flag is 0), the RSPI does not update the data in the transmit buffer. Transmit data should be written to the SPDR register in the transmit buffer empty interrupt request routine or in the transmit buffer empty detecting process by polling the SPTEF flag. To use a transmit buffer empty interrupt, set the SPTIE bit in SPCR to 1.
When setting the SPCR.SPE bit to 0 (RSPI disabled), the SPCR.SPTIE bit should also be set to 0. Otherwise (if the SPCR.SPE bit is 0 and the SPCR.SPTIE is 1), a transmit buffer empty interrupt request will occur.
When serial transfer ends with the receive buffer being full (the SPRF flag is 1), the RSPI does not copy data from the shift register to the receive buffer, and detects an overrun error (refer to section 37.3.10, Error Detection). To prevent a receive data overrun error, read the received data using a receive buffer full interrupt request before the next serial transfer ends. To use an receive buffer full interrupt, set the SPCR.SPRIE bit to 1.
Transmit and receive interrupts or the corresponding IRn.IR flags (where n is the interrupt vector number) in the ICU can be used to confirm the states of the transmit and receive buffers. Refer to section 14, Interrupt Controller (ICUG), for the interrupt vector numbers. The status of the transmit and receive buffers can be also confirmed by the SPTEF and SPRF flags.

37.3.8 Idle Interrupt

When the SPSR.IDLNF flag becomes 0 while the SPCR2.SPIIE bit is 1, an idle interrupt request (SPII) is generated.
In master mode, the IDLNF flag is 0 before transmission. Therefore, write data in the transmit buffer and set the SPIIE bit to 1 after the IDLNF flag becomes 1 so that an idle interrupt is not generated at this time. When the SSLA0 signal is negated after the transmission is completed and the next data is not supplied until next-access delay time (t3) elapses, the IDLNF flag becomes 0.

37.3.9 Communication End Interrupt

When the SPSR.SPCF flag becomes 1 while the SPCR3.SPCIE bit is 1 or when the SPCIE bit is set to 1 while the SPCR.SPE bit is 1 and SPCF flag is 1, a communication end interrupt request (SPCI) is generated.
The conditions for setting the SPCF flag to 1 differ depending on the operating mode of the RSPI. For details, refer to section 37.2.4, RSPI Status Register (SPSR).

37.3.9.1 In Master Mode

In master mode, the conditions for setting the SPCF flag to 1 are the same in any combination of SPI operation or clock synchronous operation and full-duplex or transmit-only simplex communications.
When the SPSSR.SPCP[2:0] bits becomes 000b and there is no next transmit data, the SPSR.IDLNF flag becomes 0 and the SPCF flag becomes 1.
The SPCF flag is cleared when 0 is written to the SPCF flag after confirming the flag is 1 or when the next data is written to the transmit buffer.

37.3.9.2 Full-duplex or transmit-only simplex communications in slave mode under SPI operation

When the SSLA0 signal is negated while the transmit buffer and the transmit shift register are empty, the SPSR.SPCF flag becomes 1.
The SPCF flag is cleared when 0 is written to the SPCF flag after confirming the flag is 1 or when the next data is written

to the transmit buffer.

37.3.9.3 Receive-only simplex communications in slave mode under SPI operation

When the SSLA0 signal is negated after the number of frames set in the SPDCR.SPFC[1:0] bits have been received, the SPSR.SPCF flag becomes 1.
The SPCF flag is cleared when 0 is written to the SPCF flag after confirming the flag is 1 or when the SSLA0 signal is asserted when the next communication is started.

37.3.9.4 Full-duplex or transmit-only simplex communications in slave mode under clock synchronous operation

When the transmit buffer and the transmit shift register are empty, the SPSR.SPCF flag becomes 1 at the sampling timing of the last bit (the last even edge of the RSPCK).
The SPCF flag is cleared when 0 is written to the SPCF flag after confirming the flag is 1 or when the next data is written to the transmit buffer.

37.3.9.5 Receive-only simplex communications in slave mode under clock synchronous operation

When the number of frames set in the SPDCR.SPFC[1:0] bits have been received (the last even edge of the RSPCK), the SPSR.SPCF flag becomes 1.
The SPCF flag is cleared when 0 is written to the SPCF flag after confirming the flag is 1 or when the RSPCK signal changes when next communication is started.

37.3.10 Error Detection

In the normal RSPI serial transfer, the data written to the transmit buffer of the SPDR register is transmitted, and the received data can be read from the receive buffer of the SPDR register. If access is made to the SPDR register, depending on the status of the transmit/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 underrun error, overrun error, parity error, or mode fault error. Table 37.7 lists the relationship between non-normal transfer operations and the RSPI's error detection function.

Table 37.7 Relationship between Non-Normal Transfer Operations and RSPI Error Detection Function

	Occurrence Condition	RSPI Operation	Error Detection
1	The SPDR register is written when the transmit buffer is full.	• The contents of the transmit buffer are kept. • Missing write data.	None
2	The SPDR register is read when the receive buffer is empty.	If reception has completed, then the received data is output to the bus. Otherwise, data received previously is output instead.	None
3	Serial transfer is started when transmit data is still not loaded on the shift register while the RSPI is set to perform full-duplex or transmit-only simplex communications in slave mode.	• Serial transfer is suspended • Transmit/receive data is missing • The MISO signal output is disabled • RSPI function is disabled	Underrun error
4	Serial transfer terminates when the receive buffer is full.	• The contents of the receive buffer are kept. • Missing receive data.	Overrun error
5	An incorrect parity bit is received when performing full-duplex or receive-only simplex communications with the parity function enabled.	The parity error flag is set.	Parity error
6	The SSLA0 input signal is asserted when the serial transfer is idle in multi-master mode.	• Driving of the RSPCKA, MOSIA, SSLA1 to SSLA3 output signals is stopped. • RSPI function is disabled.	Mode fault error
7	The SSLA0 input signal is asserted during serial transfer in multi-master mode.	• Serial transfer is suspended. • Missing transmit/receive data. • Driving of the RSPCKA, MOSIA, SSLA1 to SSLA3 output signals is stopped. • RSPI function is disabled.	Mode fault error
8	The SSLA0 input signal is negated during serial transfer in slave mode.	• Serial transfer is suspended. • Missing transmit/receive data. • Driving of the MISOA output signal is stopped. • RSPI function is disabled.	Mode fault error

On operation 1 described in Table 37.7, the RSPI does not detect an error. To prevent data omission during the writing to the SPDR register, the SPDR register should be written when a transmit buffer empty interrupt request occurs or while the SPSR.SPTEF flag is 1.

Likewise, the RSPI does not detect an error on operation 2. To prevent extraneous data from being read, the SPDR register should be read when a receive buffer full interrupt request occurs or while the SPSR.SPRF flag is 1.

An underrun error shown in 3 is described in section 37.3.10.4, Underrun Error. An overrun error shown in 4 is described in section 37.3.10.1, Overrun Error. A parity error shown in 5 is described in section 37.3.10.2, Parity Error. A mode fault error shown in 6 to 8 is described in section 37.3.10.3, Mode Fault Error.

For the transmit and receive interrupts, refer to section 37.3.7, Transmit Buffer Empty/Receive Buffer Full Interrupts.

37.3.10.1 Overrun Error

If a serial transfer ends when the receive buffer of the SPDR register is full, the RSPI detects an overrun error, and sets the SPSR.OVRF flag to 1. When the OVRF flag 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 set the OVRF flag to 0, write 0 to the OVRF flag after the CPU has read the SPSR register with the OVRF flag set to 1.
Figure 37.30 shows an example of operations of the SPRF and OVRF flags. 'SPSR access' and 'SPDR access' shown in Figure 37.30 indicate the condition of accesses to the SPSR and SPDR registers, respectively, where 'W' denotes a write cycle, and 'R' a read cycle. In the example in Figure 37.30, the RSPI performs an 8-bit serial transfer in which the SPCMDm.CPHA bit is 1 and the SPCMDm.CPOL bit is 0. The numbers given under the RSPCKA waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

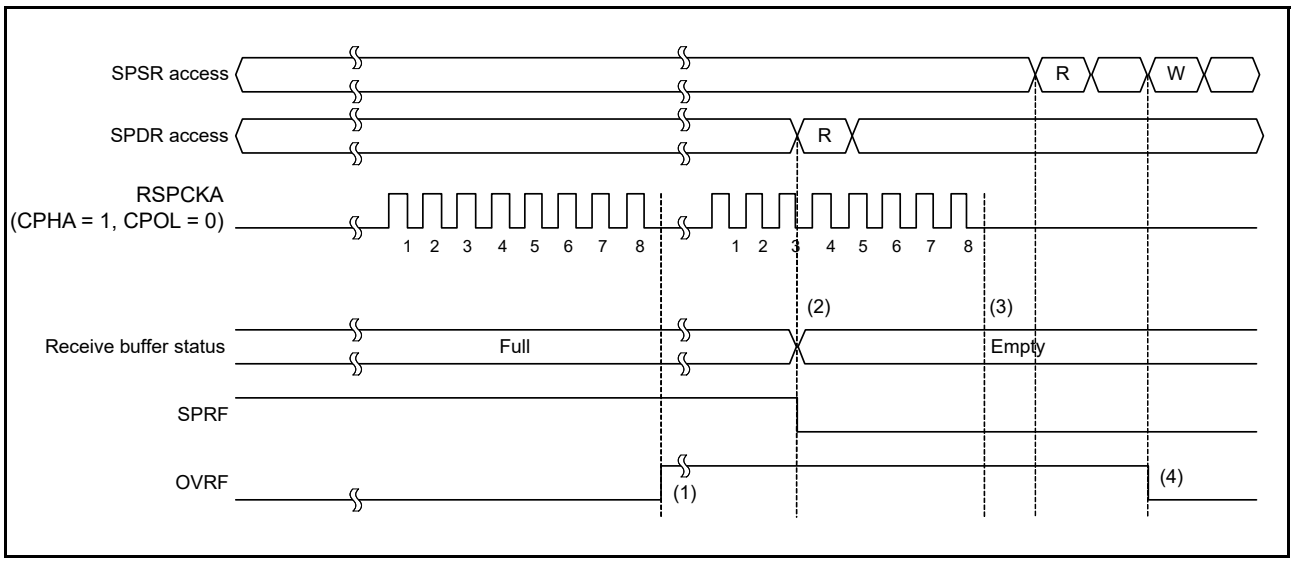


Figure 37.30 Operation Example of the SPRF and OVRF Flags

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 receive buffer full (the SPRF flag is 1), the RSPI detects an overrun error, and sets the OVRF flag to 1. The RSPI does not copy the data in the shift register to the receive buffer. Even if the SPPE bit is 1, parity errors are not detected. In master mode, the RSPI copies the pointer value to the SPCMDm register to the SPSSR.SPECM[2:0] bits.

(2) When the SPDR register is read, the RSPI outputs the data in the receive buffer. At this time the SPRF flag becomes 0. Even if the receive buffer becomes empty, the OVRF flag does not become 0.

(3) If the serial transfer ends with the OVRF flag being 1 (an overrun error occurs), the RSPI does not copy the data in the shift register to the receive buffer (the SPRF flag remains 0). A receive buffer full interrupt is not generated. Even if the SPPE bit is 1, parity errors are not detected. When in master mode, the RSPI does not update the SPSSR.SPECM[2:0] bits. When in an overrun error state and 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 from the transmit buffer to the shift register is enabled.

(4) If 0 is written to the OVRF flag after the SPSR register is read when the OVRF flag is 1, the OVRF flag is set to 0.

The occurrence of an overrun can be checked either by reading the SPSR register or by using an error interrupt and reading the SPSR register. When executing a serial transfer, measures should be taken to ensure the early detection of overrun errors, such as reading the SPSR register immediately after the SPDR register is read. When the RSPI is used in master mode, the pointer value to the SPCMDm register at the occurrence of the error can be checked by reading the SPSSR.SPECM[2:0] bits.
If an overrun error occurs and the OVRF flag is set to 1, normal reception operations cannot be performed until the OVRF flag is set to 0.

When the RSPCK auto-stop function is enabled in master mode, an overrun error does not occur. Figure 37.31 and Figure 37.32 show the clock stop waveform when a serial transfer continues while the receive buffer is full in master mode.

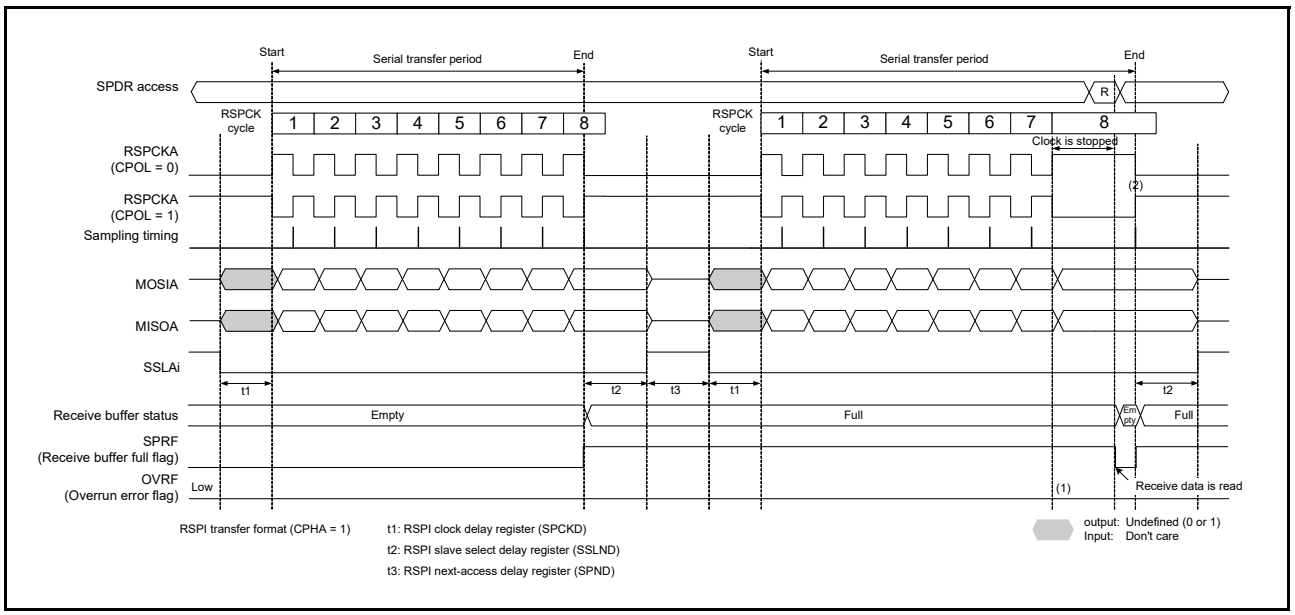


Figure 37.31 Clock Stop Waveform When a Serial Transfer Continues While the Receive Buffer is Full in Master Mode (CPHA = 1)

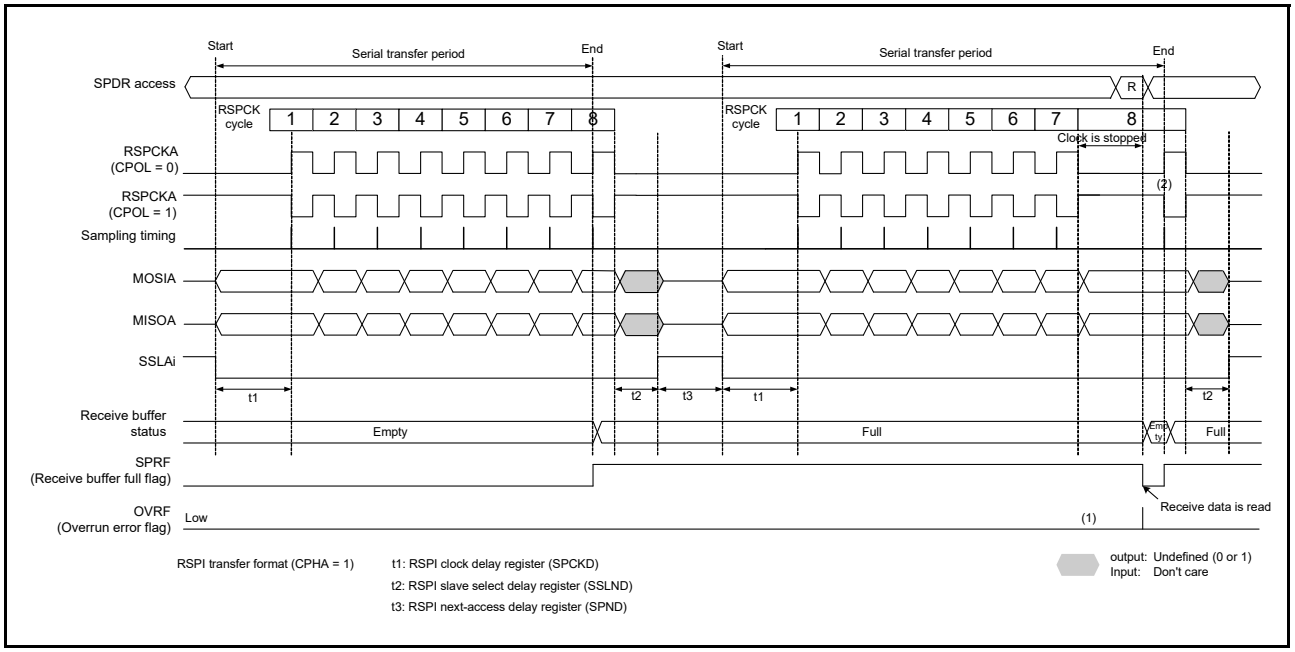


Figure 37.32 Clock Stop Waveform When a Serial Transfer Continues While the Receive Buffer is Full in Master Mode (CPHA = 0)

The operation of the flags at the timings shown in steps (1) and (2) in the figure is described below.

(1) When the receive buffer is full, an overrun error does not occur because the RSPCK clock is stopped.

(2) If the SPDR register is read while the clock is stopped, data in the receive buffer can be read. The RSPCK clock restarts after reading the receive buffer (after the SPRF flag becomes 0).

37.3.10.2 Parity Error

If full-duplex communication or receive-only simplex communication is performed with the SPCR.TXMD bit set to 0 and the SPCR2.SPPE bit set to 1, when serial transfer ends, the RSPI checks whether there are parity errors. Upon detecting a parity error in the received data, the RSPI sets the SPSR.PERF flag to 1. Since the RSPI does not copy the data in the shift register to the receive buffer when the SPSR.OVRF flag is set to 1, parity error detection is not performed for the received data. To set the PERF flag to 0, write 0 to the PERF flag after the SPSR register is read with the PERF flag set to 1.

Figure 37.33 shows an example of operation of the OVRF and PERF flags. 'SPSR access' shown in Figure 37.33 indicates the condition of access to the SPSR register, where 'W' denotes a write cycle, and 'R' a read cycle. In the example of Figure 37.33, full-duplex communication or receive-only simplex communication is performed while the SPCR.TXMD bit is 0 and the SPCR2.SPPE bit is 1. The RSPI performs an 8-bit serial transfer in which the SPCMDm.CPHA bit is 1 and the SPCMDm.CPOL bit is 0. The numbers given under the RSPCKA waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

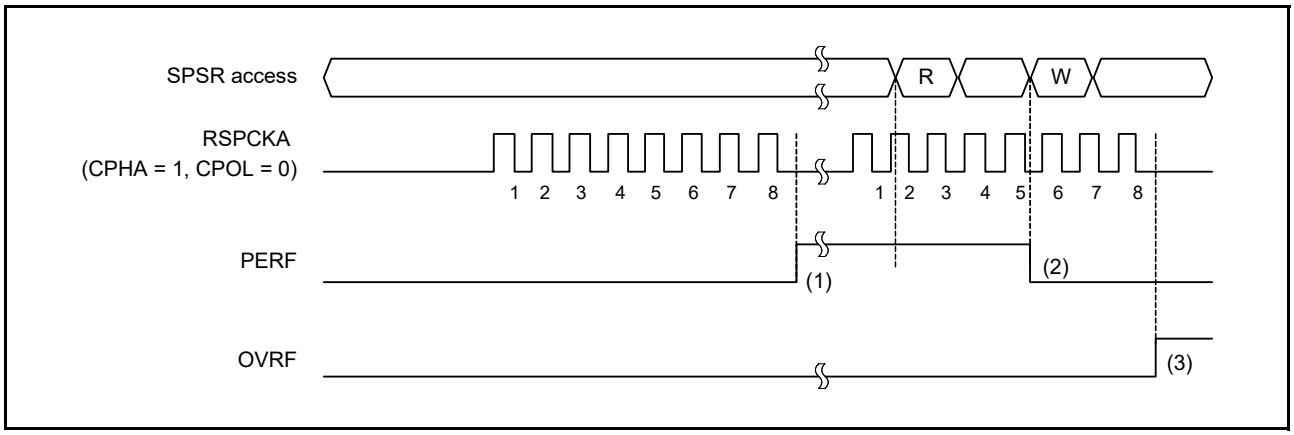


Figure 37.33 Operation Example of PERF Flag

The operation of the flags at the timing shown in steps (1) to (3) in the figure is described below.

(1) If a serial transfer terminates with the RSPI not detecting an overrun error, the RSPI copies the data in the shift register to the receive buffer. The RSPI judges the received data at this timing, and sets the PERF flag to 1 if a parity error is detected. In master mode, the RSPI copies the pointer value to the SPCMDm register to the SPSSR.SPECM[2:0] bits.

(2) If 0 is written to the PERF flag after the SPSR register is read when the PERF flag is 1, the PERF flag is set to 0.

(3) When the RSPI detects an overrun error and serial transfer is terminated, the data in the shift register is not copied to the receive buffer. The RSPI does not perform parity error detection at this timing.

The occurrence of a parity error can be checked either by reading the SPSR register or by using an error interrupt and reading the SPSR register. When executing a serial transfer, measures should be taken to ensure the early detection of parity errors, such as reading the SPSR register. When the RSPI is used in master mode, the pointer value to the SPCMDm register at the occurrence of the error can be checked by reading the SPSSR.SPECM[2:0] bits.

37.3.10.3 Mode Fault Error

The RSPI operates in multi-master mode when the SPCR.MSTR bit is 1, the SPCR.SPMS bit is 0, and the SPCR.MODFEN bit is 1. If the active level is input with respect to the SSLA0 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 SPSR.MODF flag to 1. Upon detecting the mode fault error, the RSPI copies the value of the pointer to the SPCMDm register to the SPSSR.SPECM[2:0] bits. The active level of the SSLA0 signal is determined by the SSLP.SSL0P bit.
When the MSTR bit is 0, the RSPI operates in slave mode. The RSPI detects a mode fault error if the MODFEN bit of the RSPI in slave mode is 1, and the SPMS bit is 0, and if the SSLA0 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 driving of the output signals and clears the SPCR.SPE bit to 0 (refer to section 37.3.11, Initializing RSPI). In the case of multi-master configuration, detection of a mode fault error is used to stop driving of the output signals and the RSPI function, which allows the master right to be released.
The occurrence of a mode fault error can be checked either by reading the SPSR register or by using an error interrupt and reading the SPSR register. Detecting mode fault errors without utilizing the error interrupt requires polling of the SPSR register. When using the RSPI in master mode, the pointer value to the SPCMDm register at the occurrence of the error can be checked by reading the SPSSR.SPECM[2:0] bits.
When the MODF flag is 1, writing of the value 1 to the SPE bit is ignored by the RSPI. To enable the RSPI function after the detection of a mode fault error, set the MODF flag to 0. When setting the MODF flag to 0, the SPE bit becomes 1.

37.3.10.4 Underrun Error

If a serial transfer is started when the SPCR.SPE bit is 1 (RSPI function is enabled) and transmit data is still not loaded on the shift register while RSPI operates in slave mode (the SPCR.MSTR bit is 0) and transmitter is enabled (the SPCR3.RXMD bit is 0), the RSPI detects an underrun error, and sets the SPSR.MODF and SPSR.UDRF flags to 1. Upon detecting an underrun error, the RSPI stops driving of the output signals and clears the SPCR.SPE bit to 0. Clearing of the SPE bit disables the RSPI function. (Refer to section 37.3.11, Initializing RSPI). The occurrence of an underrun error can be checked either by reading the SPSR register or by using an error interrupt and reading the SPSR register. Detecting underrun errors without utilizing the error interrupt requires polling of the SPSR register. When the MODF flag is 1, writing of the value 1 to the SPE bit is ignored by the RSPI. To enable the RSPI function after the detection of an underrun error, set the MODF flag to 0. When setting the MODF flag to 0, the SPE bit becomes 1.

37.3.11 Initializing RSPI

If 0 is written to the SPCR.SPE bit or the RSPI sets the SPE bit to 0 because of the detection of a mode fault error or an underrun error, the RSPI disables the RSPI function, and initializes some of the module functions. When a system reset is generated, the RSPI initializes all of the module functions. The following describes initialization by the clearing of the SPCR.SPE bit and initialization by a system reset.

37.3.11.1 Initialization by Clearing the SPE Bit

When the SPCR.SPE bit is set to 0, the RSPI performs the following initialization:

- Aborting the transmission and reception that is being executed
- Stopping the driving of output signals (Hi-Z) in slave mode
- Initializing the internal state of the RSPI
- Initializing the transmit buffer of the RSPI (Set the SPTEF flag to 1)

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 SPE bit is set to 1 again.
The SPSR.SPRF, SPCF, UDRF, PERF, MODF, and OVRF flags 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 and the communication status that was being executed before the initialization and the status of error occurrence during the RSPI transfer can be checked.
The transmit buffer is initialized to an empty state (the SPTEF flag is 1). Therefore, if the SPCR.SPTIE bit is set to 1 after RSPI initialization, a transmit buffer empty interrupt is generated. When the RSPI is initialized, in order to disable any transmit buffer empty interrupt, 0 should be written to the SPTIE bit simultaneously with the writing of 0 to the SPE bit.

37.3.11.2 System Reset

The initialization by a system reset 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 37.3.11.1, Initialization by Clearing the SPE Bit.

37.3.12 SPI Operation

37.3.12.1 Master Mode Operation

The only difference between single-master mode operation and multi-master mode operation lies in mode fault error detection (refer to section 37.3.10, 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-master mode and multi-master mode.

(1) Starting a Serial Transfer

The RSPI updates the data in the transmit buffer (SPTX) when data is written to the RSPI data register (SPDR) with the RSPI transmit buffer being empty (the SPTEF flag is 1 and data for the next transfer is not set). When the shift register is empty after the number of frames set in the SPDCR.SPFC[1:0] bits are written to the SPDR register, the RSPI copies data from the transmit buffer to the shift register and starts 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.
For details on the RSPI transfer format, refer to section 37.3.5, Transfer Format. The polarity of the SSLAi output pins depends on the SSLP register settings.

(2) Terminating a Serial Transfer

Irrespective of the SPCMDm.CPHA bit, the RSPI terminates the serial transfer after transmitting an RSPCKA edge corresponding to the final sampling timing. If free space is available in the receive buffer (SPRX) (the SPRF flag is 0), upon termination of serial transfer, the RSPI copies data from the shift register to the receive buffer of the SPDR register. 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 SPCMDm.SPB[3:0] bit setting. The polarity of the SSLAi output pin depends on the SSLP register settings.
For details on the RSPI transfer format, refer to section 37.3.5, Transfer Format.

(3) Sequence Control

The transfer format that is employed in master mode is determined by the SPSCR, SPCMDm, SPBR, SPCKD, SSLND, and SPND registers.

The SPSCR register is a register used to determine the sequence configuration for serial transfers that are executed by the RSPI in master mode. The following items are set in the SPCMDm register: SSLAi pin output signal value, MSB/LSB first, data length, some of the bit rate settings, RSPCK polarity/phase, whether the SPCKD register is to be referenced, whether the SSLND register is to be referenced, and whether the SPND register is to be referenced. The SPBR register holds some of the bit rate settings; the SPCKD register, an RSPI clock delay value; the SSLND register, an SSL negation delay; and the SPND register, a next-access delay value.

According to the sequence length that is assigned to the SPSCR register, the RSPI makes up a sequence comprised of a part or all of the SPCMDm register. The RSPI contains a pointer to the SPCMDm register that makes up the sequence. The value of this pointer can be checked by reading the SPSSR.SPCP[2:0] bits. When the SPCR.SPE bit is set to 1 and the RSPI function is enabled, the RSPI loads the pointer to the commands in the SPCMD0 register, 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 the SPCMD0 register, and in this manner the sequence is executed repeatedly.

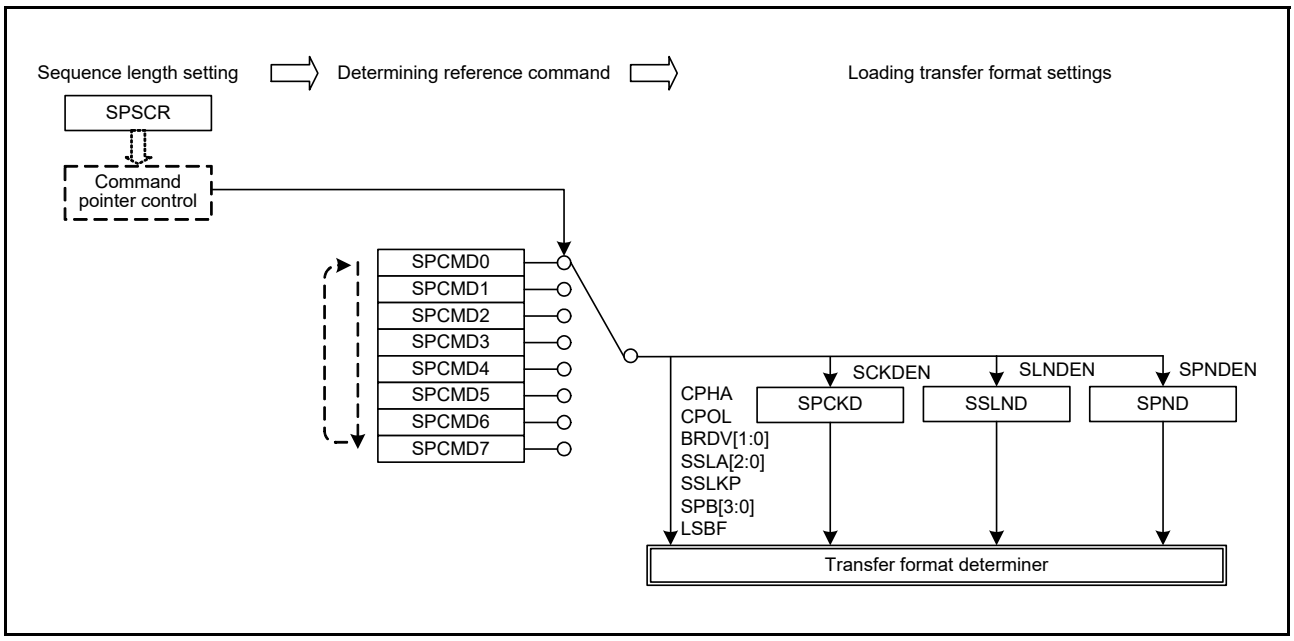


Figure 37.34 Procedure for Determining the Form of Serial Transfer in Master Mode

In this section, a frame is the combination of the data (SPDR) and the settings (SPCMDm).

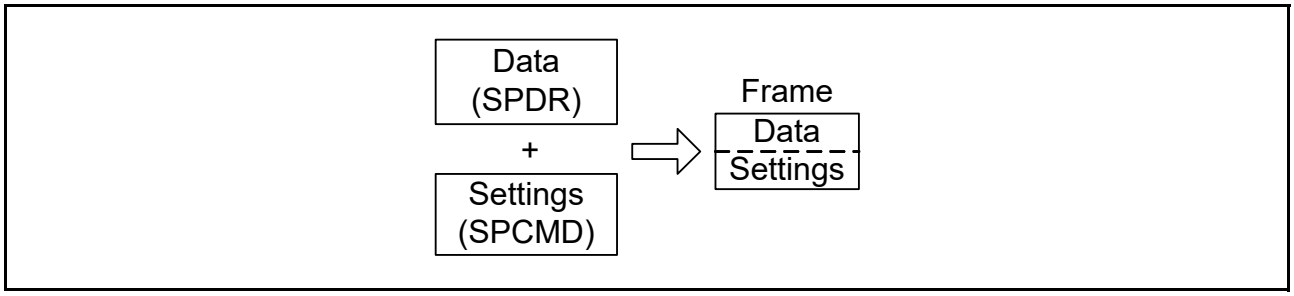


Figure 37.35 Concept of a Frame

Figure 37.36 shows the relationship between the command and the transmit and receive buffers in the sequence of operations specified by the settings in Table 37.4.

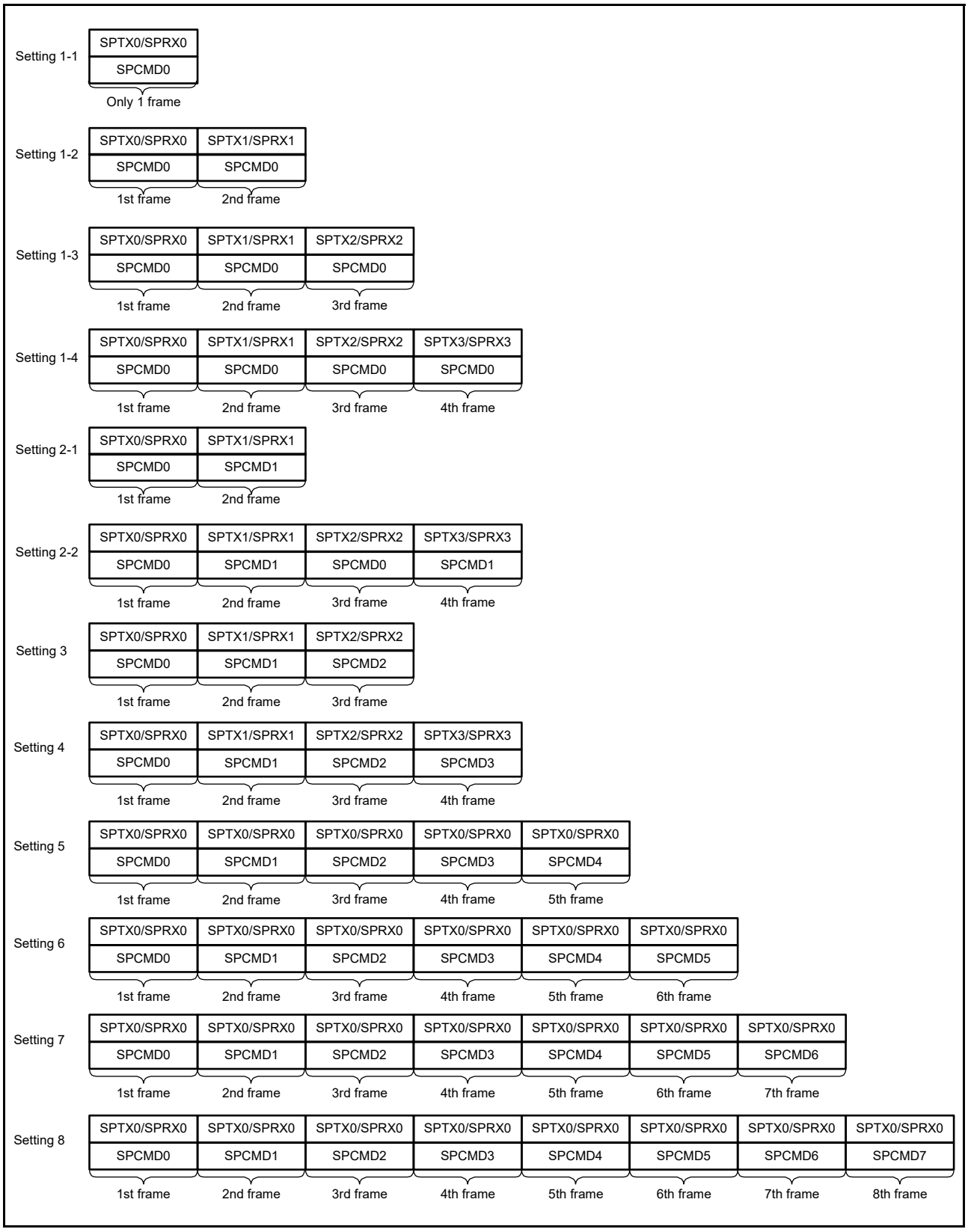


Figure 37.36 Correspondence between the RSPI Command Register and Transmit/Receive Buffers in Sequence Operations

(4) Burst Transfer

If the SPCMDm.SSLKP bit that the RSPI references during the current serial transfer is 1, the RSPI keeps the SSLAi signal level during the serial transfer until the beginning of the SSLAi signal assertion for the next serial transfer. If the SSLAi signal level for the next serial transfer is the same as the SSLAi signal level for the current serial transfer, the RSPI can execute continuous serial transfers while keeping the SSLAi signal assertion status (burst transfer).
Figure 37.37 shows an example of an SSLAi signal operation for the case where a burst transfer is implemented using SPCMD0 and SPCMD1 register settings. The text below explains the RSPI operations (1) to (7) as shown in Figure 37.37. It should be noted that the polarity of the SSLAi output signal depends on the SSLP register settings.

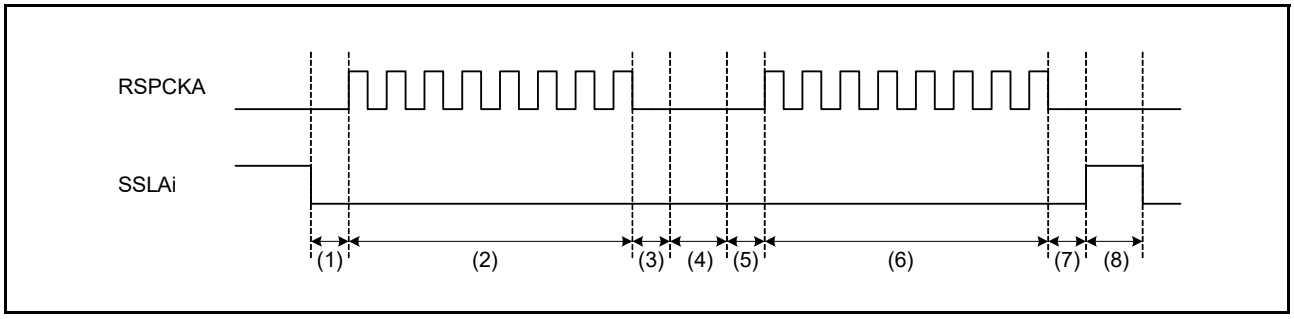


Figure 37.37 Example of Burst Transfer Operation Using SSLKP Bit (CPHA = 1, CPOL = 0)

(1) Based on the SPCMD0 register, the RSPI asserts the SSLAi signal and inserts RSPCK delays.
(2) The RSPI executes serial transfers according to the SPCMD0 register.
(3) The RSPI inserts SSL negation delays.
(4) Since the SPCMD0.SSLKP bit is 1, the RSPI keeps the SSLAi signal value on the SPCMD0 register. This period is sustained, at the shortest, for a period equal to the next-access delay of the SPCMD0 register. If the shift register is empty after the passage of a minimum period, this period is sustained until the transmit data is stored in the shift register for the next transfer.
(5) Based on the SPCMD1 register, the RSPI inserts RSPCK delays.
(6) The RSPI executes serial transfers according to the SPCMD1 register.
(7) The RSPI inserts SSL negation delays.
(8) Because the SPCMD1.SSLKP bit is 0, the RSPI negates the SSLAi signal. In addition, a next-access delay is inserted according to the SPCMD1 register.

If the SSLAi signal output settings in the SPCMDm register in which 1 is assigned to the SSLKP bit are different from the SSLAi signal output settings in the SPCMDm register to be used in the next transfer, the RSPI switches the SSLAi signal status to SSLAi signal assertion ((5) in Figure 37.37) corresponding to the command for the next transfer. Note that if such an SSLAi signal switching occurs, the slaves that drive the MISOA signal compete, and collision of signal levels may occur.
The RSPI in master mode references the SSLAi signal operation within the module for the case where the SSLKP bit is not used. Even when the SPCMDm.CPHA bit is 0, the RSPI can accurately start serial transfers by using the SSLAi signal assertion for the next transfer that is detected internally.
When the SPCR3.SCKDDIS bit is set to 1 (Does not insert delays between data bytes during burst transfer), the delays described in (3) to (5) above are not inserted and only delay of 0.5 cycles of RSPCK is inserted.

(5) RSPCK Delay (t1)

The RSPCK delay value in master mode depends on the SPCMDm.SCKDEN bit setting and the SPCKD register setting. The RSPI determines the SPCMDm register to be referenced during serial transfer by pointer control, and determines an RSPCK delay value during serial transfer by using the SPCMDm.SCKDEN bit and the SPCKD register, as listed in Table 37.8. For a definition of RSPCK delay, refer to section 37.3.5, Transfer Format.
When the SPCMDm.SSLKP bit is set to 1 (burst transfer) and the SPCR3.SCKDDIS bit is set to 1 (Does not insert delays between data bytes during burst transfer), the RSPCK delay is inserted only in the first frame.

Table 37.8 Relationship among SCKDEN Bit, SPCKD, and RSPCK Delay Value

SPCMDm.SCKDEN Bit	SPCKD.SCKDL[2:0] Bits	RSPCK Delay Value
0	000b to 111b	1 RSPCK
1	000b	1 RSPCK
	001b	2 RSPCK
	010b	3 RSPCK
	011b	4 RSPCK
	100b	5 RSPCK
	101b	6 RSPCK
	110b	7 RSPCK
	111b	8 RSPCK

(6) SSL Negation Delay (t2)

The SSL negation delay value in master mode depends on the SPCMDm.SLNDEN bit setting and the SSLND register setting. The RSPI determines the SPCMDm register to be referenced during serial transfer by pointer control, and determines an SSL negation delay value during serial transfer by using the SPCMDm.SLNDEN bit and the SSLND register, as listed in Table 37.9. For a definition of SSL negation delay, refer to section 37.3.5, Transfer Format.
When the SPCMDm.SSLKP bit is set to 1 (burst transfer) and the SPCR3.SCKDDIS bit is set to 1 (Does not insert delays between data bytes during burst transfer), the SSL negation delay is inserted only in the last frame.

Table 37.9 Relationship among SLNDEN Bit, SSLND, and SSL Negation Delay Value

SPCMDm.SLNDEN Bit	SSLND.SLNDL[2:0] Bits	SSL Negation Delay Value
0	000b to 111b	1 RSPCK
1	000b	1 RSPCK
	001b	2 RSPCK
	010b	3 RSPCK
	011b	4 RSPCK
	100b	5 RSPCK
	101b	6 RSPCK
	110b	7 RSPCK
	111b	8 RSPCK

(7) Next-Access Delay (t3)

The next-access delay value in master mode depends on the SPCMDm.SPNDEN bit setting and the SPND setting. The RSPI determines the SPCMDm register to be referenced during serial transfer by pointer control, and determines a next-access delay value during serial transfer by using the SPCMDm.SPNDEN bit and SPND, as listed in Table 37.10. For a definition of next-access delay, refer to section 37.3.5, Transfer Format.

When the SPCMDm.SSLKP bit is set to 1 (burst transfer) and the SPCR3.SCKDDIS bit is set to 1 (Does not insert delays between data bytes during burst transfer), the next-access delay is inserted only in the last frame.

Table 37.10 Relationship among SPNDEN Bit, SPND, and Next-Access Delay Value

SPCMDm.SPNDEN Bit	SPND.SPNDL[2:0] Bits	Next-Access Delay Value
0	000b to 111b	1 RSPCK + 2 PCLK
1	000b	1 RSPCK + 2 PCLK
	001b	2 RSPCK + 2 PCLK
	010b	3 RSPCK + 2 PCLK
	011b	4 RSPCK + 2 PCLK
	100b	5 RSPCK + 2 PCLK
	101b	6 RSPCK + 2 PCLK
	110b	7 RSPCK + 2 PCLK
	111b	8 RSPCK + 2 PCLK

(8) Initialization Flowchart

Figure 37.38 is a flowchart illustrating an example of initialization in SPI operation when the RSPI is used in master mode. For a description of how to set up the interrupt controller, DMAC, and I/O ports, refer to the descriptions given in the individual blocks.

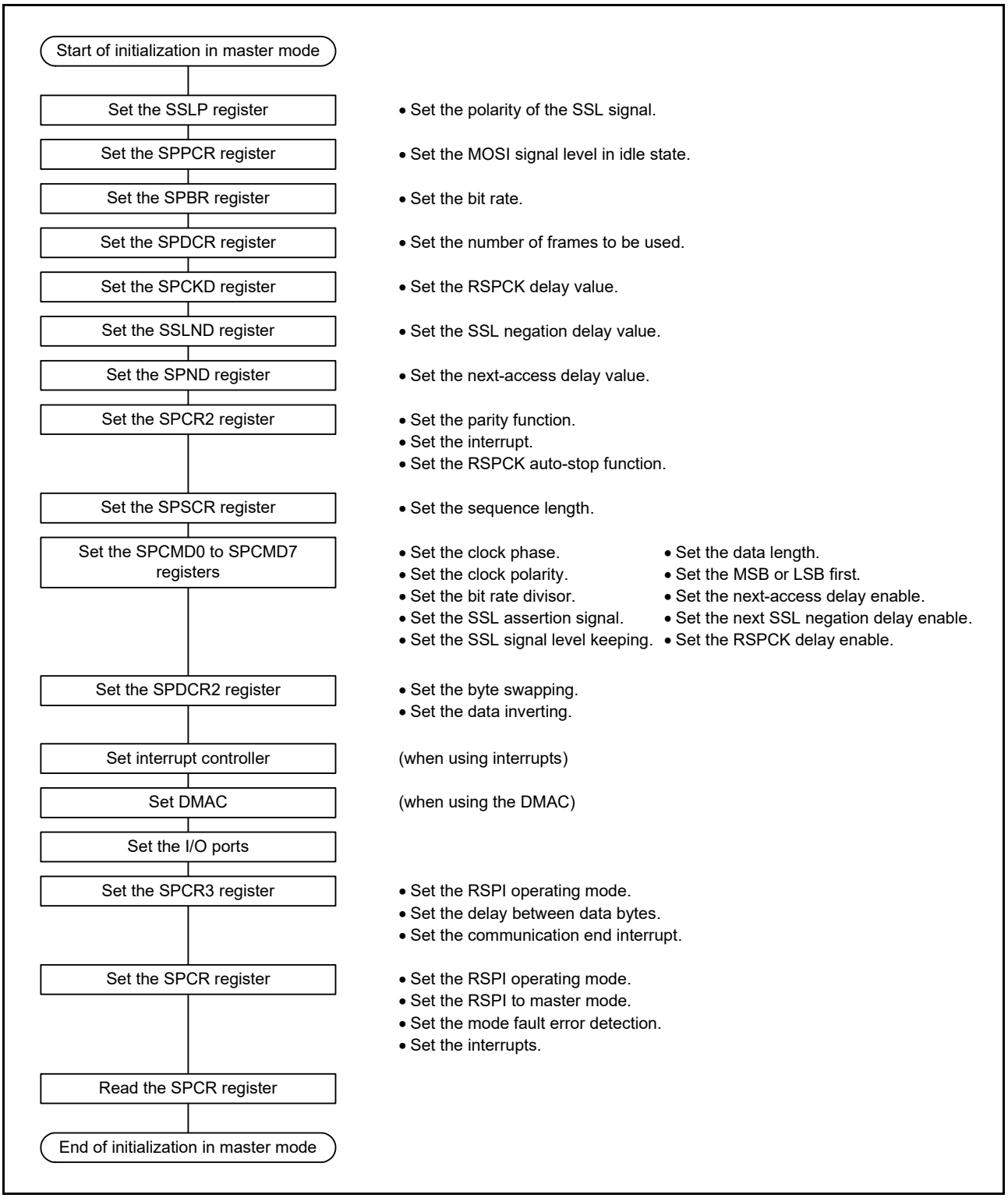


Figure 37.38 Example of Initialization Flowchart in Master Mode (SPI Operation)

(9) Software Processing Flow

Figure 37.40 to Figure 37.42 show examples of the flow of software processing.

(a) Communication Preprocessing Flow

Before starting communications, clear the error flags and disable the idle interrupt and communication end interrupt. Then enable the RSPI function and the required interrupts at the same time.

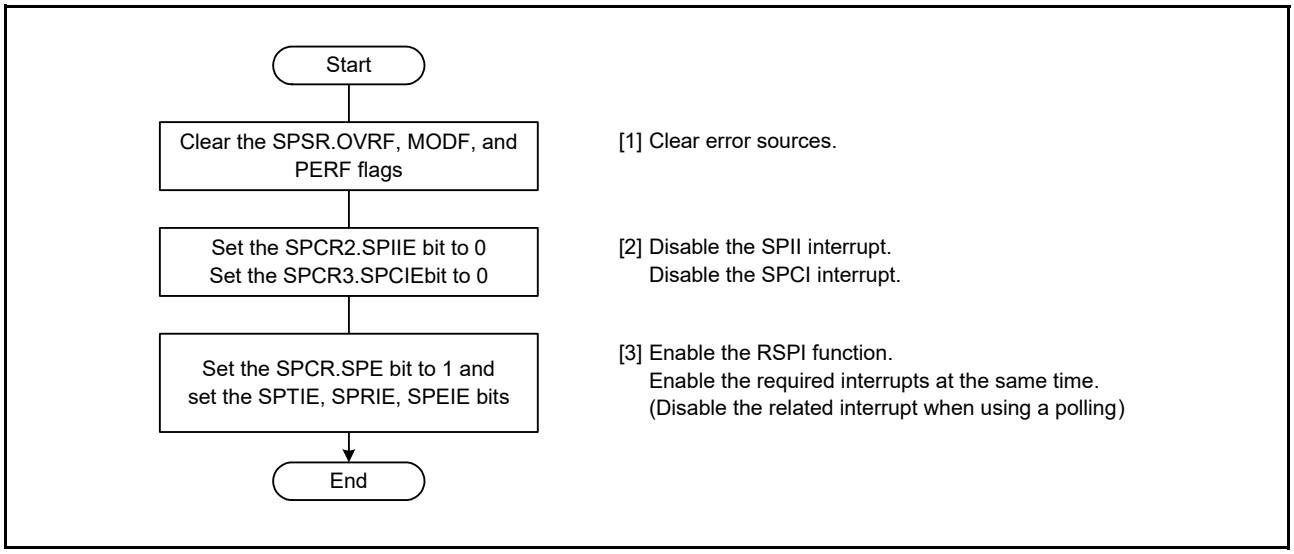


Figure 37.39 Flowchart in Master Mode (Communication Preprocess)

(b) Transmit Processing Flow

When transmitting data, the CPU will be notified of the completion of data transmission by enabling the SPII or SPCI interrupt after the last writing of data for transmission.

The completion of data transmission can also be checked by polling to see if the SPSR.IDLNF flag has become 0 or the SPCF flag has become 1, instead of using the SPII or SPCI interrupt. However, one cycle of PCLK is required for the time from when data for transmission is written in the SPDR register to when the IDLNF flag becomes 1 and the SPCF flag becomes 0. After the last data is written in the SPDR register, discard the value of the SPSR register once not to judge the condition with the IDLNF flag which has not yet become 1 or the SPCF flag which has not yet become 0, and read and use the value of the IDLNF or SPCF flag to confirm the completion of data transmission.

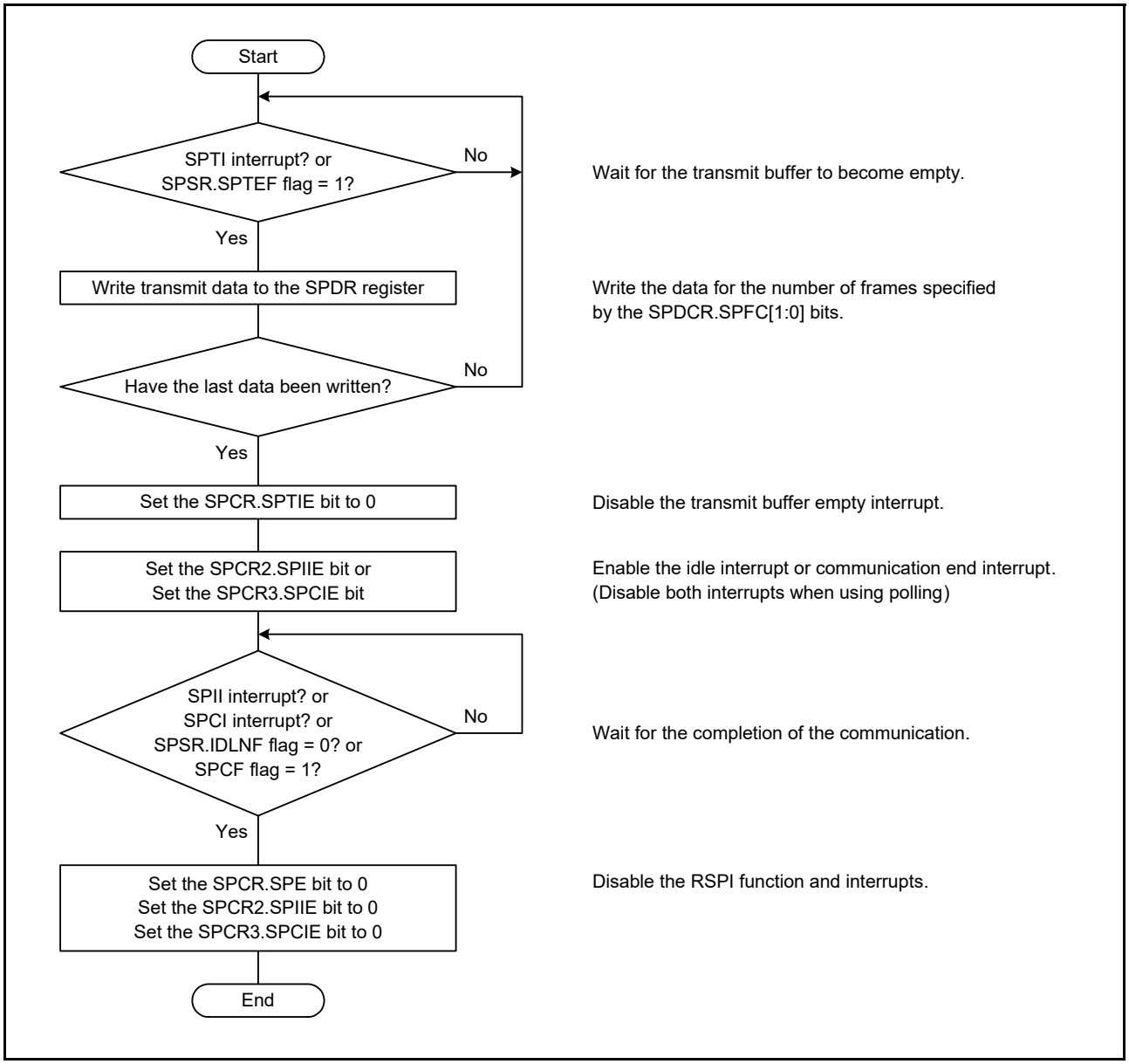


Figure 37.40 Flowchart in Master Mode (Transmission)

(c) Receive Processing Flow

The RSPI does not support receive-only simplex communications in master mode, so processing for transmission is required.

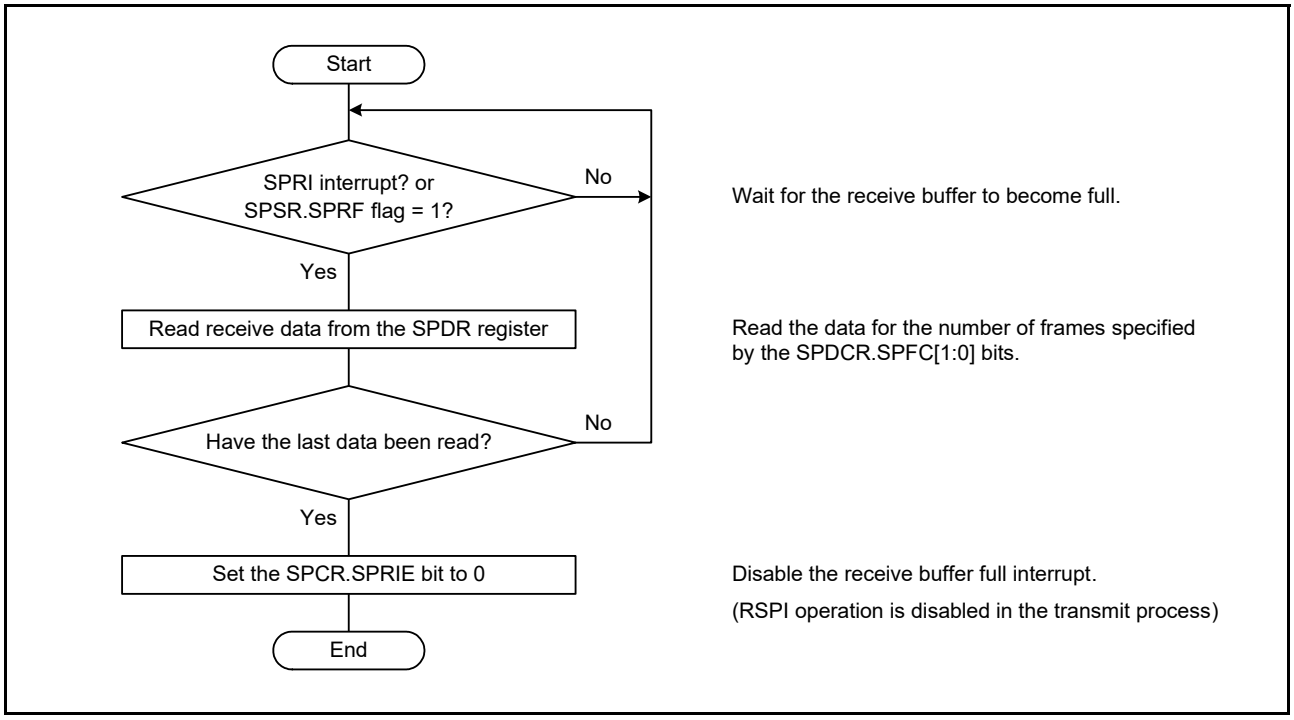


Figure 37.41 Flowchart in Master Mode (Reception)

(d) Flow of Error Processing

When a mode fault error is generated, the SPCR.SPE bit is automatically cleared, stopping operations for transmission and reception. For errors from other sources, however, the SPCR.SPE bit is not cleared and operations for transmission and reception continue; accordingly, we recommend clearing of the SPCR.SPE bit to stop operations in the case of errors other than mode fault errors. Not doing so will lead to updating of the SPSSR.SPECM[2:0] bits.
When interrupts are used and an error occurs, if the ICU.IRn.IR flag for the SPTI or SPRI interrupt request is set to 1, clear the ICU.IRn.IR flag in the error processing routine. If the SPRI interrupt request is indicated, read the receive buffer and initialize the sequencer in the RSPI.

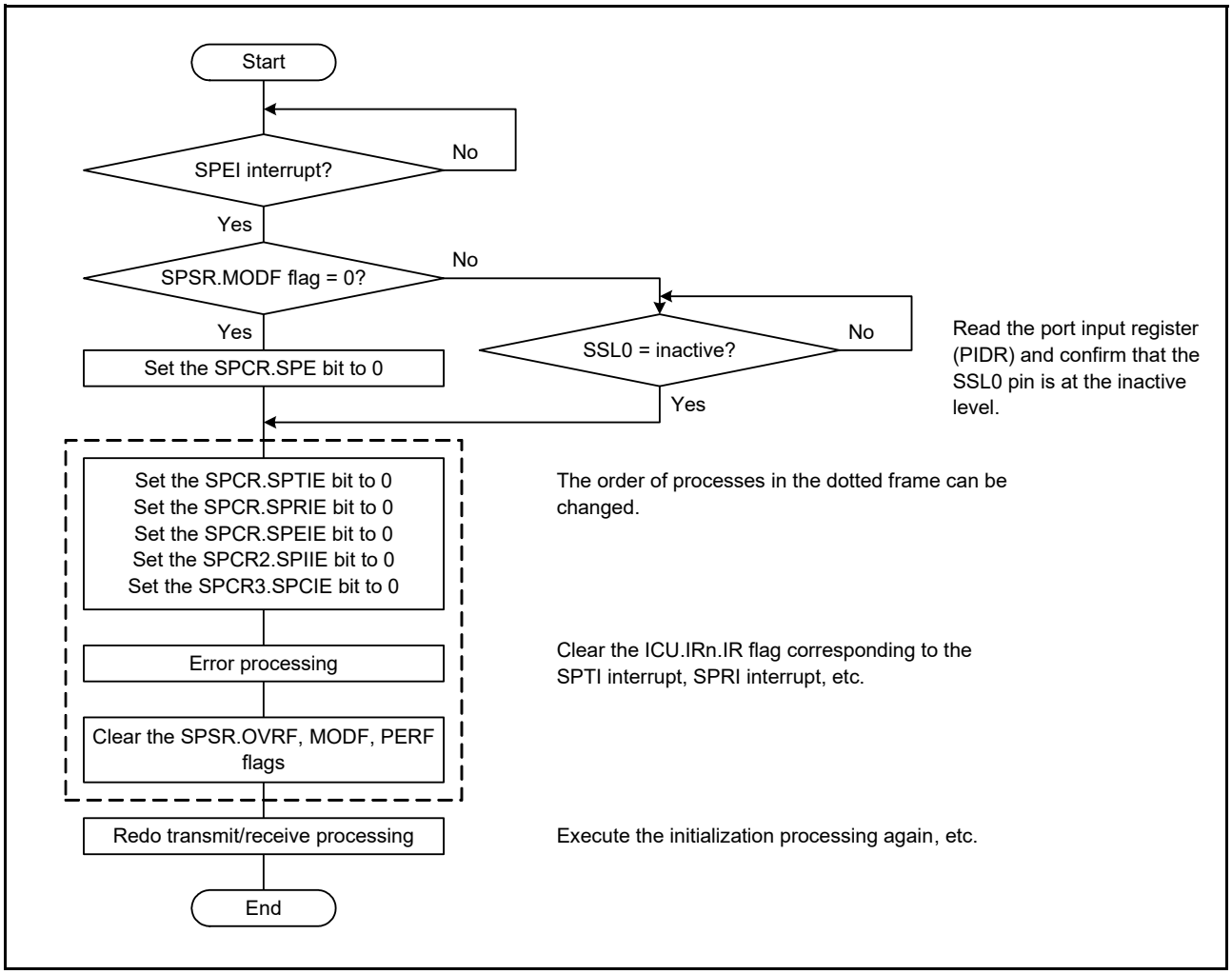


Figure 37.42 Flowchart for Master Mode (Error Processing)

37.3.12.2 Slave Mode Operation

(1) Starting a Serial Transfer

If the SPCMD0.CPHA bit is 0, when detecting an SSLA0 input signal assertion, the RSPI needs to start driving valid data to the MISOA output signal. For this reason, when the CPHA bit is 0, the assertion of the SSLA0 input signal triggers the start of a serial transfer.
If the CPHA bit is 1, when detecting the first RSPCKA edge in an SSLA0 signal asserted condition, the RSPI needs to start driving valid data to the MISOA output signal. For this reason, when the CPHA bit is 1, the first RSPCKA edge in an SSLA0 signal asserted condition triggers the start of a serial transfer.
Irrespective of the CPHA bit setting, the timing at which the RSPI starts driving of the MISOA output signal is the SSLA0 signal assertion timing. The data which is output by the RSPI is either valid or invalid, depending on the CPHA bit setting.
For details on the RSPI transfer format, refer to section 37.3.5, Transfer Format. The polarity of the SSLA0 input signal depends on the setting of the SSLP.SSL0P bit.

(2) Terminating a Serial Transfer

Irrespective of the SPCMD0.CPHA bit, the RSPI terminates the serial transfer after detecting an RSPCKA edge corresponding to the final sampling timing. When free space is available in the receive buffer (the SPRF flag is 0), upon termination of serial transfer the RSPI copies received data from the shift register to the receive buffer of the SPDR register. Upon termination of a serial transfer the RSPI changes the status of the shift register to “empty”, regardless of the receive buffer state. A mode fault error occurs if the RSPI detects an SSLA0 input signal negation from the beginning of serial transfer to the end of serial transfer (refer to section 37.3.10, Error Detection).
The final sampling timing changes depending on the bit length of transfer data. In slave mode, the RSPI data length depends on the SPCMD0.SPB[3:0] bit setting. The polarity of the SSLA0 input signal depends on the SSLP.SSL0P bit setting.
For details on the RSPI transfer format, refer to section 37.3.5, Transfer Format.

(3) Notes on Single-Slave Operations

If the SPCMD0.CPHA bit is 0, the RSPI starts serial transfers when it detects the assertion edge for an SSLA0 input signal. In the type of configuration shown in Figure 37.7 as an example, if the RSPI is used in single-slave mode, the SSLA0 signal is fixed at the active state. Therefore, when the CPHA bit is set to 0, the RSPI cannot correctly start a serial transfer. To correctly execute transmit/receive operations by the RSPI in slave mode in a configuration in which the SSLA0 input signal is fixed at the active state, the CPHA bit should be set to 1. If there is a need for setting the CPHA bit to 0, the SSLA0 input signal should not be fixed.

(4) Burst Transfer

If the SPCMD0.CPHA bit is 1, continuous serial transfer (burst transfer) can be executed while retaining the assertion state for the SSLA0 input signal. If the CPHA bit is 1, the period from the first RSPCKA edge to the sampling timing for the reception of the final bit in an SSLA0 signal active state corresponds to a serial transfer period. Even when the SSLA0 input signal remains at the active level, the RSPI can accommodate burst transfers because it can detect the start of an access.
If the CPHA bit is 0, the second and subsequent serial transfers during burst transfer cannot be executed correctly.

(5) Initialization Flowchart

Figure 37.43 is a flowchart illustrating an example of initialization in SPI operation when the RSPI is used in slave mode. For a description of how to set up the interrupt controller, DMAC, and I/O ports, refer to the descriptions given in the individual blocks.

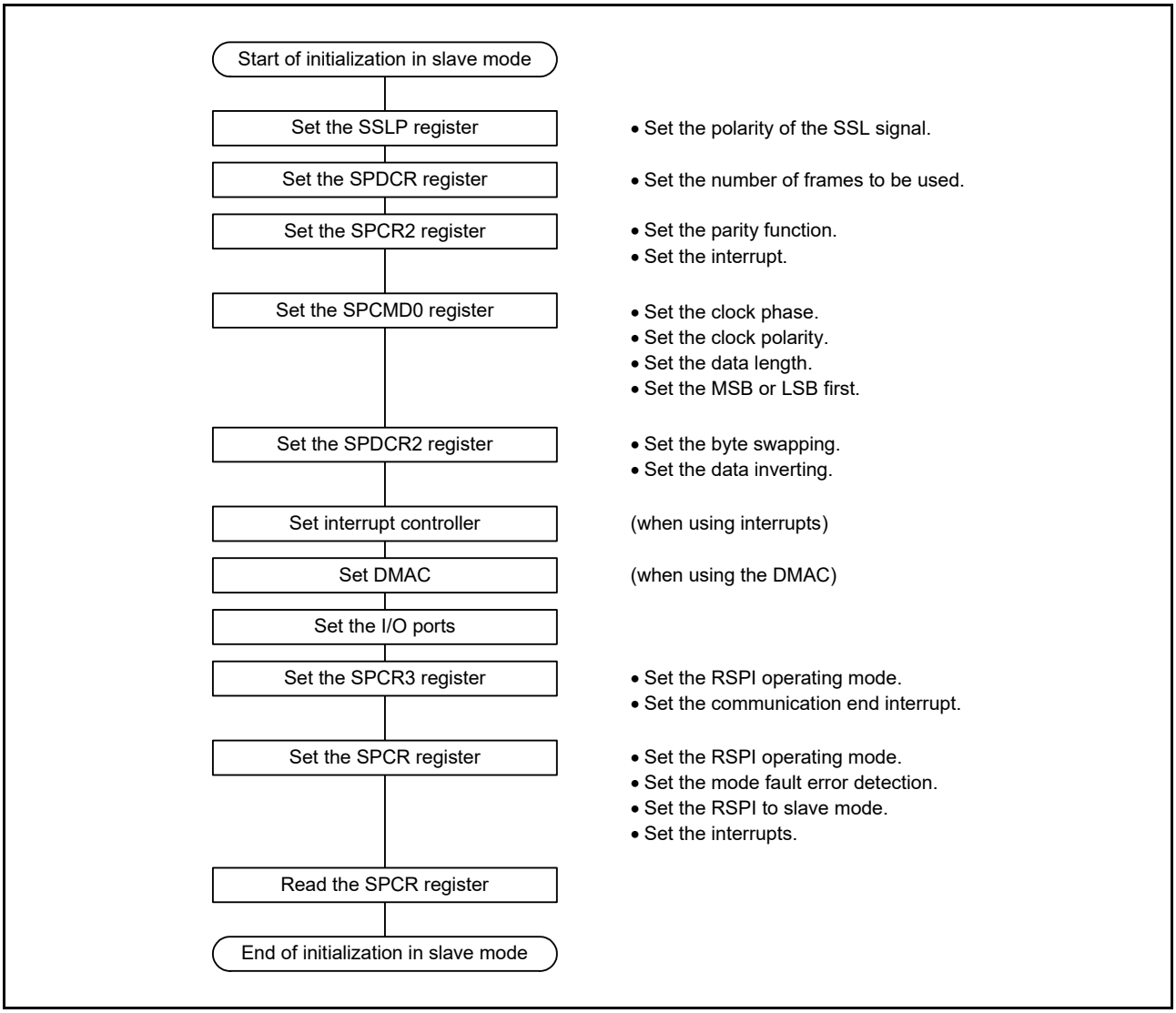


Figure 37.43 Example of Initialization Flowchart in Slave Mode (SPI Operation)

(6) Software Processing Flow

Figure 37.45 to Figure 37.47 show examples of the flow of software processing.

(a) Communication Preprocessing Flow

Before starting communications, clear the error flags and disable the idle interrupt and communication end interrupt. Then enable the RSPI function and the required interrupts at the same time.

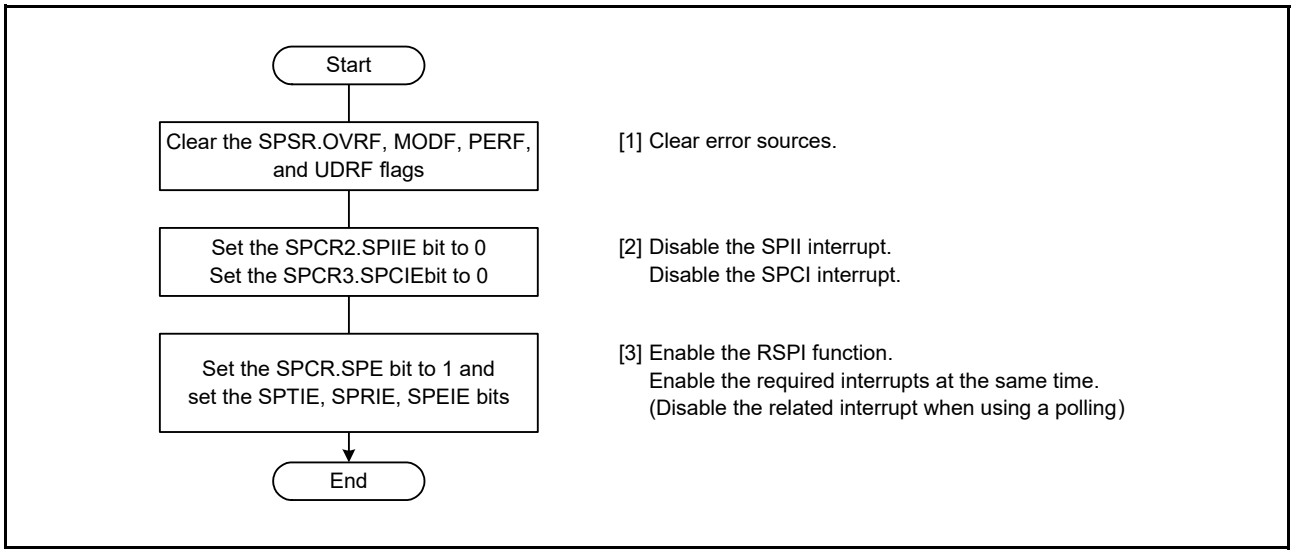


Figure 37.44 Flowchart in Slave Mode (Communication Preprocess)

(b) Transmit Processing Flow

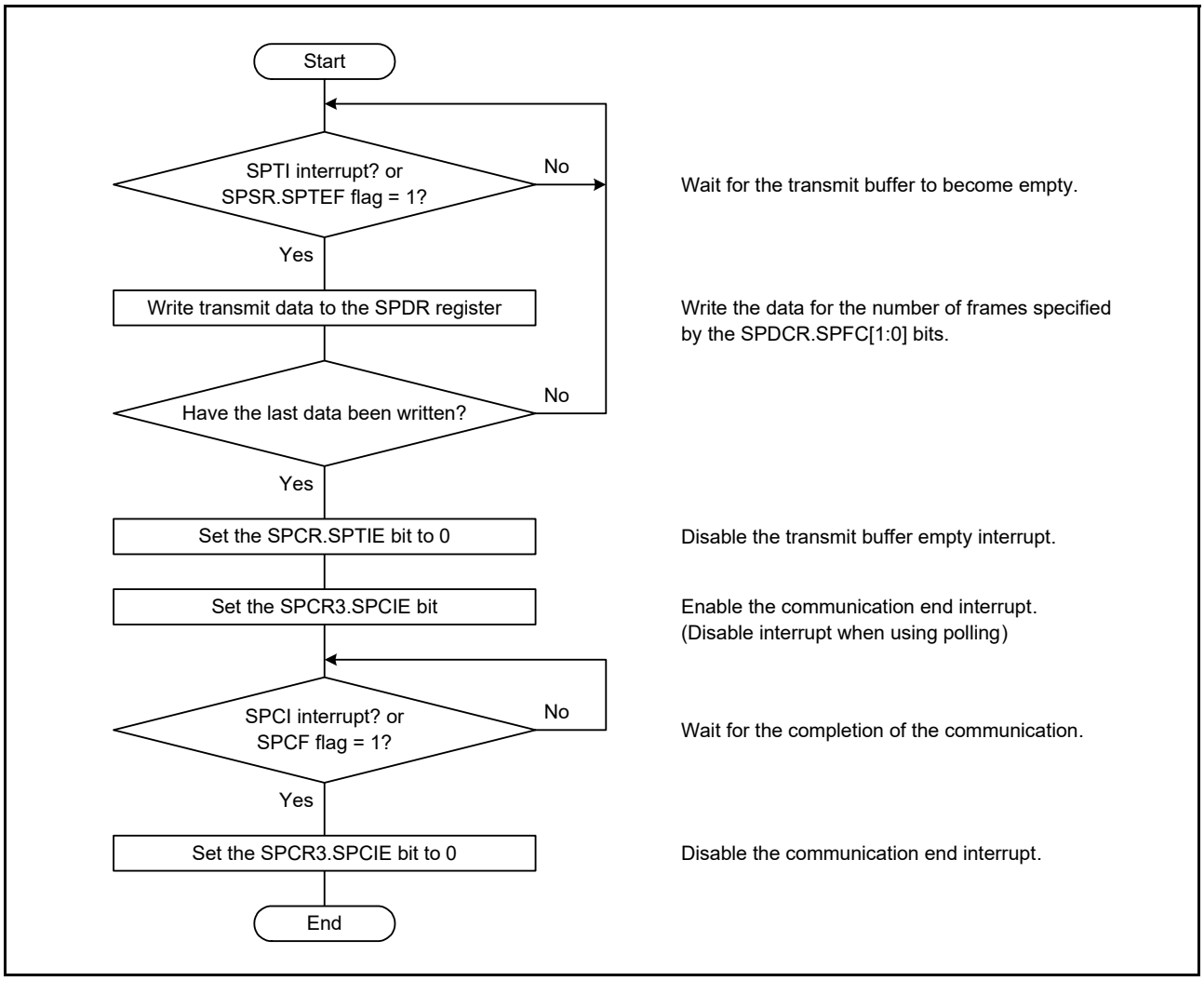


Figure 37.45 Flowchart in Slave Mode (Transmission)

(c) Receive Processing Flow

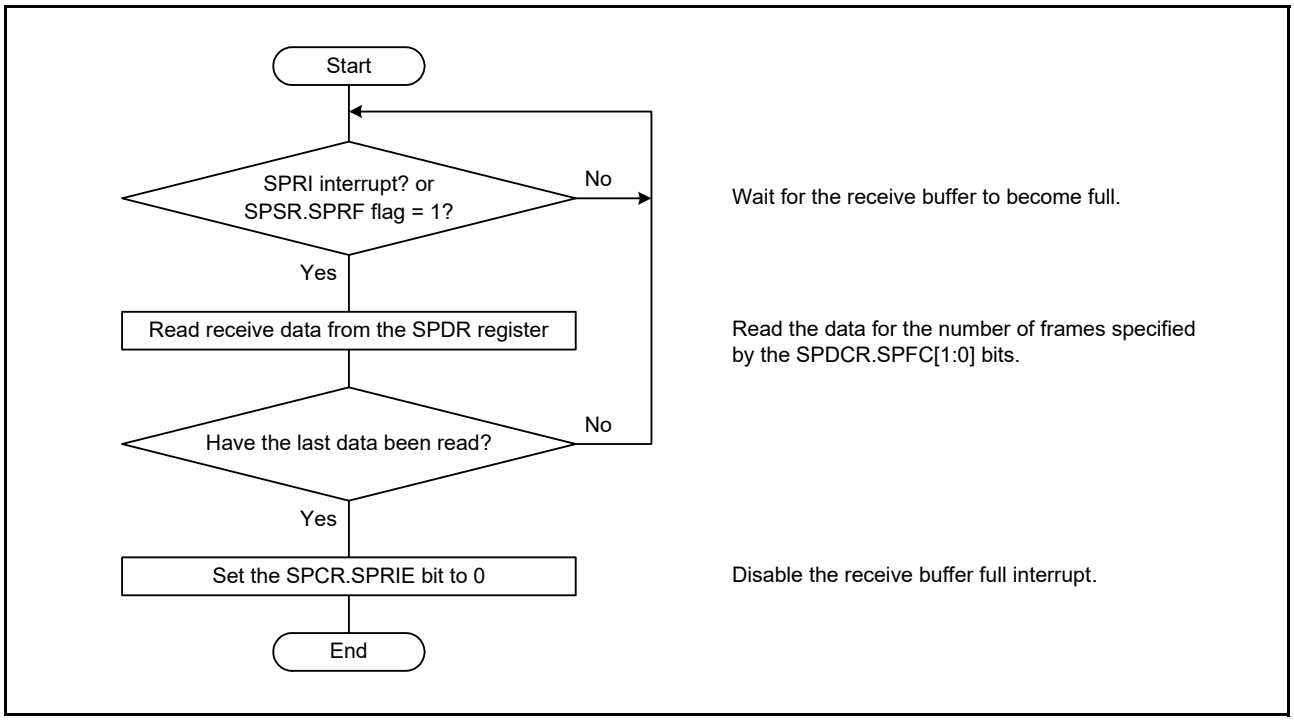


Figure 37.46 Flowchart in Slave Mode (Reception)

(d) Flow of Error Processing

In slave mode, even when a mode fault error is generated, the SPSR.MODF flag can be cleared regardless of the status of the SSLA0 pin.

When interrupts are used and an error occurs, if the ICU.IRn.IR flag for the SPTI or SPRI interrupt request is set to 1, clear the ICU.IRn.IR flag in the error processing routine. If the SPRI interrupt request is indicated, read the receive buffer and initialize the sequencer in the RSPI.

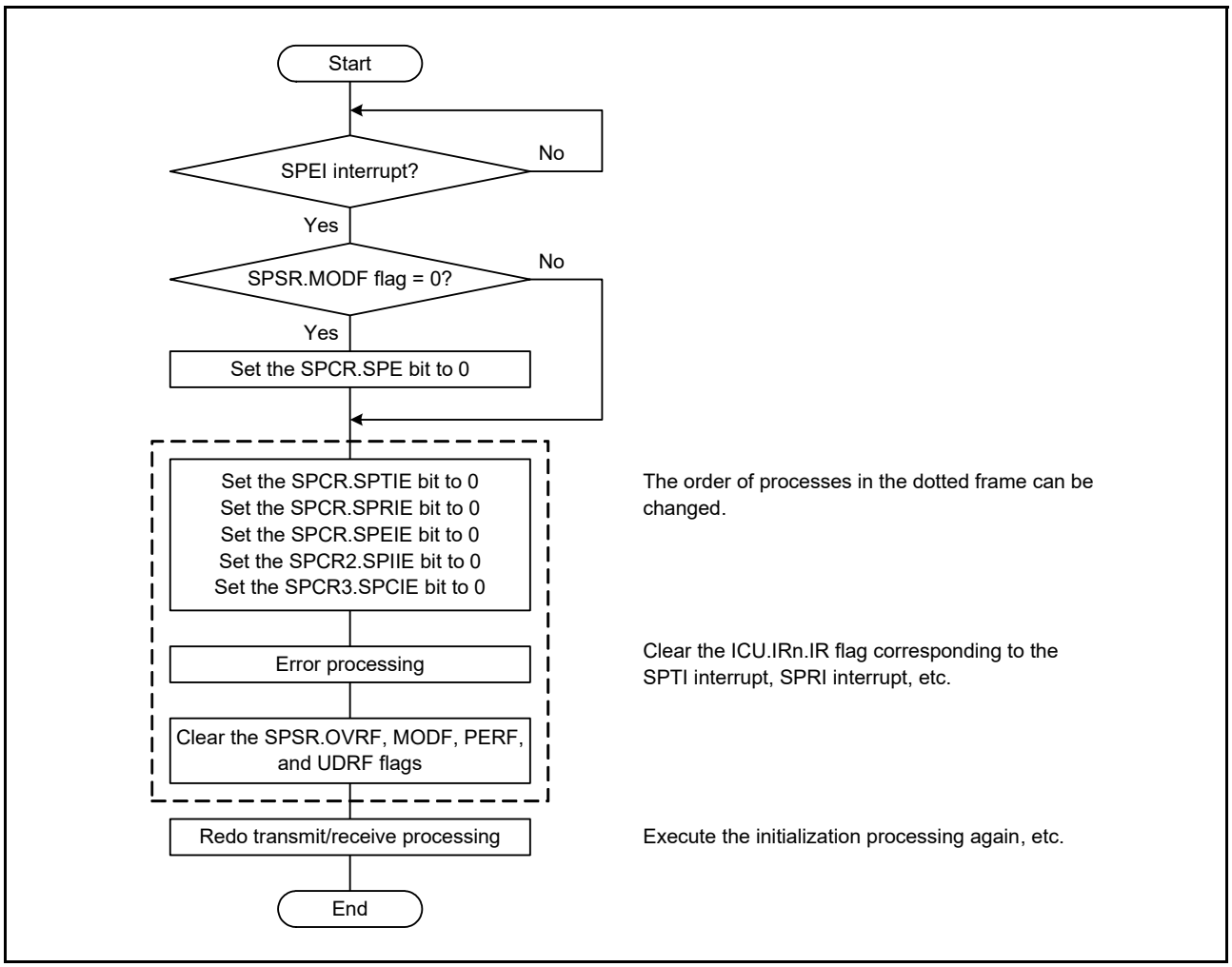


Figure 37.47 Flowchart for Slave Mode (Error Processing)

37.3.13 Clock Synchronous Operation

Setting the SPCR.SPMS bit to 1 selects clock synchronous operation of the RSPI. In clock synchronous operation, the SSLAi pin is not used, and the three pins of RSPCKA, MOSIA, and MISOA handle communications. The SSLAi pin is available as I/O port pins.
Although clock synchronous operation does not require use of the SSLAi pin, operation of the module is the same as in SPI operation. That is, in both master and slave mode operations, communications can be performed with the same flow as in SPI operation. However, mode fault errors are not detected because the SSLAi pin is not used.
Furthermore, do not set the SPCMDm.CPHA bit to 0 if clock synchronous operation is to proceed in slave mode (SPCR.MSTR = 0).

37.3.13.1 Master Mode Operation

(1) Starting a Serial Transfer

The RSPI updates the data in the transmit buffer (SPTX) of the SPDR register when data is written to the SPDR register with the transmit buffer being empty (the SPTEF flag is 1 and data for the next transfer is not set). When the shift register is empty after the number of frames set in the SPDCR.SPFC[1:0] bits are written to the SPDR register, the RSPI copies data from the transmit buffer to the shift register and starts serial transmission. 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.
For details on the RSPI transfer format, refer to section 37.3.5, Transfer Format.
However, transfer in clock synchronous operation is conducted without the SSLA0 output signal.

(2) Terminating a Serial Transfer

The RSPI terminates the serial transfer after transmitting an RSPCKA edge corresponding to the sampling timing. If free space is available in the receive buffer (SPRX) (the SPRF flag is 0), 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 SPCMDm.SPB[3:0] bit setting.
For details on the RSPI transfer format, refer to section 37.3.5, Transfer Format.
However, transfer in clock synchronous operation is conducted without the SSLA0 output signal.

(3) Sequence Control

The transfer format employed in master mode is determined by SPSCR, SPCMDm, SPBR, SPCKD, SSLND, and SPND registers. Although the SSLAi signals are not output in clock synchronous operation, these settings are valid.
The SPSCR register is a register used to determine the sequence configuration for serial transfers that are executed by the RSPI in master mode. The following items are set in the SPCMDm register: SSLAi output signal value, MSB/LSB first, data length, some of the bit rate settings, RSPCKA polarity/phase, whether the SPCKD register is to be referenced, whether the SSLND register is to be referenced, and whether the SPND register is to be referenced. The SPBR register holds some of the bit rate settings; the SPCKD register, an RSPI clock delay value; the SSLND register, an SSL negation delay; and the SPND register, a next-access delay value.
According to the sequence length that is assigned to the SPSCR register, the RSPI makes up a sequence comprised of a part or all of the SPCMDm register. The RSPI contains a pointer to the SPCMDm register that makes up the sequence. The value of this pointer can be checked by reading the SPSSR.SPCP[2:0] bits. When the SPCR.SPE bit is set to 1 and the RSPI function is enabled, the RSPI loads the pointer to the commands in the SPCMD0 register, and incorporates the SPCMD0 register setting 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 the SPCMD0 register, and in this manner the sequence is executed repeatedly.

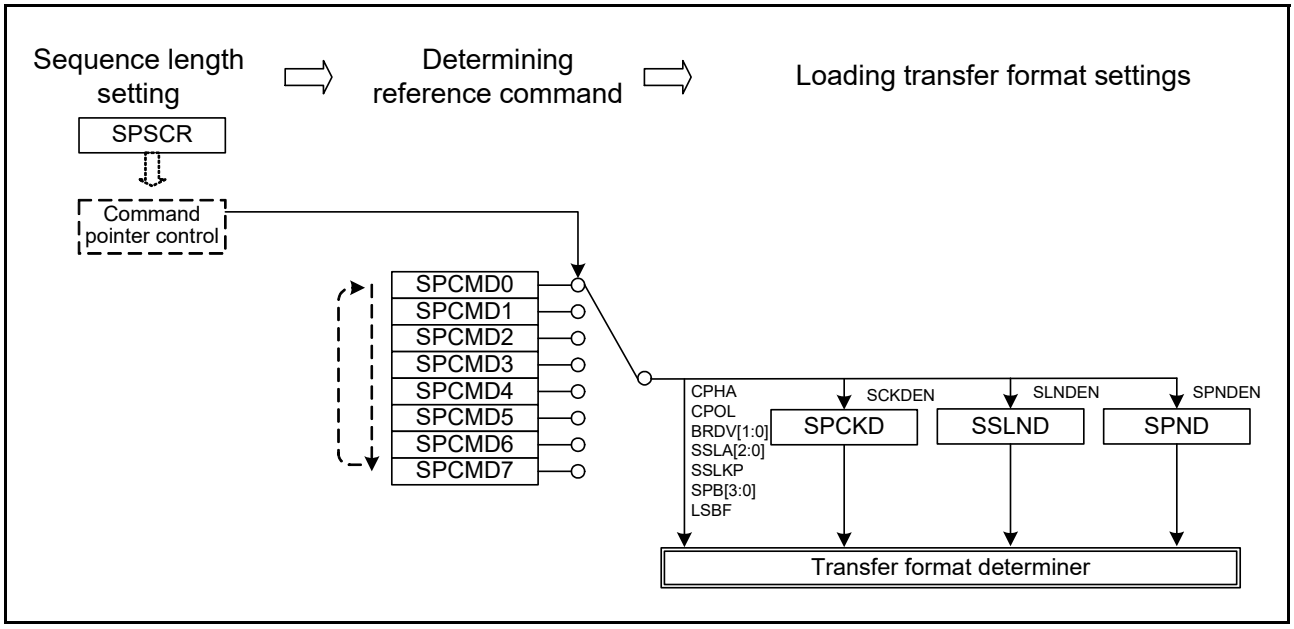


Figure 37.48 Procedure for Determining the Form of Serial Transmission in Master Mode

In this section, a frame is the combination of the data (SPDR) and the settings (SPCMDm).

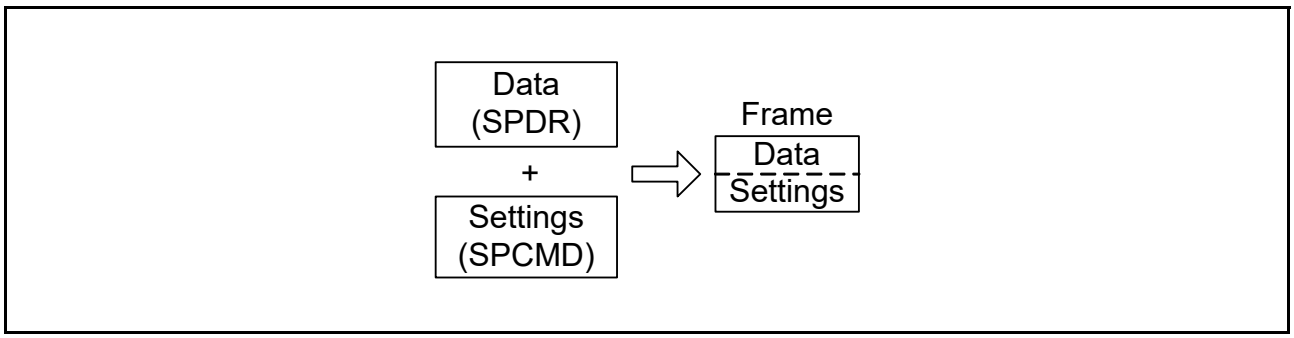


Figure 37.49 Concept of a Frame

Figure 37.50 shows the relationship between the command and the transmit and receive buffers in the sequence of operations specified by the settings in Table 37.4.

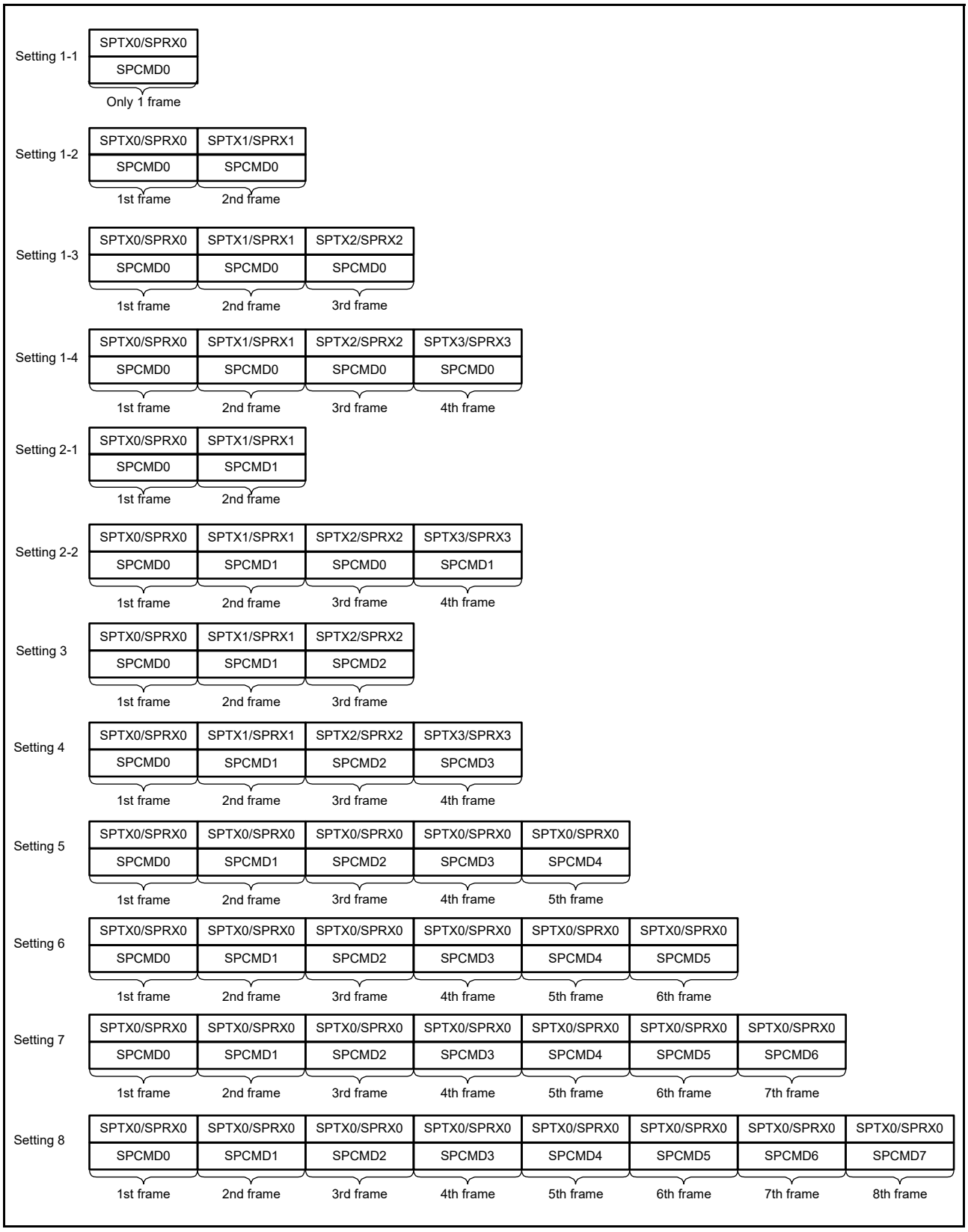


Figure 37.50 Correspondence between the RSPI Command Register and Transmit/Receive Buffers in Sequence Operations

(4) Initialization Flowchart

Figure 37.51 is a flowchart illustrating an example of initialization in clock synchronous operation when the RSPI is used in master mode. For a description of how to set up the interrupt controller, DMAC, and I/O ports, refer to the descriptions given in the individual blocks.

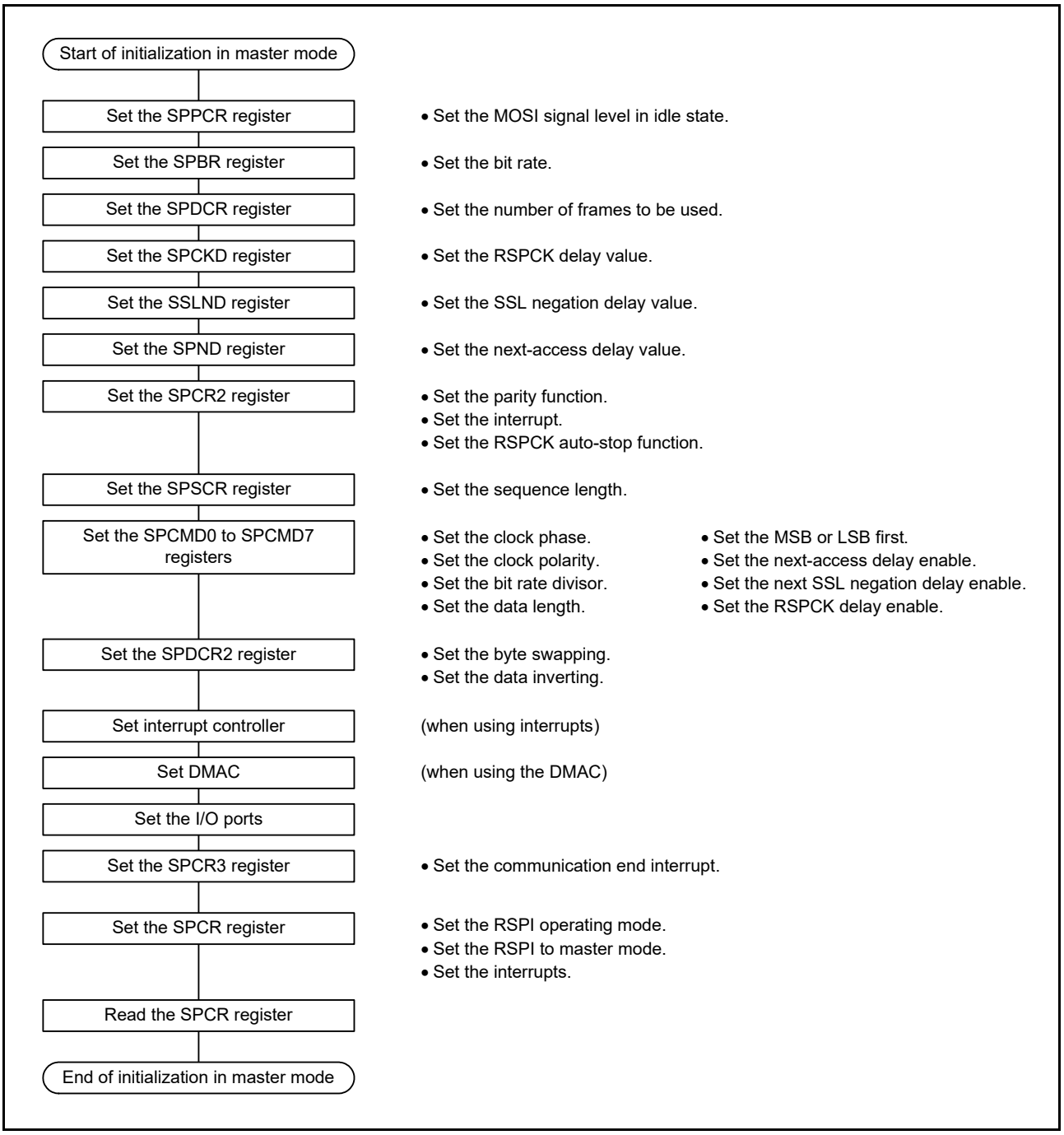


Figure 37.51 Example of Initialization Flowchart in Master Mode (Clock Synchronous Operation)

(5) Flow of Software Processing

Software processing during clock-synchronous operation when the RSPI is used in master mode is the same as that for SPI master mode operation. For details, refer to section 37.3.12.1, (9) Software Processing Flow. Note that mode fault errors will not occur.

37.3.13.2 Slave Mode Operation

(1) Starting a Serial Transfer

When the SPCR.SPMS bit is 1, the first RSPCKA edge triggers the start of a serial transfer in the RSPI.
When the SPMS bit is 1, the RSPI drives the MISOA output signal.
For details on the RSPI transfer format, refer to section 37.3.5, Transfer Format.
It should be noted that the SSLA0 input signal is not used in clock synchronous operation.

(2) Terminating a Serial Transfer

The RSPI terminates the serial transfer after detecting an RSPCKA edge corresponding to the final sampling timing. When free space is available in the receive buffer (the SPRF flag is 0), upon termination of serial transfer the RSPI copies received data from the shift register to the receive buffer of the SPDR register. Upon termination of a serial transfer the RSPI changes the status of the shift register to “empty” regardless of the receive buffer status. The final sampling timing changes depending on the bit length of transfer data. In slave mode, the RSPI data length depends on the SPCMD0.SPB[3:0] bit setting.
For details on the RSPI transfer format, refer to section 37.3.5, Transfer Format.

(3) Initialization Flowchart

Figure 37.52 is a flowchart illustrating an example of initialization in clock synchronous operation when the RSPI is used in slave mode. For a description of how to set up the interrupt controller, DMAC, and I/O ports, refer to the descriptions given in the individual blocks.

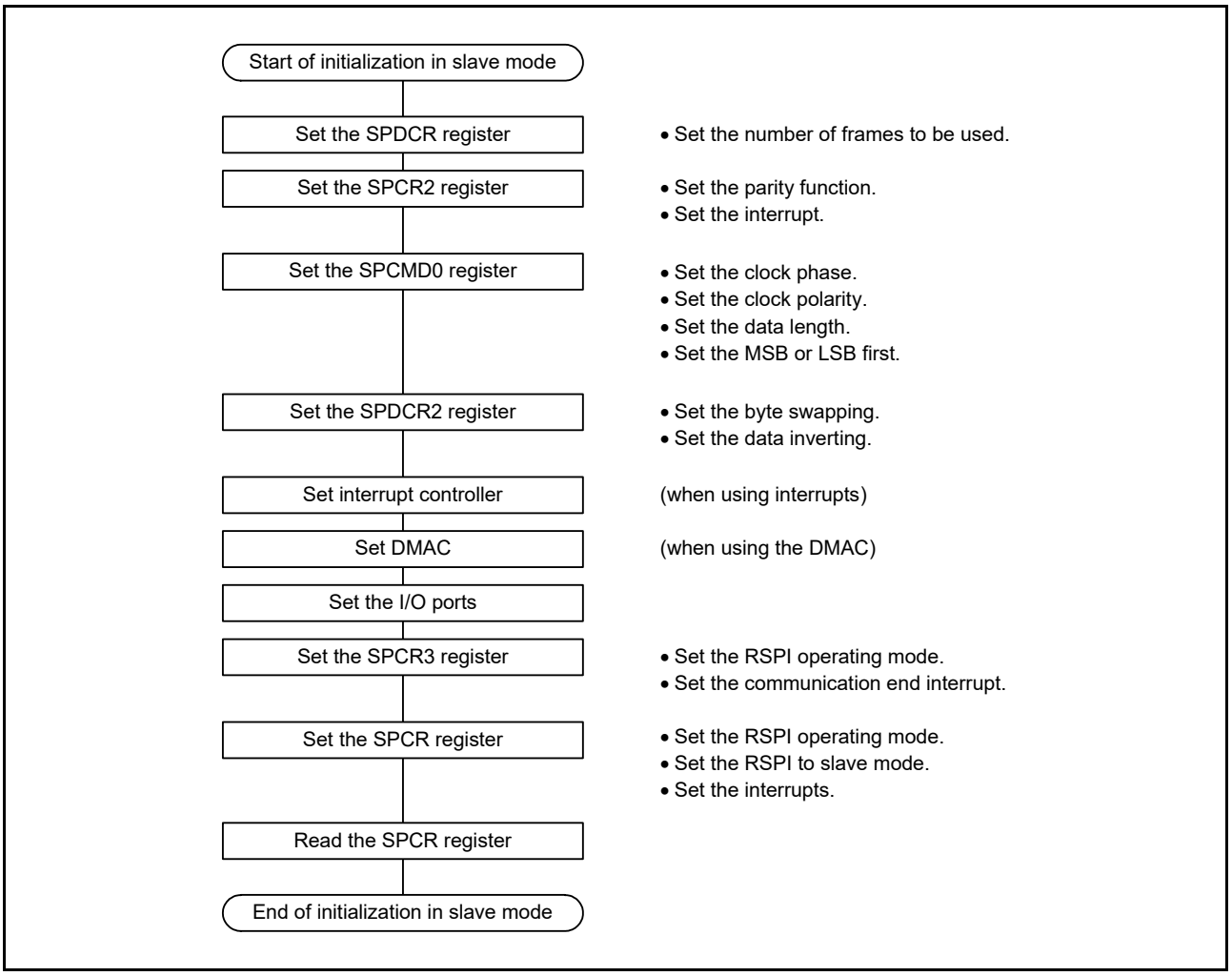


Figure 37.52 Example of Initialization Flowchart in Slave Mode (Clock Synchronous Operation)

(4) Flow of Software Processing

Software processing during clock-synchronous operation when the RSPI is used in slave mode is the same as that for SPI slave mode operation. For details, refer to section 37.3.12.2, (6) Software Processing Flow. Note that mode fault errors will not occur.

37.3.14 Loopback Mode

When 1 is written to the SPPCR.SPLP2 bit or SPPCR.SPLP bit, the RSPI shuts off the path between the MISOA pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSIA pin and the shift register if the SPCR.MSTR bit is 0, and connects the input path and output path of the shift register. The RSPI does not shut off the path between the MOSIA pin and the shift register if the SPCR.MSTR bit is 1, and between the MISOA pin and the shift register if the SPCR.MSTR bit is 0. This is called loopback mode. When a serial transfer is executed in loopback mode, the transmit data for the RSPI or the reversed transmit data becomes the received data for the RSPI.
Table 37.11 lists the relationship among the SPLP2 and SPLP bits and the received data. Figure 37.53 shows the configuration of the shift register I/O paths for the case where the RSPI in master mode is set in loopback mode (SPPCR.SPLP2 = 0, SPPCR.SPLP = 1).

Table 37.11 SPLP2 and SPLP Bit Settings and Received Data

SPPCR.SPLP2 Bit	SPPCR.SPLP Bit	Received Data
0	0	Input data from the MOSIA pin or MISOA pin
0	1	Inverted transmit data
1	0	Transmit data
1	1	Transmit data

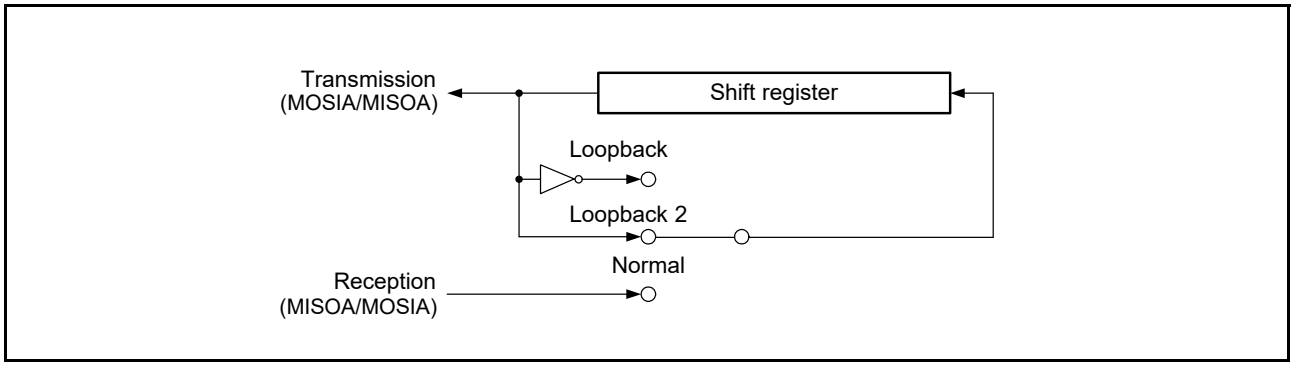


Figure 37.53 Configuration of Shift Register I/O Paths in Loopback Mode (Master Mode)

37.3.15 Self-Diagnosis of Parity Bit Function

The parity circuit consists of a parity bit adding unit used for transmit data and an error detecting unit used for received data. In order to detect defects in the parity bit adding unit and error detecting unit of the parity circuit, self-diagnosis is executed for the parity circuit following the flowchart shown in Figure 37.54.

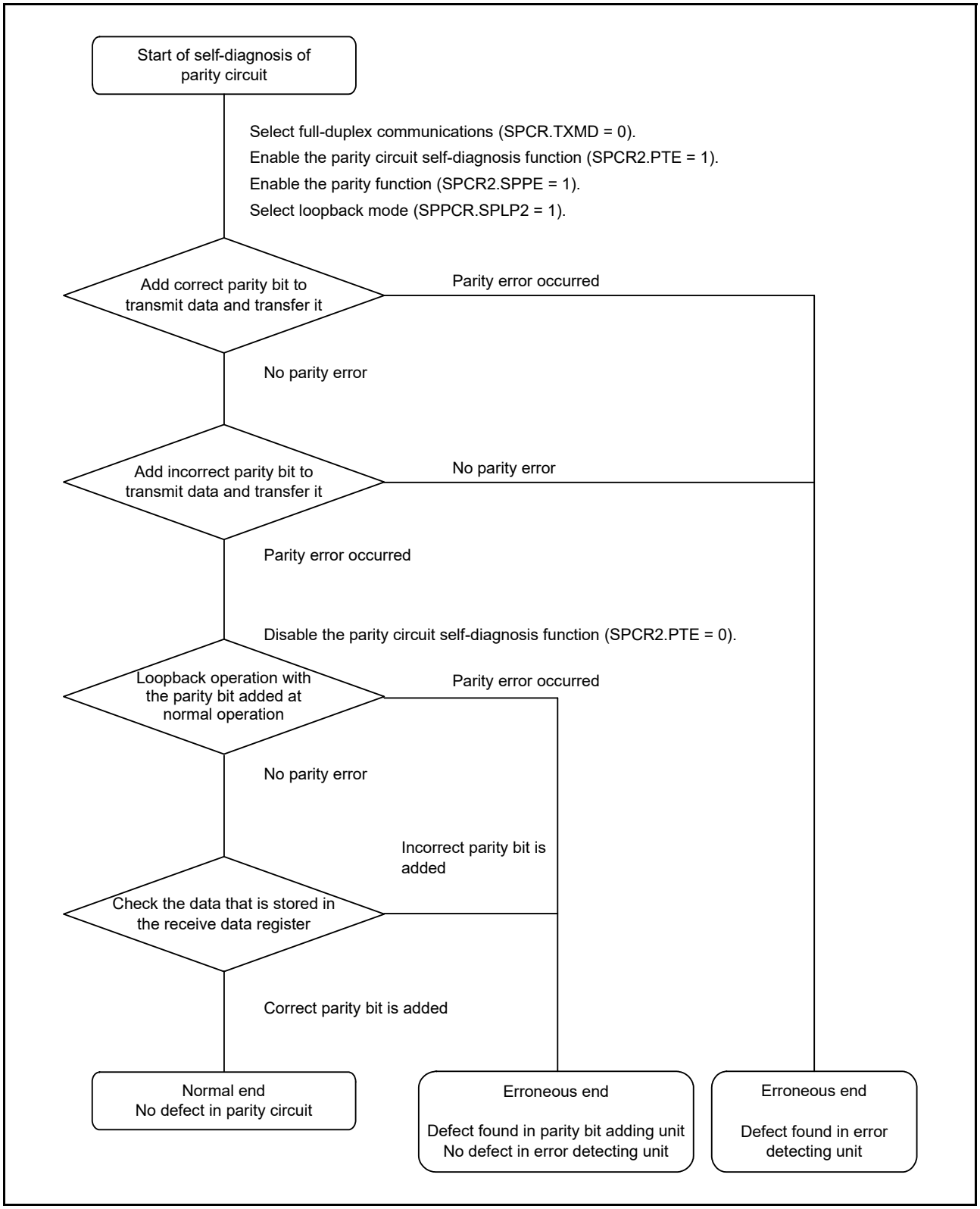


Figure 37.54 Flowchart for Self-Diagnosis of Parity Circuit

37.3.16 Interrupt Sources

The RSPI has interrupt sources of receive buffer full, transmit buffer empty, error (mode fault, underrun, overrun, and parity error), idle, and communication end. In addition, the DTC or DMAC can be activated by the receive buffer full or transmit buffer empty interrupt to perform data transfer.

Since the vector address for SPEI is allocated to interrupt requests due to mode fault, underrun, overrun, and parity errors, the actual interrupt source must be determined from the flags. Interrupt sources for the RSPI are listed in Table 37.12. An interrupt is generated on satisfaction of an interrupt condition in Table 37.12. Clear the receive buffer full and transmit buffer empty sources through data transfer.

When using the DTC or DMAC to perform data transmission/reception, the DTC or DMAC must be set up first to be in a status in which transfer is enabled before making the RSPI settings. For the method for setting the DTC or DMAC, refer to section 17, DMA Controller (DMACAa), or section 18, Data Transfer Controller (DTCb).

If the conditions for generating a transmit buffer empty or receive buffer full interrupt are generated while the ICU.IRn.IR flag is 1, the interrupt is not output as a request for ICU but is retained internally (the capacity for retention is one request per source). A retained interrupt request is output when the ICU.IRn.IR flag becomes 0. A retained interrupt request is automatically discarded once it is output as an actual interrupt request. The interrupt enable bit (the SPCR.SPTIE or SPCR.SPRIE bit) for an internally retained interrupt request can also be cleared to 0.

Table 37.12 Interrupt Sources of RSPI

Interrupt Source	Symbol	Interrupt Condition	DMAC/DTC Activation
Receive buffer full	SPRI	The receive buffer becomes full (the SPRF flag becomes 1) while the SPCR.SPRIE bit is 1.	Possible
Transmit buffer empty	SPTI	The transmit buffer becomes empty (the SPTEF flag becomes 1) while the SPCR.SPTIE bit is 1.	Possible
Errors (mode fault, underrun, overrun, and parity error)	SPEI	The SPSR.MODF, UDRF, OVRF, or PERF flag is set to 1 while the SPCR.SPEIE bit is 1.	Impossible
Idle	SPII	The SPSR.IDLNF flag is set to 0 while the SPCR2.SPIIE bit is 1.	Impossible
Communication end	SPCI	The SPSR.SPCF flag is set to 1 while the SPCR3.SPCIE bit is 1.	Impossible

37.4 Link Operation by Event Linking

The RSPI0 supports the following event output for the event link controller (ELC). The event link output signal is output regardless of the interrupt enable bit setting.

37.4.1 Receive Buffer Full Event Output

This event signal is output when received data have been transferred from the shift register to the SPDR register on completion of serial transfer.

37.4.2 Transmit Buffer Empty Event Output

This event signal is output when data for transmission have been transferred from the transmit buffer to the shift register and when the value of the SPE bit has changed from 0 to 1.

37.4.3 Mode Fault, Underrun, Overrun, or Parity Error Event Output

(1) Mode Fault

Table 37.13 lists the occurrence conditions of a mode fault event.

Table 37.13 Occurrence Conditions of Mode Fault Event

	SPCR.MODFEN Bit	SSLA0 Pin	Remarks
Master (SPCR.MSTR bit = 1)	1	Active	Under the condition (the SPCR.MSTR bit is 1 and the MODFEN bit is 1), if the SPCR.SPMS bit is 0, mode fault error, overrun error, and parity error event output cannot be used. Do not set the ELSRn register to 52h.
Slave (SPCR.MSTR bit = 0)	1	Not active	Event is output only when the pin is deactivated during transmission.

(2) Underrun

The condition for this event signal being output in response to an underrun is the start of serial transfer with the transmit buffer containing no transmit data while the value of the SPCR.MSTR bit is 0 and the value of the SPCR.SPE bit is 1, in which case the UDRF and MODF flags are set to 1.

(3) Overrun

The condition for this event signal being output in response to an overrun is completion of serial transfer while the receive buffer contains data that have not been read and the value of the SPCR.TXMD bit is 0, in which case the OVRF flag is set to 1.

(4) Parity Error

The condition for this event signal being output in response to a parity error is detection of a parity error on completion of serial transfer while the value of the TXMD bit in SPCR is 0 and the value of the SPPE bit in SPCR2 is 1.

37.4.4 Idle Event Output

(1) In Master Mode

In master mode, an event is output when the condition for setting the IDLNF flag (idle flag) to 0 is satisfied.

(2) In Slave Mode

In slave mode, an event is output when the SPCR.SPE bit is set to 0 (RSPI is initialized).

37.4.5 Communication End Event Output

During both SPI operation and clock synchronous operation in master mode, an event is output under the condition for setting the IDLNF flag (idle flag) from 1 to 0. In slave mode, an event is output under the conditions listed in Table 37.14 and Table 37.14.

Table 37.14 Generating Conditions of Communication End Event (Slave mode, full-duplex or transmit-only simplex communications)

RSPI Mode	Transmit Buffer State	Shift Register State	Others
SPI operation (SPMS = 0)	Empty	Empty	Negation of the SSLA0 input
Clock synchronous operation (SPMS = 1)	Empty	Empty	Even edge detection of the last RSPCKA for the last data

Table 37.15 Generating Conditions of Communication End Event (Slave mode, receive-only simplex communications)

RSPI Mode	Condition
SPI operation (SPMS = 0)	Negation of the SSLA0 input after the last data was received
Clock synchronous operation (SPMS = 1)	Even edge detection of the last RSPCKA for the last data

Whether the operation is in master mode or slave mode, an event is not output if 0 is written to the SPCR.SPE bit during communication or the SPCR.SPE bit is cleared by the mode fault error.

37.5 Usage Notes

37.5.1 Setting Module Stop Function

Module stop control register B (MSTPCRB) can be used to enable or disable the RSPI. Immediately after a reset, operation of the RSPI is disabled. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

37.5.2 Note on Low Power Consumption Functions

When using the module stop function and entering a low power consumption mode other than sleep mode, set the SPCR.SPE bit to 0 before completing communication.

37.5.3 Notes on Starting Transfer

If the ICU.IRn.IR flag is 1 at the time transfer is to be started, an interrupt request is internally retained after transfer starts, and this can lead to unanticipated behavior of the ICU.IRn.IR flag.
When the ICU.IRn.IR flag is 1 at the time transfer is to start, follow the procedure below to clear interrupt requests before enabling operations (by setting the SPCR.SPE bit to 1).

1. Confirm that transfer has stopped (i.e. that the SPCR.SPE bit is 0).
2. Set the relevant interrupt enable bit (the SPCR.SPTIE or SPCR.SPRIE bit) to 0.
3. Read the relevant interrupt enable bit (the SPCR.SPTIE or SPCR.SPRIE bit) and confirm that its value is 0.
4. Set the ICU.IRn.IR flag to 0.

37.5.4 Notes on the SPRF and SPTEF flags

When polling the SPSR.SPRF flag and/or SPSR.SPTEF flag, set the SPCR.SPRIE bit and/or SPCR.SPTIE bit to 0.

38. Serial Peripheral Interface (RSPIA)

38.1 Overview

The serial peripheral interface (RSPIA) supports high-speed, full-duplex or simplex synchronous serial communications with multiple processors and peripheral devices.

38.1.1 Features

The RSPIA has the following features.

Table 38.1 RSPIA Specifications (1/2)

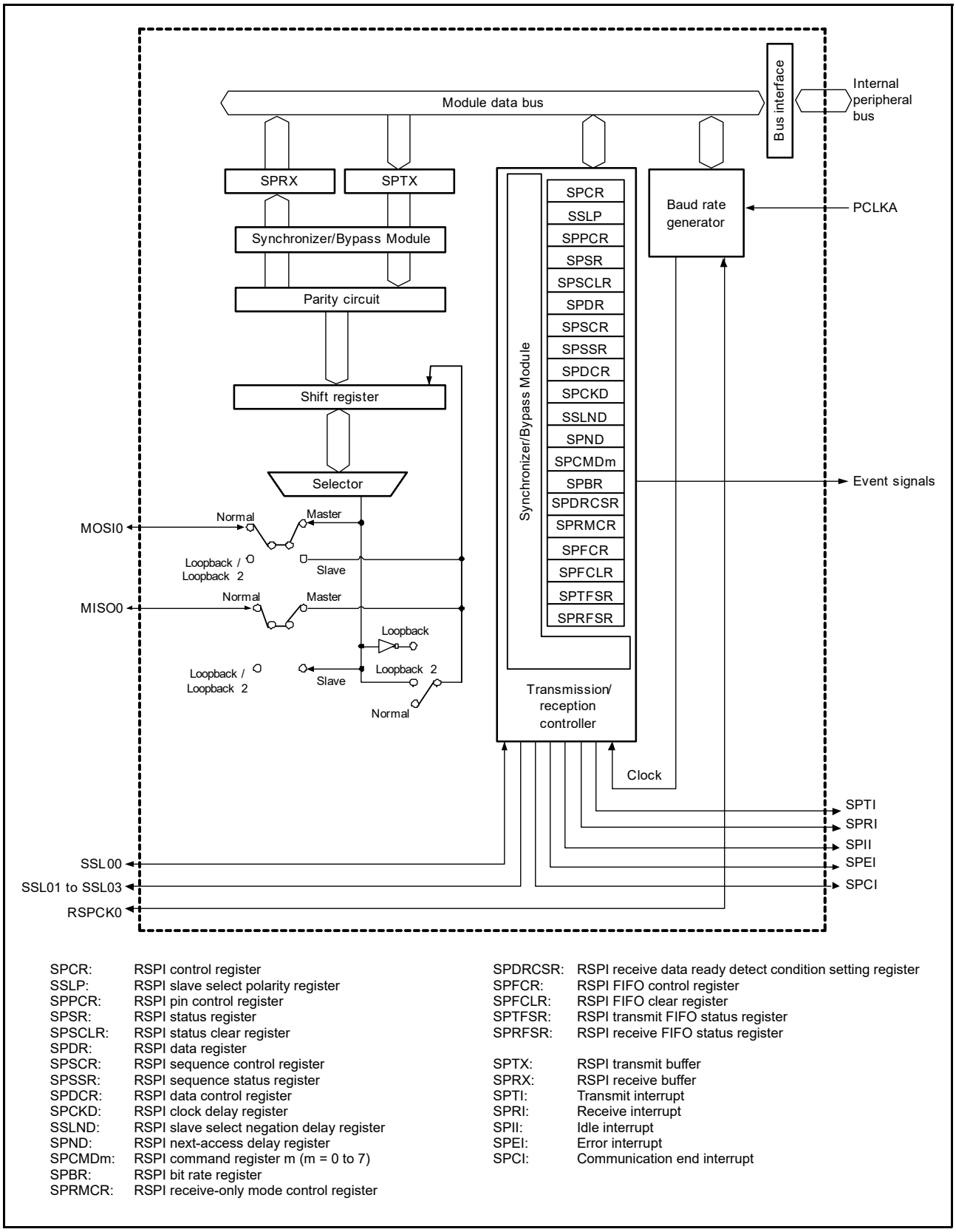
Item	Description
Transfer functions	• Use of MOSI (master out/slave in), MISO (master in/slave out), SSL (slave select), and RSPCK (RSPI clock) signals allows serial communications through SPI operation (4-wire method) or clock synchronous operation (3-wire method). • Full-duplex or simplex (transmit-only or receive-only) communications can be selected. • Serial communication is possible in master mode and slave mode. • Variable serial transfer clock polarity • Variable serial transfer clock phase
Data format	• MSB first/LSB first selectable • Transfer bit length can be changed to 4 through 32 bits. • Transmit buffer size/receive buffer size: 32 bits × 4 stages FIFO • Up to four frames can be transferred in one round of transmission/reception (each frame consisting of up to 32 bits). • Byte swapping of transmit and receive data is selectable. • Logic level of transmit and receive data can be inverted.
Bit rate	• In master mode, the on-chip baud rate generator generates RSPCK by frequency-dividing PCLKA. The division ratio ranges from divided by 2 to divided by 4096. • In slave mode, an external input clock is used as a serial clock. The maximum frequency of RSPCK is that of PCLKA divided by 2 (width at high level: 1 cycle of PCLKA; width at low level: 1 cycle of PCLKA).
Buffer configuration	• Transmit buffer and receive buffer are configured independently.
Error detection	• Mode fault error detection • Underrun error detection • Overrun error detection • Parity error detection • Receive data ready detection
SSL control function	[Motorola SPI mode] • Four SSL signals per RSPIA channel (SSL00 to SSL03) • In single-master mode, the SSL00 to SSL03 signals are output. • In multi-master mode: The SSL00 signal is input and the SSL01 to SSL03 signals are output or Hi-Z. • In slave mode: The SSL00 signal is input and the SSL01 to SSL03 signals are Hi-Z (not used). • Controllable delay from SSL output assertion to RSPCK operation (RSPCK delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Controllable delay from RSPCK stop to SSL output negation (SSL negation delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Controllable wait for next-access SSL output assertion (next-access delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Function for changing SSL polarity

Table 38.1 RSPIA Specifications (2/2)

Item	Description
SSL control function	[TI SSP mode] • Four SSL signals per RSPIA channel (SSL00 to SSL03) • In single-master mode, the SSL00 to SSL03 signals are output. • In multi-master mode: The SSL00 signal is input and the SSL01 to SSL03 signals are output or Hi-Z. • In slave mode: The SSL00 signal is input and the SSL01 to SSL03 signals are Hi-Z (not used). • Controllable delay from SSL output assertion to RSPCK operation (RSPCK delay) Range: 0 to 8 RSPCK cycles (set in RSPCK-cycle units) • Controllable delay from RSPCK stop to OE output negation (OE negation delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Controllable wait for next-access SSL output assertion (next-access delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Function for changing SSL polarity
Communication Protocol	• Motorola SPI • TI SSP (synchronous serial protocol)
Control in master transfer	[Motorola SPI mode] • A transfer of up to eight commands can be executed sequentially in looped execution. • For each command, the following can be set: SSL signal value, bit rate, RSPCK polarity/phase, transmit data length, MSB/LSB first, burst, RSPCK delay, SSL negation delay, and next-access delay • A transfer can be initiated by writing to the transmit buffer. • MOSI signal value specifiable in SSL negation • RSPCK auto-stop function (enabled or disabled by setting) • Delay between data bytes under burst transfer can be reduced. [TI SSP mode] • A transfer of up to eight commands can be executed sequentially in looped execution. • For each command, the following can be set: SSL signal value, bit rate, RSPCK polarity/phase, transmit data length, MSB/LSB first, burst, RSPCK delay, SSL negation delay (OE negation delay), and next-access delay • A transfer can be initiated by writing to the transmit buffer. • MOSI signal value specifiable in SSL negation • RSPCK auto-stop function (enabled or disabled by setting) • Delay between data bytes under burst transfer can be reduced.
Interrupt sources	Five interrupt sources • Receive buffer full/receive data ready interrupt • Transmit buffer empty interrupt • Communication end interrupt • Error interrupt (mode fault error, underrun error, overrun error, parity error, or receive data ready) • Idle interrupt
Event link function	Five event types can be output to the event link controller. • Receive buffer full/receive data ready event signal • Transmit buffer empty event signal • Mode fault error, underrun error, overrun error, parity error, or receive data ready event signal • Idle event signal • Communication end event signal
Others	• Function for disabling (initializing) the RSPI • Loopback mode • SPE bit status polling function
Low power consumption function	Module stop state can be set.

38.1.2 Block Diagram

Figure 38.1 shows a block diagram of the RSPIA.



SPCR: RSPI control register
SSLP: RSPI slave select polarity register
SPPCR: RSPI pin control register
SPSR: RSPI status register
SPSCLR: RSPI status clear register
SPDR: RSPI data register
SPSCR: RSPI sequence control register
SPSSR: RSPI sequence status register
SPDCR: RSPI data control register
SPCKD: RSPI clock delay register
SSLND: RSPI slave select negation delay register
SPND: RSPI next-access delay register
SPCMDm: RSPI command register m (m = 0 to 7)
SPBR: RSPI bit rate register
SPRMCR: RSPI receive-only mode control register
SPDRCSR: RSPI receive data ready detect condition setting register
SPFCR: RSPI FIFO control register
SPFCLR: RSPI FIFO clear register
SPTFSR: RSPI transmit FIFO status register
SPRFSR: RSPI receive FIFO status register

SPTX: RSPI transmit buffer
SPRX: RSPI receive buffer
SPTI: Transmit interrupt
SPRI: Receive interrupt
SPII: Idle interrupt
SPEI: Error interrupt
SPCI: Communication end interrupt

Figure 38.1 RSPIA Block Diagram

38.1.3 Pin Configuration

The RSPIA has serial pins listed in Table 38.2 for each channel. The RSPIA automatically switches the I/O direction of the SSL00 pin. SSL00 is set as an output when the RSPIA is a single master and as an input when the RSPIA is a multi-master or a slave. Pins RSPCK0, MOSI0, and MISO0 are automatically set as inputs or outputs according to the setting of master or slave, and the SPI operation (4-wire method) or clock synchronous operation (3-wire method) level input on the SSL00 pin. Refer to section 38.3.2, Controlling RSPI Pins for details.

Table 38.2 RSPIA Pin Configuration

Pin Name	Abbreviation	I/O	Function
RSPI clock	RSPCK0	I/O	Clock I/O
Master transmit data	MOSI0	I/O	Master transmit data I/O
Slave transmit data	MISO0	I/O	Slave transmit data I/O
Slave select 0	SSL00	I/O	Slave selection I/O
Slave select 1	SSL01	Output	Slave selection output
Slave select 2	SSL02	Output	Slave selection output
Slave select 3	SSL03	Output	Slave selection output

38.2 Register Descriptions

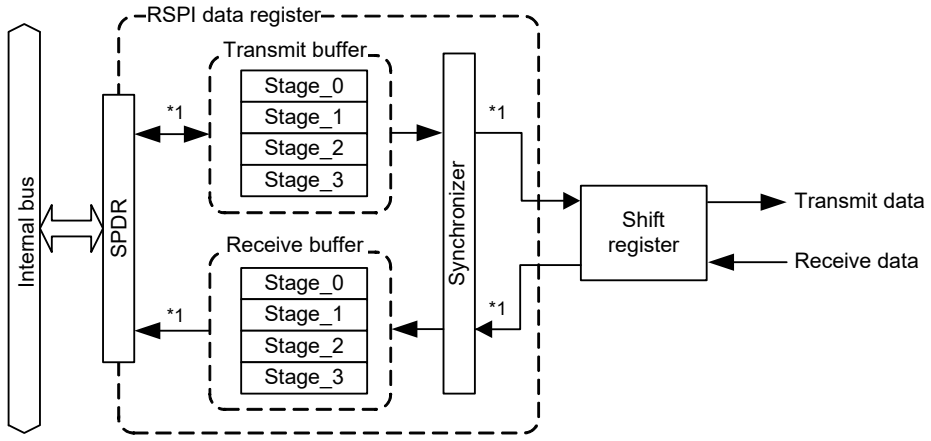
38.2.1 RSPI Data Register (SPDR)

Address(es): RSPIA0.SPDR 000E 2800h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The SPDR register is the interface with the buffers that hold data for transmission and reception by the RSPI. The transmit buffer and receive buffer are independent. Figure 38.2 shows the configuration of the SPDR register.



Note 1. The destination buffer and stage for access is automatically switched by hardware.

Figure 38.2 Configuration of SPDR Register

The transmit and receive buffers each have 32 bits × 4 stages of FIFO. The eight stages of FIFO are all mapped to the single address of the SPDR register. Data written to the SPDR register are written to a transmit-buffer stage (SPTXn, n = 0 to 3) and then transmitted from the buffer. The receive buffer (SPRXn, n = 0 to 3) holds received data on completion of reception. The receive buffer is not updated if an overrun is generated.

(1) Bus Interface

The SPDR register is the interface with 32-bit wide transmit buffer (SPTXn, n = 0 to 3) and receive buffer (SPRXn, n = 0 to 3), each of which has four stages, for a total of 32 bytes. In other words, the 32 bytes are mapped to the 4-byte address space for the SPDR register.
Data for transmission should be flush with the LSB end of the register. Received data are stored flush with the LSB end.
Operations involved in writing to and reading from the SPDR register are described below.

(a) Writing

The transmit buffer includes a transmit buffer write pointer which is automatically updated to indicate the next stage each time data are written to the SPDR register. Figure 38.3 shows the configuration of the bus interface with the transmit buffer in the case of writing to the SPDR register.

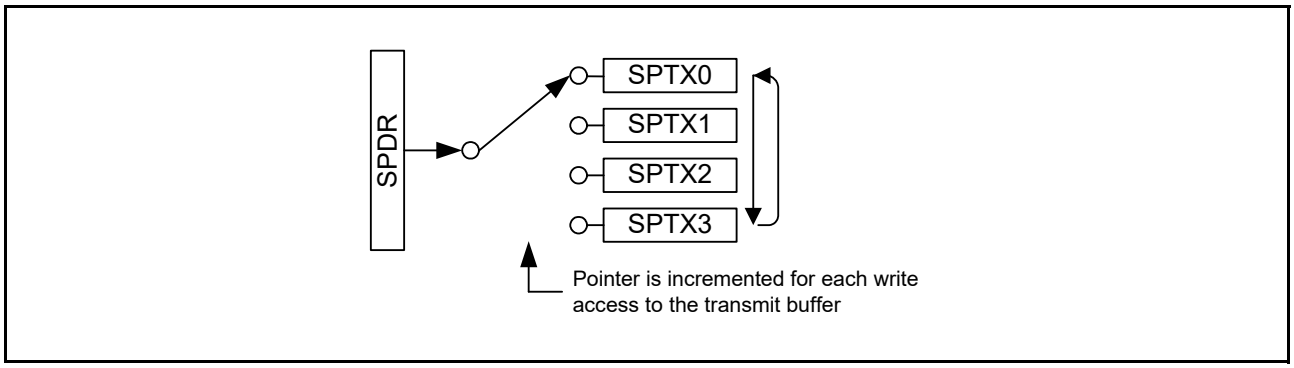


Figure 38.3 Configuration of SPDR Register (Writing)

Sequence of switching the pointer among SPTX0 to SPTX3:
SPTX0 → SPTX1 → SPTX2 → SPTX3 → SPTX0 → SPTX1 → …

When writing to the transmit buffer (SPTXn) after generation of the transmit buffer empty interrupt (after the SPSR.SPTEF flag becomes 1), write the number of frames + 1 set by the transmission FIFO threshold setting bits of RSPI FIFO control register (SPFCR.TTRG[1:0]). Writing to the transmit buffer (SPTXn, n = 0 to 3) in the state where there is no empty stages in the transmit FIFO does not update the buffer value.

(b) Reading

The SPDR register can be read to read the value of a receive buffer (SPRXn, n = 0 to 3) or a transmit buffer (SPTXn, n = 0 to 3). The setting of the RSPI receive/transmit data select bit in the RSPI data control register (SPDCR.SPRDTD) selects whether reading is of the receive or transmit buffer.
The sequence of reading the SPDR register is controlled by independent pointers, receive buffer read pointer and transmit buffer read pointer.
Figure 38.4 shows the configuration of the bus interface with the receive and transmit buffers in the case of reading from the SPDR register.

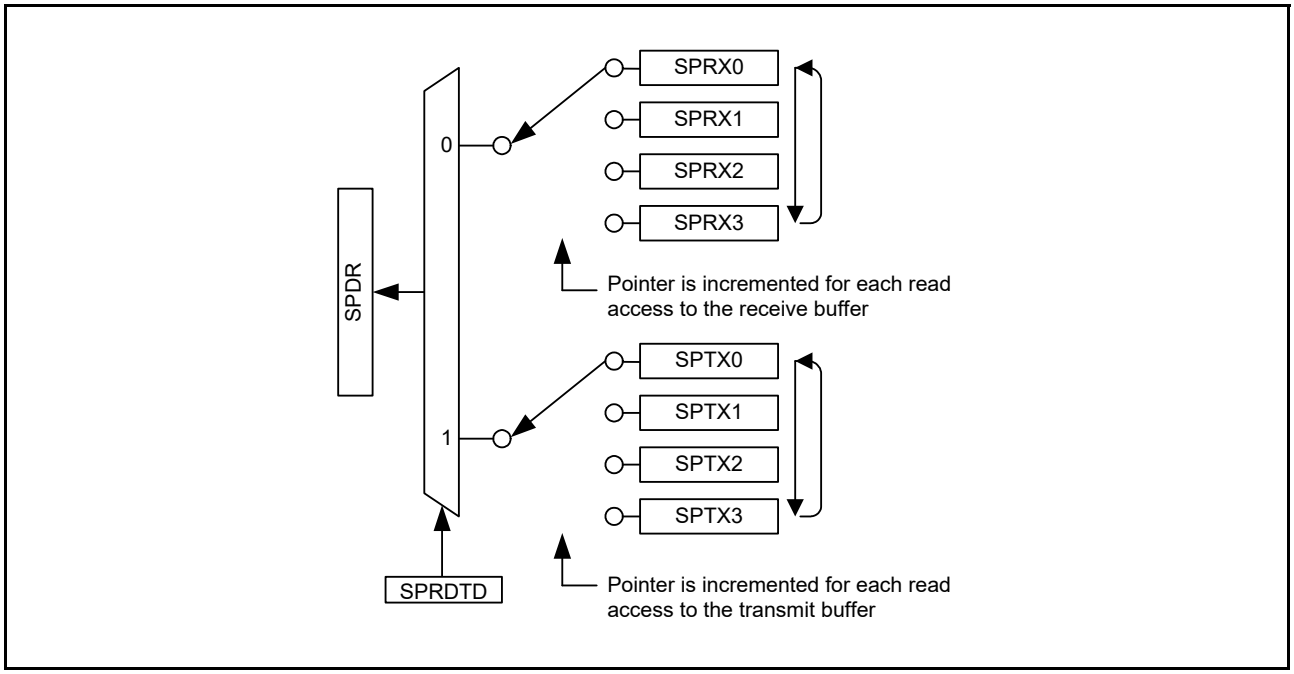


Figure 38.4 Configuration of SPDR (Reading)

Reading the receive buffer switches the receive buffer read pointer to the next buffer automatically.
The sequence of switching the receive buffer read pointer is the same as that for the transmit buffer write pointer.
The transmit buffer read pointer is updated when writing to the SPDR register, and not updated when reading from the transmit buffer. When reading from the transmit buffer, the value most recently written to the SPDR register is read.

38.2.2 RSPI Clock Delay Register (SPCKD)

Address(es): RSPIA0.SPCKD 000E 2804h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SCKDL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SCKDL[2:0]	RSPCK Delay Setting	b2 b0 0 0 0: 1 RSPCK 0 0 1: 2 RSPCK 0 1 0: 3 RSPCK 0 1 1: 4 RSPCK 1 0 0: 5 RSPCK 1 0 1: 6 RSPCK 1 1 0: 7 RSPCK 1 1 1: 8 RSPCK	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R

[Motorola SPI]
The SPCKD register sets a period from the beginning of SSL0n signal assertion to RSPCK oscillation (RSPCK delay) when the SPCMDm.SCKDEN bit is 1. If the value of the SPCKD register is changed while both the SPCR.MSTR and SPCR.SPE bits are 1, the operation after the change is not guaranteed.

[TI SSP]
The SPCKD register sets a period from the beginning of SSL0n signal assertion to RSPCK oscillation (RSPCK delay) when the SPCMDm.SCKDEN bit is 1, and also sets a period until the SSL0n signal is negated. If the value of the SPCKD register is changed while both the SPCR.MSTR and SPCR.SPE bits are 1, the operation after the change is not guaranteed.

SCKDL[2:0] Bits (RSPCK Delay Setting)
The SCKDL[2:0] bits set an RSPCK delay value when the SPCMDm.SCKDEN bit is 1.
When using the RSPIA in slave mode, set the SCKDL[2:0] bits to 000b.

38.2.3 RSPI Slave Select Negation Delay Register (SSLND)

Address(es): RSPIA0.SSLND 000E 2805h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SLNDL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SLNDL[2:0]	SSL Negation Delay Setting	[In master mode] b2 b0 0 0 0: 1 RSPCK 0 0 1: 2 RSPCK 0 1 0: 3 RSPCK 0 1 1: 4 RSPCK 1 0 0: 5 RSPCK 1 0 1: 6 RSPCK 1 1 0: 7 RSPCK 1 1 1: 8 RSPCK [In slave mode, TI SSP] b2 b0 0 0 0: 1 PCLKA 0 0 1: 2 PCLKA 0 1 0: 3 PCLKA 0 1 1: 4 PCLKA 1 0 0: 5 PCLKA 1 0 1: 6 PCLKA 1 1 0: 7 PCLKA 1 1 1: 8 PCLKA	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R

[Motorola SPI]
The SSLND register sets a period (SSL negation delay) from the transmission of a final RSPCK edge to the negation of the SSL0n signal during a serial transfer by the RSPIA in master mode when the SPCMDm.SLNDEN bit is 1. If the value of the SSLND register is changed while both the SPCR.MSTR and SPCR.SPE bits are 1, the operation after the change is not guaranteed.

[TI SSP]
The SSLND register sets a period (SSL negation delay) from the transmission of a final RSPCK edge to the negation of the OE signal during a serial transfer by the RSPIA in master mode, or from the detection of a final RSPCK edge to the negation of the OE signal during a serial transfer by the RSPIA in slave mode. If the value of the SSLND register is changed while the SPCR.SPE bit is 1, the operation after the change is not guaranteed.

SLNDL[2:0] Bits (SSL Negation Delay Setting)

[Motorola SPI]
The SLNDL[2:0] bits set an SSL negation delay value when the SPCMDm.SLNDEN bit is 1.
When using the RSPIA in slave mode, set the SLNDL[2:0] bits to 000b.

[TI SSP]
The SLNDL[2:0] bits set an OE negation delay value when the SPCMDm.SLNDEN bit is 1.

[Clock synchronous]
When using the RSPIA in slave mode, set the SLNDL[2:0] bits to 000b.

38.2.4 RSPI Next-Access Delay Register (SPND)

Address(es): RSPIA0.SPND 000E 2806h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SPNDL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SPNDL[2:0]	RSPI Next-Access Delay Setting	b2 b0 0 0 0: 1 RSPCK + 5 PCLKA 0 0 1: 2 RSPCK + 5 PCLKA 0 1 0: 3 RSPCK + 5 PCLKA 0 1 1: 4 RSPCK + 5 PCLKA 1 0 0: 5 RSPCK + 5 PCLKA 1 0 1: 6 RSPCK + 5 PCLKA 1 1 0: 7 RSPCK + 5 PCLKA 1 1 1: 8 RSPCK + 5 PCLKA	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R

The SPND register sets a non-active period (next-access delay) of the SSL0n signal after termination of a serial transfer when the SPCMDm.SPNDEN bit is 1. If the value of the SPND register is changed while both the SPCR.MSTR and SPCR.SPE bits are 1, the operation after the change is not guaranteed.

SPNDL[2:0] Bits (RSPI Next-Access Delay Setting)

The SPNDL[2:0] bits set a next-access delay when the SPCMDm.SPNDEN bit is 1.
When using the RSPIA in slave mode, set the SPNDL[2:0] bits to 000b.

38.2.5 RSPI Control Register (SPCR)

Address(es): RSPIA0.SPCR 000E 2808h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SYNDIS	MSTR	CMMD[1:0]		—	—	FRFS	SPMS	—	—	SPCIE	SPTIE	RDRIS	SPIIE	SPRIE	SPEIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	MODFEN	SCKDDIS	SCKASE	PTE	—	SPOE	SPPE	MRCKS	—	—	—	—	—	—	SPE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SPE	RSPI Function Enable	0: Disables the RSPI function 1: Enables the RSPI function	R/W
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R
b7	MRCKS	Master Receive Clock Select	Set this bit to 1.	R/W
b8	SPPE	Parity Enable	0: A parity bit is not added to transmit data. Received-data parity check is not performed. 1: A parity bit is added to transmit data. Received-data parity check is performed.	R/W
b9	SPOE	Parity Mode	0: Even parity is used for transmission and reception. 1: Odd parity is used for transmission and reception.	R/W
b10	—	Reserved	This bit is read as 0. The write value should be 0.	R
b11	PTE	Parity Self-Diagnosis	0: Disables the self-diagnosis function of the parity circuit 1: Enables the self-diagnosis function of the parity circuit	R/W
b12	SCKASE	RSPCK Auto-Stop Function Enable	0: Disables the RSPCK auto-stop function 1: Enables the RSPCK auto-stop function	R/W
b13	SCKDDIS	RSPCK Delay Between Data Bytes Disable	0: Delay between frames is inserted in burst transfer 1: Delay between frames is not inserted in burst transfer.	R/W
b14	MODFEN	Mode Fault Error Detection Enable	0: Disables the detection of mode fault error 1: Enables the detection of mode fault error	R/W
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R
b16	SPEIE	Error Interrupt Enable	0: Disables the error interrupt 1: Enables the error interrupt	R/W
b17	SPRIE	Receive Buffer Full Interrupt Enable	0: Disables the receive buffer full interrupt 1: Enables the receive buffer full interrupt	R/W
b18	SPIIE	Idle Interrupt Enable	0: Disables the idle interrupt 1: Enables the idle interrupt	R/W
b19	RDRIS	Receive Data Ready Interrupt Select	0: Receive data full interrupt 1: Error interrupt	R/W
b20	SPTIE	Transmit Buffer Empty Interrupt Enable	0: Disables the transmit buffer empty interrupt 1: Enables the transmit buffer empty interrupt	R/W
b21	SPCIE	Communication End Interrupt Enable	0: Disables the communication end interrupt 1: Enables the communication end interrupt	R/W
b23, b22	—	Reserved	These bits are read as 0. The write value should be 0.	R
b24	SPMS	RSPI Mode Select	0: SPI operation (4-wire method) 1: Clock synchronous operation (3-wire method)	R/W
b25	FRFS	Frame Format Select	0: Motorola SPI 1: TI SSP When the SPMS bit is 1 (clock synchronous operation (3-wire method)), the setting of this bit is disabled.	R/W
b27, b26	—	Reserved	These bits are read as 0. The write value should be 0.	R

Bit	Symbol	Bit Name	Description	R/W
b29, b28	CMMD[1:0]	Communication Mode Select	b29 b28 0 0: Transmit-receive mode (full-duplex) 0 1: Transmit-only mode (simplex, receiver disabled) 1 0: Receive-only mode (simplex, transmitter disabled) 1 1: Setting prohibited	R/W
b30	MSTR	RSPI Master/Slave Mode Select	0: Slave mode 1: Master mode	R/W
b31	SYNDIS	Synchronizer Disable	Set this bit to 1.	R/W

The SPCR register is used to specify the operating modes. If the value of the MRCKS, SPPE, SPOE, PTE, SCKASE, SCKDDIS, MODFEN, SPMS, FRFS, CMMD[1:0], MSTR, or SYNDIS bit is changed while the SPCR.SPE bit is 1, the operation after the change is not guaranteed.

SPE Bit (RSPI Function Enable)

This bit enables or disables the RSPI function. Setting this bit to 1 enables the RSPI function. When the SPSR.MODF flag is 1, this bit is set to 0. This bit cannot be set to 1 until the MODF flag is cleared. For details, refer to section 38.3.10, Error Detection.

Setting this bit to 0 disables the RSPI function, and initializes a part of the module function. For details, refer to section 38.3.12, Initializing RSPIA.

MRCKS Bit (Master Receive Clock Select)

This bit selects the clock used to sample the received data in master mode. Set this bit to 1 for this MCU. This bit setting is effective only when the loopback function is not used (when SPPCR.SPLP = 0 and SPPCR.SPLP2 = 0) and in master mode.

SPPE Bit (Parity Enable)

This bit is used to enable or disable the parity function.

SPOE Bit (Parity Mode)

This bit is used to specify even parity or odd parity.

In even parity mode, the parity bit is determined so that the sum of 1 (parity bit + transmit characters or receive characters) becomes an even number. In the same way, in odd parity mode, a parity bit is determined so that the sum of 1 (parity bit + transmit characters or receive characters) becomes an odd number. The SPOE bit is effective only when the SPCR.SPPE bit is set to 1.

PTE Bit (Parity Self-Diagnosis)

This bit is used to enable or disable self-diagnosis of the parity circuit to confirm that the parity function is normal.

SCKASE Bit (RSPCK Auto-Stop Function Enable)

This bit is used to enable or disable the RSPCK auto-stop function. When this function is enabled, the RSPCK stops immediately before an overrun error occurs during data reception in master mode. For details, refer to section 38.3.10.1, Overrun Error.

SCKDDIS Bit (RSPCK Delay Between Data Bytes Disable)

This bit controls whether insert the delay time between the burst transfer frames.

Valid in the master mode (SPCR.MSTR = 1) for frames with the SPCMDm.SSLKP bit set to 1.

This bit should be set to 0 in slave mode (SPCR.MSTR = 0). Table 38.3 shows the usage of SSL delay control between transfer frames. For details, refer to (4) Burst Transfer in section 38.3.13.1, Master Mode Operation.

Table 38.3 Usage of SSL Delay Control between Transfer Frames (Master mode)

Conditions		SPCMDm.SSLKP bit	SPCR.SCKDDIS bit	SSL delay control register *1 (RSPCK delay, SSL negation delay, next access delay)
Non-burst transfers		0	0	It is possible to control each delay of RSPCK delay, SSL negation delay and next access delay
Burst transfer with delay between frames	From the 1st frame to the last previous frame	1	0	
	The last frame	0	0	
Burst transfer with no delay between frames	From the 1st frame to the last previous frame	1	1	• RSPCK delay of the 1st frame • SSL negation delay and next access delay of the last frame
	The last frame	0	1	

Note 1. RSPCK delay is controlled by the SPCKD.SCKDL[2:0] bits and SPCMDm.SCKDEN bit.
SSL negation delay is controlled by the SSLND.SLNDL[2:0] bits and SPCMDm.SLNDEN bit.
Next access delay is controlled by the SPND.SPNDL[2:0] bits and SPCMDm.SPNDEN bit.

Setting	Operation example (Motorola SPI, SCKDDIS = 1)
SPCMD0.SSLKP = 1	Burst transfer/no interframe delay between 0 and 1 (SSL0n keep active)
SPCMD1.SSLKP = 1	Burst transfer/no interframe delay between 1 and 2 (SSL0n keep active)
SPCMD2.SSLKP = 1	Burst transfer/no interframe delay between 2 and 3 (SSL0n keep active)
SPCMD3.SSLKP = 1	Burst transfer/no interframe delay between 3 and 4 (SSL0n keep active)
SPCMD4.SSLKP = 0	Does not perform burst transfer, and once in-activate SSL0n.*1
SPCMD5.SSLKP = 1	Burst transfer/no interframe delay between 5 and 6 (SSL0n keep active)
SPCMD6.SSLKP = 1	Burst transfer/no interframe delay between 6 and 7 (SSL0n keep active)
SPCMD7.SSLKP = 0	Does not perform burst transfer, and once in-activate SSL0n.*1

Note 1. SCKDDIS setting is invalid because it does not perform burst transfer.

MODFEN Bit (Mode Fault Error Detection Enable)

This bit is used to enable or disable detection of a mode fault error. (Refer to section 38.3.10, Error Detection.) The RSPIA determines SSL00 pin input or output direction according to the combination of the MODFEN and MSTR bits (refer to section 38.3.2, Controlling RSPI Pins).

SPEIE Bit (Error Interrupt Enable)

This bit is used to enable or disable an error interrupt request.
When the SPSR.MODF, OVRF, or PERF flag is set to 1 while this bit is 1, an error interrupt request is generated (refer to section 38.3.10, Error Detection).

SPRIE Bit (Receive Buffer Full Interrupt Enable)

This bit is used to enable or disable a receive buffer full interrupt request.

SPIIE Bit (Idle Interrupt Enable)

This bit is used to enable or disable an idle interrupt requests.
When the SPSR.IDLNF flag changes from 1 to 0 while this bit is 1, an idle interrupt request is generated.

RDRIS Bit (Receive Data Ready Interrupt Select)

This bit is used to select the SPRI interrupt request or SPEI interrupt request to be generated when the receive data ready is detected (SPSR.RRDYF = 1).

SPTIE Bit (Transmit Buffer Empty Interrupt Enable)

This bit is used to enable or disable a transmit buffer empty interrupt request.

SPCIE Bit (Communication End Interrupt Enable)

This bit controls generation of a communication end interrupt request.

SPMS Bit (RSPI Mode Select)

This bit is used to select SPI operation (4-wire method) or clock synchronous operation (3-wire method).
For clock synchronous operation, the SSL0n pin is not used but three pins RSPCK0, MOSI0, and MISO0 are used for communication. When SPMS = 1 (clock synchronous operation (3-wire)), the setting of the FRFS bit is invalid.
To perform clock synchronous operation in master mode (SPCR.MSTR = 1), the SPCMDm.CPHA bit can be set to 0 or 1. To perform clock synchronous operation in slave mode (SPCR.MSTR = 0), set the CPHA bit to 1. If this bit is set to 0 for clock synchronous operation in slave mode (SPCR.MSTR = 0), subsequent operation is not guaranteed.
Table 38.4 lists the communication status according to the settings of the MSTR bit, CMMD[1:0] bits, FRFS bit, and SPMS bit of the RSPI control register (SPCR) as follows.

Table 38.4 RSPI Communication Status

SPCR. MSTR	SPCR. CMMD[1:0]	SPCR. FRFS	SPCR. SPMS	Communication Status	Communi-cation Status No.
1	00b	0	0	Transmit-Receive Master mode/Motorola SPI/SPI operation (4-wire)	1-1
1	00b	1	0	Transmit-Receive Master mode/TI SSP/SPI operation (4-wire)	1-2
1	00b	—	1	Transmit-Receive Master mode/Clock synchronous operation (3-wire)	1-3
1	01b	0	0	Transmit-only Master mode/Motorola SPI/SPI operation (4-wire)	1-4
1	01b	1	0	Transmit-only Master mode/TI SSP/SPI operation (4-wire)	1-5
1	01b	—	1	Transmit-only Master mode/Clock synchronous operation (3-wire)	1-6
1	10b	0	0	Receive-only Master mode/Motorola SPI/SPI operation (4-wire)	1-7
1	10b	1	0	Receive-only Master mode/TI SSP/SPI operation (4-wire)	1-8
1	10b	—	1	Receive-only Master mode/Clock synchronous operation (3-wire)	1-9
0	00b	0	0	Transmit-Receive Slave mode/Motorola SPI/SPI operation (4-wire) (default)	0-1
0	00b	1	0	Transmit-Receive Slave mode/TI SSP/SPI operation (4-wire)	0-2
0	00b	—	1	Transmit-Receive Slave mode/Clock synchronous operation (3-wire)	0-3
0	01b	0	0	Transmit-only Slave mode/Motorola SPI/SPI operation (4-wire)	0-4
0	01b	1	0	Transmit-only Slave mode/TI SSP/SPI operation (4-wire)	0-5
0	01b	—	1	Transmit-only Slave mode/Clock synchronous operation (3-wire)	0-6
0	10b	0	0	Receive-only Slave mode/Motorola SPI/SPI operation (4-wire)	0-7
0	10b	1	0	Receive-only Slave mode/TI SSP/SPI operation (4-wire)	0-8
0	10b	—	1	Receive-only Slave mode/Clock synchronous operation (3-wire)	0-9

—: don't care

FRFS Bit (Frame Format Select)

This bit selects the communication protocol. The format of the RSPI terminal (RSPCK0, SSL0n (n = 0 to 3) can be set according to the set communication protocol.
When the SPMS bit is 1 (clock synchronous operation (3-wire method)), this bit is invalid because SSL0n pin is not used.

CMMD[1:0] Bits (Communication Mode Select)

This bit is used to select the transmit-receive, transmit-only, and receive-only serial communication.
When CMMD[1:0] is set to 01b for communication, transmit-only is performed without reception.
When CMMD[1:0] is set to 10b for communication, receive-only is performed without transmission.

When CMMD[1:0] is set to 01b for communication, a receive buffer full interrupt request cannot be used.
When CMMD[1:0] is set to 10b for communication, a transmit buffer empty interrupt request cannot be used.
Refer to section 38.3.6, Communication Mode.

MSTR Bit (RSPI Master/Slave Mode Select)

This bit is used to select master mode or slave mode of the RSPIA. The RSPIA determines input/output directions of pins RSPCK0, MOSI0, MISO0, and SSL01 to SSL03 according to the MSTR bit setting.

SYNDIS Bit (Synchronizer Disable)

Set this bit to 1 for this MCU.

38.2.6 RSPI Receive-Only Mode Control Register (SPRMCR)

Address(es): RSPIA0.SPRMCR 000E 280Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	START	TERM	—	RFC[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	RFC[4:0]	Receive Frame Count Select	The number of received frames can be adjusted in receive-only master mode. b4 b0 0 0 0 0 0: This function is not used. 0 0 0 0 1: Automatically stop communication after processing 1 received frame : 1 1 1 1 1: Automatically stop communication after processing 31 received frames Settings are not effective except in receive-only master mode.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R
b6	TERM	Reception Terminate	1: Receive End (Writable only in receive-only master mode) The read value is always 0.	W
b7	START	Reception Start	1: Receive Start (Writable only in receive-only master mode) The read value is always 0.	W

The SPRMCR register is used to control the start and completion of communication for receive-only master mode. If the RFC[4:0] bits are changed while the SPCR.SPE bit is 1, the operation after the change is not guaranteed.

RFC[4:0] Bits (Receive Frame Count Select)

The number of received frames can be adjusted when operating in receive-only master mode. Valid only when the master mode (SPCR.MSTR = 1) and the communication mode select bits (SPCR.CMMD[1:0]) are 10b. After reception is started by the START bit, the communication is automatically stopped after processing frames according to the value set in these bits.

TERM Bit (Reception Terminate)

This bit is used to end reception in receive-only master mode. Valid only when the master mode (SPCR.MSTR = 1) and the communication mode select bits (SPCR.CMMD[1:0]) are 10b.

START Bit (Reception Start)

This bit is used to start reception in receive-only master mode. Valid only when the master mode (SPCR.MSTR = 1) and the communication mode select bits (SPCR.CMMD[1:0]) are 10b.
Writing 1 to this bit during reception is not accepted. Write again after reception is completed.

38.2.7 RSPI Receive Data Ready Detect Condition Setting Register (SPDRCSR)

Address(es): RSPIA0.SPDRCSR 000E 280Dh

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	—	0h: Disables receive data ready detection function 1h: Performs receive data ready detection after 1 PCLKA : FFh: Performs receive data ready detection after 255 PCLKA	R/W

The SPDRCSR register is used to set the RSPI receive data ready detection function. If the setting value is changed while the SPCR.SPE bit is 1, the operation after the change is not guaranteed.
The receive data ready detection function can be disabled or, if used, the period until detection can be set from 1 to 255 PCLKA. The value set in the SPDRCSR register is used to 1 set the RRDYF flag. For details, see the description of the RRDYF flag in section 38.2.16, RSPI Status Register (SPSR).

38.2.8 RSPI Pin Control Register (SPPCR)

Address(es): RSPIA0.SPPCR 000E 280Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	MOIFE	MOIFV	—	—	SPLP2	SPLP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SPLP	RSPI Loopback	0: Normal mode 1: Loopback mode (data is inverted for transmission)	R/W
b1	SPLP2	RSPI Loopback 2	0: Normal mode 1: Loopback mode (data is not inverted for transmission)	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b4	MOIFV	MOSI Idle Fixed Value	0: The fixed value during MOSI idling corresponds to low 1: The fixed value during MOSI idling corresponds to high	R/W
b5	MOIFE	MOSI Idle Value Fixing Enable	0: MOSI output value equals final data from previous transfer 1: MOSI output value equals the value set in the MOIFV bit	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R

The SPPCR register is used to set pin mode of the RSPIA. If this register is changed while the SPCR.SPE bit is 1, the operation after the change is not guaranteed.

SPLP Bit (RSPI Loopback)

When the SPLP bit is set to 1, the RSPIA shuts off the path between the MISO0 pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSI0 pin and the shift register if the SPCR.MSTR bit is 0, and connects (inverts) the input path and output path for the shift register (loopback mode).

SPLP2 Bit (RSPI Loopback 2)

When the SPLP2 bit is set to 1, the RSPIA shuts off the path between the MISO0 pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSI0 pin and the shift register if the SPCR.MSTR bit is 0, and connects the input path and output path for the shift register (loopback mode).
If this bit is set to 1 together with the SPLP bit, setting this bit takes precedence.

MOIFV Bit (MOSI Idle Fixed Value)

If the MOIFE bit is 1 in master mode, the MOIFV bit determines the MOSI0 pin output value during the SSL negation period (including the SSL retention period during a burst transfer).

MOIFE Bit (MOSI Idle Value Fixing Enable)

The MOIFE bit fixes the MOSI0 output value when the RSPIA in master mode is in an SSL negation period (including the SSL retention period during a burst transfer). When the MOIFE bit is 0, the RSPIA outputs the last data from the previous serial transfer during the SSL negation period to the MOSI0 pin. When the MOIFE bit is 1, the RSPIA outputs the fixed value set in the MOIFV bit to the MOSI0 pin.

38.2.9 RSPI Slave Select Polarity Register (SSLP)

Address(es): RSPIA0.SSLP 000E 2810h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	SSL3P	SSL2P	SSL1P	SSL0P
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SSL0P	SSL00 Signal Polarity Setting	[Motorola SPI] 0: The SSL00 signal is active low. 1: The SSL00 signal is active high. [TI SSP] 0: The SSL00 signal is active high. 1: The SSL00 signal is active low.	R/W
b1	SSL1P	SSL01 Signal Polarity Setting	[Motorola SPI] 0: The SSL01 signal is active low. 1: The SSL01 signal is active high. [TI SSP] 0: The SSL01 signal is active high. 1: The SSL01 signal is active low.	R/W
b2	SSL2P	SSL02 Signal Polarity Setting	[Motorola SPI] 0: The SSL02 signal is active low. 1: The SSL02 signal is active high. [TI SSP] 0: The SSL02 signal is active high. 1: The SSL02 signal is active low.	R/W
b3	SSL3P	SSL03 Signal Polarity Setting	[Motorola SPI] 0: The SSL03 signal is active low. 1: The SSL03 signal is active high. [TI SSP] 0: The SSL03 signal is active high. 1: The SSL03 signal is active low.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SSLP register is used to set the polarity of SSL0n signals (n = 0 to 3) of the RSPIA. If this register is changed while the SPCR.SPE bit is 1, the operation after the change is not guaranteed.

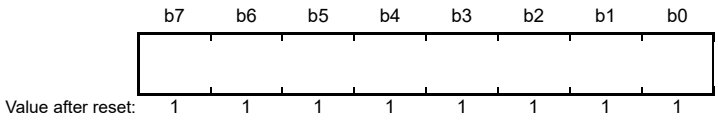
SSLnP Bits (SSL0n Signal Polarity Setting) (n = 0 to 3)

These bits are used to specify the polarity of SSL0n signals. The values of the SSLnP bit indicate the active polarity of SSL0n signals.

Note: SSL00 is different from SSL01 to SSL03. When slave or multi-master, it functions as an input.
For details, refer to section 38.3.3.2, Single Master/Single Slave (with This MCU Acting as Slave), and section 38.3.3.5, Multi-Master/Multi-Slave (with This MCU Acting as Master).

38.2.10 RSPI Bit Rate Register (SPBR)

Address(es): RSPIA0.SPBR 000E 2811h



The SPBR register is used to set the bit rate in master mode. If the value of the SPBR register is changed while both the SPCR.MSTR and SPCR.SPE bits are 1, the operation after the change is not guaranteed.
The bit rate is determined by combinations of the SPBR register setting and the SPCMDm.BRDV[1:0] bit setting.
When the RSPIA is used in slave mode, the bit rate depends on the frequency of the input clock (bit rate satisfying the electrical characteristics should be used) regardless of the settings of the SPBR register and the SPCMDm.BRDV[1:0] bits.
The equation for calculating the bit rate is given below. In the equation, n denotes the SPBR register setting (0, 1, 2, …, 255), and N denotes the BRDV[1:0] bit setting (0, 1, 2, 3).

$$\text{Bit rate} = \frac{f(PCLKA)}{2 \times (n+1) \times 2^{N}}$$

Table 38.5 lists examples of the relationship among the SPBR register settings, the BRDV[1:0] bit settings, and bit rates.

Table 38.5 Relationship among SPBR Register Settings, BRDV[1:0] Bit Settings, and Bit Rates

SPBR (n)	BRDV[1:0] Bits (N)	Division Ratio	Bit Rate				
			PCLKA = 32 MHz	PCLKA = 36 MHz	PCLKA = 40 MHz	PCLKA = 50 MHz	PCLKA = 120 MHz
0	0	2	16.0 Mbps	18.0 Mbps	20.0 Mbps	25.0 Mbps	60.0 Mbps
1	0	4	8.00 Mbps	9.00 Mbps	10.0 Mbps	12.5 Mbps	30.0 Mbps
2	0	6	5.33 Mbps	6.00 Mbps	6.67 Mbps	8.33 Mbps	20.0 Mbps
3	0	8	4.00 Mbps	4.50 Mbps	5.00 Mbps	6.25 Mbps	15.0 Mbps
4	0	10	3.20 Mbps	3.60 Mbps	4.00 Mbps	5.00 Mbps	12.0 Mbps
5	0	12	2.67 Mbps	3.00 Mbps	3.33 Mbps	4.16 Mbps	10.0 Mbps
5	1	24	1.33 Mbps	1.50 Mbps	1.67 Mbps	2.08 Mbps	5.00 Mbps
5	2	48	677 kbps	750 kbps	833 kbps	1.04 Mbps	2.50 Mbps
5	3	96	333 kbps	375 kbps	417 kbps	521 kbps	1.25 Mbps
255	3	4096	7.81 kbps	8.80 kbps	9.78 kbps	12.2 kbps	29.3 kbps

38.2.11 RSPI Sequence Control Register (SPSCR)

Address(es): RSPIA0.SPSCR 000E 2813h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SPSLN[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SPSLN[2:0]	RSPI Sequence Length Setting	b2 b0 Sequence Length — Referenced SPCMD0 to SPCMD7 (No.) 0 0 0: 1 — 0→0→… 0 0 1: 2 — 0→1→0→… 0 1 0: 3 — 0→1→2→0→… 0 1 1: 4 — 0→1→2→3→0→… 1 0 0: 5 — 0→1→2→3→4→0→… 1 0 1: 6 — 0→1→2→3→4→5→0→… 1 1 0: 7 — 0→1→2→3→4→5→6→0→… 1 1 1: 8 — 0→1→2→3→4→5→6→7→0→… The order in which the SPCMD0 to SPCMD7 registers are to be referenced is changed according to the sequence length that is set in these bits. The relationship among the setting of these bits, sequence length, and SPCMD0 to SPCMD7 registers referenced by the RSPIA is shown above. However, the RSPIA in slave mode references SPCMD0.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R

The SPSCR register is used to set the sequence length when the RSPIA operates in master mode. When changing the SPSCR.SPSLN[2:0] bits while both the SPCR.MSTR and SPCR.SPE bits are 1, the bits should be changed while the SPSR.IDLNF flag is 0.

SPSLN[2:0] Bits (RSPI Sequence Length Setting)

The SPSLN[2:0] bits specify a sequence length when the RSPIA in master mode performs sequential operations. The RSPIA in master mode changes SPCMD0 to SPCMD7 registers to be referenced and the order in which they are referenced according to the sequence length that is set in the SPSLN[2:0] bits. For details, refer to (3) Sequence Control in section 38.3.13.1, Master Mode Operation.

In slave mode, SPCMD0 is referred.

38.2.12 RSPI Command Register m (SPCMDm) (m = 0 to 7)

Address(es): RSPIA0.SPCMD0 000E 2814h, RSPIA0.SPCMD1 000E 2818h, RSPIA0.SPCMD2 000E 281Ch, RSPIA0.SPCMD3 000E 2820h, RSPIA0.SPCMD4 000E 2824h, RSPIA0.SPCMD5 000E 2828h, RSPIA0.SPCMD6 000E 282Ch, RSPIA0.SPCMD7 000E 2830h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	SSLA[2:0]			—	—	—	SPB[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SCKDEN	SLNDEN	SPNDEN	LSBF	—	—	—	—	SSLKP	—	—	—	BRDV[1:0]		CPOL	CPHA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CPHA	RSPCK Phase Setting	0: Data sampling on odd edge, data variation on even edge 1: Data variation on odd edge, data sampling on even edge	R/W
b1	CPOL	RSPCK Polarity Setting	0: RSPCK is low when idle 1: RSPCK is high when idle	R/W
b3, b2	BRDV[1:0]	Bit Rate Division Setting	b3 b2 0 0: These bits select the base bit rate 0 1: These bits select the base bit rate divided by 2 1 0: These bits select the base bit rate divided by 4 1 1: These bits select the base bit rate divided by 8	R/W
b6 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R
b7	SSLKP	SSL Signal Level Keeping	0: Negates all SSL signals upon completion of transfer 1: Keeps the SSL signal level from the end of transfer until the beginning of the next access	R/W
b11 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R
b12	LSBF	RSPI LSB First	0: MSB first 1: LSB first	R/W
b13	SPNDEN	RSPI Next-Access Delay Enable	0: A next-access delay of 1 RSPCK + 5 PCLKA 1: A next-access delay is equal to the setting of the RSPI next-access delay register (SPND)	R/W
b14	SLNDEN	SSL Negation Delay Setting Enable	0: In master mode, an SSL negation delay of 1 RSPCK In slave and TI SSP mode, an SSL negation delay of 1 PCLKA 1: An SSL negation delay is equal to the setting of the RSPI slave select negation delay register (SSLND)	R/W
b15	SCKDEN	RSPCK Delay Setting Enable	0: In Motorola SPI mode, an RSPCK delay of 1 RSPCK In TI SSP mode, an RSPCK delay of 0 RSPCK 1: An RSPCK delay is equal to the setting of the RSPI clock delay register (SPCKD)	R/W
b20 to b16	SPB[4:0]	RSPI Data Length Setting	b20 b16 00000 to 00010: Setting prohibited 0 0 0 1 1: 4 bits 0 0 1 0 0: 5 bits 0 0 1 0 1: 6 bits : : 1 1 1 1 0: 31 bits 1 1 1 1 1: 32 bits	R/W
b23 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R
b26 to b24	SSLA[2:0]	SSL Signal Assertion Setting	b26 b24 0 0 0: SSL00 0 0 1: SSL01 0 1 0: SSL02 0 1 1: SSL03 100 to 111: Setting prohibited	R/W
b31 to b27	—	Reserved	These bits are read as 0. The write value should be 0.	R

The SPCMDm register is used to set a transfer format for the RSPIA in master mode. Each channel has eight RSPI command registers (SPCMD0 to SPCMD7). Some of the bits in SPCMD0 register is used to set a transfer mode for the RSPIA in slave mode. The RSPIA in master mode sequentially references SPCMDm register according to the settings in the SPSCR.SPSLN[2:0] bits, and executes the serial transfer that is set in the referenced SPCMDm register.
The SPCMDm register should be set while the transmit buffer is empty (data for the next transfer is not set) and before setting of the data that is to be transmitted when that SPCMDm register is referenced.
SPCMDm that is referenced by the RSPIA in master mode can be checked by means of the SPSSR.SPCP[2:0] bits. If this register is changed while the SPCR.SPE bit is 1, the operation after the change is not guaranteed.

CPHA Bit (RSPCK Phase Setting)

The CPHA bit is used to set an RSPCK phase of the RSPIA in master mode or slave mode. Data communications between RSPIA modules require the same RSPCK phase setting between the modules.
When the SPCR.SPMS is 0 and the SPCR.FRFS bit is 1 in TI SSP mode, setting the CPHA bit to 0 is not effective.

CPOL Bit (RSPCK Polarity Setting)

The CPOL bit is used to set an RSPCK polarity of the RSPIA in master mode or slave mode. Data communications between RSPIA modules require the same RSPCK polarity setting between the modules.

BRDV[1:0] Bits (Bit Rate Division Setting)

The BRDV[1:0] bits are used to determine the bit rate. A bit rate is determined by combinations of the settings in the BRDV[1:0] bits and the SPBR register (refer to section 38.2.10, RSPI Bit Rate Register (SPBR)). The settings in the SPBR register determine the base bit rate. The settings in the BRDV[1:0] bits are used to select a bit rate which is obtained by dividing the base bit rate by 1, 2, 4, or 8. In the SPCMDm register, different BRDV[1:0] bit settings can be specified. This enables execution of serial transfers at a different bit rate for each command.

SSLKP Bit (SSL Signal Level Keeping)

When the RSPIA in master mode performs a serial transfer, the SSLKP bit specifies whether the SSL0n (n = 0 to 3) signal level for the current command is to be kept or negated between the SSL0n negation timing associated with the current command and the SSL0n assertion timing associated with the next command.
Setting the SSLKP bit to 1 enables a burst transfer. For details, refer to section 38.3.13.1, Master Mode Operation (4) Burst Transfer.
When using the RSPIA in slave mode, the SSLKP bit should be set to 0.

LSBF Bit (RSPI LSB First)

The LSBF bit is used to set the data format of the RSPIA in master mode or slave mode to MSB first or LSB first.

SPNDEN Bit (RSPI Next-Access Delay Enable)

The SPNDEN bit is used to set the period from the time the RSPIA in master mode terminates a serial transfer and sets the SSL0n signal inactive until the RSPIA enables the SSL0n signal assertion for the next access (next-access delay). If the SPNDEN bit is 0, the RSPIA sets the next-access delay to 1 RSPCK + 5 PCLKA. If the SPNDEN bit is 1, the RSPIA inserts a next-access delay in compliance with the SPND setting.
When using the RSPIA in slave mode, the SPNDEN bit should be set to 0.

SLNDEN Bit (SSL Negation Delay Setting Enable)

[Motorola SPI]
The SLNDEN bit is used to set the period from the time the RSPIA in master mode stops RSPCK oscillation until the RSPIA sets the SSL signal inactive (SSL negation delay). If the SLNDEN bit is 0, the RSPIA sets the SSL negation delay to 1 RSPCK. If the SLNDEN bit is 1, the RSPIA negates the SSL signal at an SSL negation delay in compliance

with the SSLND setting.
When using the RSPIA in slave mode, set the SLNDL[2:0] bits to 000b.

[TI SSP]
The SLNDEN bit is used to set the period from the time the RSPIA in master mode stops RSPCK oscillation until the RSPIA sets the OE signal inactive, or from the detection of a final RSPCK edge to the negation of the OE signal by the RSPIA in slave mode. If the SLNDEN bit is 0, the SSL negation delay is 1 RSPCK in master mode and 1 PCLKA in slave mode. If the SLNDEN bit is 1, the RSPIA negates the SSL signal at an SSL negation delay in compliance with the SSLND setting.

[Clock synchronous]
When using the RSPIA in slave mode, set this bit to 0.

SCKDEN Bit (RSPCK Delay Setting Enable)

[Motorola SPI]
The SCKDEN bit is used to set the period from the point when the RSPIA in master mode activates the SSL signal until the RSPCK starts oscillation (RSPCK delay). If the SCKDEN bit is 0, the RSPIA sets the RSPCK delay to 1 RSPCK. If the SCKDEN bit is 1, the RSPIA starts the oscillation of RSPCK at an RSPCK delay in compliance with the SPCKD setting.

[TI SSP]
The SCKDEN bit is used to set the period from the point when the RSPIA in master mode activates the SSL signal until the RSPCK starts oscillation (RSPCK delay) and to the negation of the SSL0n signal. If the SCKDEN bit is 0, the RSPIA does not set the RSPCK delay. If the SCKDEN bit is 1, the RSPIA starts the oscillation of RSPCK at an RSPCK delay in compliance with the SPCKD setting.
When using the RSPIA in slave mode, the SCKDEN bit should be set to 0.

SPB[4:0] Bits (RSPI Data Length Setting)

The SPB[4:0] bits are used to set a transmit data length for the RSPIA in master mode or slave mode.

SSLA[2:0] Bits (SSL Signal Assertion Setting)

The SSLA[2:0] bits control the SSL0n signal assertion when the RSPIA performs serial transfers in master mode. Setting the SSLA[2:0] bits controls the assertion for the SSL0n signal. When an SSL0n (n = 0 to 3) signal is asserted, its polarity is determined by the set value in the corresponding SSLP. When the SSLA[2:0] bits are set to 000b in multi-master mode, serial transfers are performed with all the SSL signals in the negated state (as the SSL00 pin acts as input).
When using the RSPIA in slave mode, set the SSLA[2:0] bits to 000b.

38.2.13 RSPI Data Control Register (SPDCR)

Address(es): RSPIA0.SPDCR 000E 2840h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	SPFC[1:0]		—	—	—	DINV	SPRDTD	—	—	BYSW
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BYSW	Byte Swap	0: Byte swapping of SPDR data disabled 1: Byte swapping of SPDR data enabled	R/W
b2, b1	—	Reserved	These bits are read as 0. The write value should be 0.	R
b3	SPRDTD	RSPI Receive/Transmit Data Select	0: SPDR values are read from the receive buffer 1: SPDR values are read from the transmit buffer	R/W
b4	DINV	Transfer Data Invert	0: Not invert serial data 1: Invert serial data	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R
b9, b8	SPFC[1:0]	Number of Frames Setting	b9 b8 0 0: 1 frame 0 1: 2 frames 1 0: 3 frames 1 1: 4 frames	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R

The SPDCR register is used to control the data format of the RSPIA. If this register is changed while the SPCR.SPE bit is 1, the operation after the change is not guaranteed.

BYSW Bit (Byte Swap)

The BYSW bit is used to swap a transmit/receive data in byte units. A data alignment after byte swap is different by a data length (setting of SPCMDm.SPB[4:0]).

When using byte swap, set the data length (setting of SPCMDm.SPB[4:0]) to 32 bits or 16 bits. Other case of data length (i.e. 4 to 15, 17 to 31-bit length), byte swap is not guaranteed. For the arrangement of data before and after swapping data lengths of 32 bits and 16 bits, refer to section 38.3.4.4, Byte Swap Transmission and section 38.3.4.5, Byte Swap Reception.

If this bit is changed while the SPCR.SPPE bit is 1, the operation is not guaranteed.

SPRDTD Bit (RSPI Receive/Transmit Data Select)

The SPRDTD bit is used to whether the SPDR register reads values from the receive buffer or from the transmit buffer. If reading is from the transmit buffer, the value written to the SPDR register immediately beforehand is read.

DINV Bit (Transfer Data Invert)

The DINV bit is used to invert transmit data and receive data.

When the DINV bit is set to 1, transmit buffer (SPTX) data is inverted to invert transmit data and receive data, and then the inverted data is stored in the receive buffer (SPRX). The parity bit is the value corresponding to the inverted transmission/reception data.

SPFC[1:0] Bits (Number of Frames Setting)

The SPFC[1:0] bits are used for the condition to set the SPSR.SPCF flag in slave receive-only mode.

For details on the SPCF flag setting conditions, refer to [Setting condition] of the SPCF flag in section 38.2.16.

Note that these bits are not effective except in the slave receive-only mode.

38.2.14 RSPI FIFO Control Register (SPFCR)

Address(es): RSPIA0.SPFCR 000E 2844h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	TTRG[1:0]		—	—	—	—	—	—	RTRG[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	RTRG[1:0]	Receive FIFO Threshold Setting	b1 b0 0 0: When the receive FIFO fill level is 0. 0 1: When the receive FIFO fill level is 1. 1 0: When the receive FIFO fill level is 2. 1 1: When the receive FIFO fill level is 3.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b9, b8	TTRG[1:0]	Transmit FIFO Threshold Setting	b9 b8 0 0: When the transmit FIFO free level is 0. 0 1: When the transmit FIFO free level is 1. 1 0: When the transmit FIFO free level is 2. 1 1: When the transmit FIFO free level is 3.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R

The SPFCR register is used to control the FIFO threshold. If this register is changed while the SPCR.SPE bit is 1, the operation after the change is not guaranteed.

RTRG[1:0] Bits (Receive FIFO Threshold Setting)

The RTRG[1:0] bits are used to select receive FIFO threshold. When the number of data stored in the receive FIFO is greater than the threshold value set by RTRG[1:0], the receive buffer full flag is set.

TTRG[1:0] Bits (Transmit FIFO Threshold Setting)

The TTRG[1:0] bits are used to select transmit FIFO threshold. When the number of empty stages in the transmit FIFO is greater than the threshold value set in TTRG[1:0], the transmit buffer empty flag is set.

38.2.15 RSPI Sequence Status Register (SPSSR)

Address(es): RSPIA0.SPSSR 000E 2851h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	SPECM[2:0]			—	SPCP[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SPCP[2:0]	RSPI Command Pointer	b2 b0 0 0 0: SPCMD0 0 0 1: SPCMD1 0 1 0: SPCMD2 0 1 1: SPCMD3 1 0 0: SPCMD4 1 0 1: SPCMD5 1 1 0: SPCMD6 1 1 1: SPCMD7	R
b3	—	Reserved	This bit is read as 0.	R
b6 to b4	SPECM[2:0]	RSPI Error Command	b6 b4 0 0 0: SPCMD0 0 0 1: SPCMD1 0 1 0: SPCMD2 0 1 1: SPCMD3 1 0 0: SPCMD4 1 0 1: SPCMD5 1 1 0: SPCMD6 1 1 1: SPCMD7	R
b7	—	Reserved	This bit is read as 0.	R

The SPSSR register indicates the sequence control status when the RSPIA operates in master mode.
Any writing to the SPSSR register is ignored.

SPCP[2:0] Bits (RSPI Command Pointer)

The SPCP[2:0] bits indicate the SPCMDm register that is currently pointed to by the pointer during sequence control by the RSPIA.
For the RSPIA's sequence control, refer to section 38.3.13.1, Master Mode Operation.

SPECM[2:0] Bits (RSPI Error Command)

The SPECM[2:0] bits indicate the SPCMDm register that is specified by the SPCP[2:0] bits when an error is detected during sequence control by the RSPIA. The RSPIA updates the SPECM[2:0] bits only when an error is detected. If the SPSR.OVRF, MODF, and PERF flags are 0 and there is no error, the values of the SPECM[2:0] bits have no meaning.
For the RSPIA's error detection function, refer to section 38.3.10, Error Detection. For the RSPIA's sequence control, refer to section 38.3.13.1, Master Mode Operation.

38.2.16 RSPI Status Register (SPSR)

Address(es): RSPIA0.SPSR 000E 2852h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SPRF	SPCF	SPTEF	UDRF	PERF	MODF	IDLNF	OVRF	RRDYF	—	—	—	—	—	—	—
Value after reset:	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0.	R
b7	RRDYF	Receive Data Ready Flag	0: Receive data ready is not detected 1: Receive data ready is detected	R
b8	OVRF	Overrun Error Flag	0: No overrun error occurs 1: An overrun error occurs	R
b9	IDLNF	Idle Flag	0: RSPIA is in the idle state 1: RSPIA is in the transfer state	R
b10	MODF	Mode Fault Error Flag	0: Neither a mode fault error nor an underrun error occurs 1: A mode fault error or an underrun error occurs	R
b11	PERF	Parity Error Flag	0: No parity error occurs 1: A parity error occurs	R
b12	UDRF	Underrun Error Flag	This flag is used with the MODF flag to check the state in terms of mode fault errors and underrun errors. MODF UDRF 0 x: Neither a mode fault error nor an underrun error occurs 1 0: A mode fault error occurs 1 1: An underrun error occurs	R
b13	SPTEF	Transmit Buffer Empty Flag	0: The number of empty stages in the transmit FIFO ≤ the value set by the SPFCR.TTRG[1:0] bits 1: The number of empty stages in the transmit FIFO > the value set by the SPFCR.TTRG[1:0] bits	R
b14	SPCF	Communication End Flag	0: RSPIA is not communicating or communicating 1: RSPIA communication is completed	R
b15	SPRF	Receive Buffer Full Flag	0: The number of data stored in the receive FIFO ≤ number of frames set by the SPFCR.RTRG[1:0] bits 1: The number of data stored in the receive FIFO > number of frames set by the SPFCR.RTRG[1:0] bits	R

x: don't care

The SPSR register is used to store flags that indicate RSPIA's operating status.

RRDYF Flag (Receive Data Ready Flag)

The RRDYF flag indicates a certain period of time has elapsed while the number of data stored in the receive FIFO during communication (SPCR.SPE = 1) is equal to or less than the receive FIFO threshold value.
This flag is set to 0 when the receive operation is not performed (SPCR.CMMD[1:0] = 01b).

[Setting condition]

When all the following conditions are satisfied.

- SPDRCSR register ≠ 00h.
- After the receive FIFO has been written, when the number of data stored in the receive FIFO is equal to or less than the receive FIFO threshold value and the value set by the SPDRCSR register has elapsed

[Clearing condition]

When 1 is written to the SPSCLR.RRDYFC bit.

OVRF Flag (Overrun Error Flag)

The OVRF flag indicates the occurrence of an overrun error. In master mode (when the SPCR.MSTR bit is 1) and when the RSPCK auto-stop function is enabled (the SPCR.SCKASE bit is 1), an overrun error does not occur; accordingly this flag does not become 1. For details, refer to section 38.3.10.1, Overrun Error.

[Setting condition]

When serial transfer is completed with no free space in the receive FIFO under any of the following conditions.

- The SPCR.CMMD[1:0] bits are 00b (transmit-receive master mode or transmit-receive slave mode).
- The SPCR.CMMD[1:0] bits are 10b (receive-only master mode or receive-only slave mode).

[Clearing condition]

When 1 is written to the SPSCLR.OVRFC bit

IDLNF Flag (Idle Flag)

The IDLNF flag indicates the transfer status of the RSPIA.

[Setting condition]

(1) Transmit-receive master mode or transmit-only master mode

- When none of the conditions are satisfied in transmit-receive master mode or transmit-only master mode under the [Clearing condition] below.

(2) Transmit-only master mode

- When 1 is written to the SPRMCR.START bit.

(3) Slave mode

- The SPCR.SPE bit is 1 (enables the RSPI function)

[Clearing condition]

(1) Transmit-receive master mode or transmit-only master mode

Communication status: No. 1-1 to 1-6 (for details, refer to Table 38.4 RSPI Communication Status)

- When the SPCR.SPE bit is set to 0 (disables the RSPI function)
- When all of the following conditions are satisfied.
  (a) The the next transmit data is not set in the transmit buffer (SPTX)
  (b) The SPSSR.SPCP[2:0] bits are 000b (beginning of sequence control)
  (c) The operations up to the next-access delay have finished (the master main state machine has entered the idle state)

(2) Transmit-only master mode

Communication status: No. 1-7 to 1-9 (for details, refer to Table 38.4 RSPI Communication Status)

- When the SPCR.SPE bit is set to 0 (disables the RSPI function)
- When any of the following conditions is satisfied.
  (a) When the SPRMCR.RFC[4:0] bits are 00000b, after 1 is written to the SPRMCR.TERM bit, the operations up to the next-access delay have finished (the master main state machine has entered the idle state)
  (b) When the SPRMCR.RFC[4:0] bits are other than 00000b, after 1 is written to the SPRMCR.TERM bit, the operations up to the next-access delay have finished (the master main state machine has entered the idle state)
  (c) When the SPRMCR.RFC[4:0] bits are other than 00000b, after the operation is completed for the number of received frames set by the SPRMCR.RFC[4:0] bits, the operations up to the next-access delay have finished (the master main state machine has entered the idle state)

(3) Slave mode

Communication status: No. 0-1 to 0-9 (for details, refer to Table 38.4 RSPI Communication Status)

- When the SPCR.SPE bit is set to 0 (disables the RSPI function)

MODF Flag (Mode Fault Error Flag)

Indicates the occurrence of a mode fault error or an underrun error. The UDRF flag indicates whether the error is a mode fault error or an underrun error.

[Setting condition]

(1) Multi-master mode

- When the input level of the SSL0n pin changes to the active level while the SPCR.MSTR bit is 1 (master mode) and the SPCR.MODFEN bit is 1 (mode fault error detection is enabled), the RSPIA detects a mode fault error

(2) Slave mode (SPI operation, Motorola SPI)

When any of the following conditions are satisfied.

- When the SSL00 pin is negated before the RSPCK cycle necessary for data transfer ends while the SPCR.MSTR bit is 0 (slave mode) and the SPCR.MODFEN bit is 1 (mode fault error detection is enabled), the RSPIA detects a mode fault error
- When the serial transfer starts while the SPCR.SPE bit is 1 (enables the RSPI function) and the transmit data are not ready for output, the RSPIA detects an underrun error

(3) Slave mode (SPI operation, TI SSP)

When any of the following conditions are satisfied.

- When the SSL00 pin is asserted before the RSPCK cycle necessary for data transfer ends while the SPCR.MSTR bit is 0 (slave mode) and the SPCR.MODFEN bit is 1 (mode fault error detection is enabled), the RSPIA detects a mode fault error
- When the serial transfer starts while the SPCR.SPE bit is 1 (enables the RSPI function) and the transmit data are not ready for output, the RSPIA detects an underrun error

The active level of the SSL0n signal is determined by the SSLP.SSLnP bit (SSL signal polarity setting bit).

[Clearing condition]

When 1 is written to the SPSCLR.MODFC bit

PERF Flag (Parity Error Flag)

Indicates the occurrence of a parity error.

[Setting condition]

When the serial transfer ends and a parity error is detected while the SPCR.SPPE bit is 1 under any of the following conditions.

- The SPCR.CMMD[1:0] bits are 00b (transmit-receive master mode or transmit-receive slave mode)
- The SPCR.CMMD[1:0] bits are 10b (receive-only master mode or receive-only slave mode)

[Clearing condition]

When 1 is written to the SPSCLR.PERFC bit

UDRF Flag (Underrun Error Flag)

When a mode fault error or an underrun error has occurred, the UDRF flag indicates whether the error is a mode fault error or an underrun error.

[Setting condition]

When the serial transfer starts while the SPCR.MSTR bit is 0, the SPCR.CMMD[1:0] bits are 00b or 01b (transmit-receive slave mode or transmit-only slave mode), the SPCR.SPE bit is 1 (enables the RSPI function) and the transmit data are not ready for output, the RSPIA detects an underrun error

[Clearing condition]

When 1 is written to the SPSCLR.UDRFC bit

SPTEF Flag (Transmit Buffer Empty Flag)

Indicates whether the transmit buffer (SPTXn, n = 0 to 3) in the RSPI data register (SPDR) has valid data.

[Setting condition]

When any of the following conditions are satisfied.

- When the SPCR.SPE bit is 0 (disables the RSPI function)
- When the number of empty transmission FIFO stages is greater than the threshold value set in SPFCR.TTRG[1:0]
- When 1 is written to the SPFCLR.FCLR bit

[Clearing condition]

When any of the following conditions are satisfied.

- At the final access when transmit data is written to SPDR (SPTXn) in one processing routine using DTC/DMAC
- When 1 is written to the SPSCLR.SPTEFC bit

The SPDR register can be set only when the SPSR.SPTEF flag is 1. The data in the transmit buffer (SPTXn) is not updated when the SPDR register is set while the SPSR.SPTEF flag is 0.

SPCF Flag (Communication End Flag)

Indicates communication end status of RSPI. It turns 1 at communication end and turns 0 at starting next communication.

[Setting condition]

(1) Transmit-receive master mode or transmit-only master mode

Communication status: No. 1-1 to 1-6 (for details, refer to Table 38.4 RSPI Communication Status)

When all of the following conditions are satisfied.

- The next transmit data is not set in the transmission buffer (SPTXn)
- The SPSSR.SPCP[2:0] bits are 000b (beginning of sequence control)
- The operations up to the next-access delay have finished (the master main state machine has entered the idle state)

(2) Transmit-only master mode

Communication status: No. 1-7 to 1-9 (for details, refer to Table 38.4 RSPI Communication Status)

When any of the following conditions are satisfied.

- When the SPRMCR.RFC[4:0] bits are 00000b, after 1 is written to the SPRMCR.TERM bit, the operations up to the next-access delay have finished (the master main state machine has entered the idle state)
- When the SPRMCR.RFC[4:0] bits are other than 00000b, after 1 is written to the SPRMCR.TERM bit, the operations up to the next-access delay have finished (the master main state machine has entered the idle state)

- When the SPRMCR.RFC[4:0] bits are other than 00000b, after the operation is completed for the number of received frames set by the SPRMCR.RFC[4:0] bits, the operations up to the next-access delay have finished (the master main state machine has entered the idle state)

(3) Transmit-receive slave mode or transmit-only slave mode (SPI operation, Motorola SPI)
Communication status: No. 0-1, 0-4 (for details, refer to Table 38.4 RSPI Communication Status)
When all of the following conditions are satisfied.

- The next transmit data is not set in the transmission buffer (SPTXn)
- The transmission shift register is empty (RSPIA does not perform serial transfer)
- The SSL00 input is negated

(4) Transmit-receive slave mode or transmit-only slave mode (SPI operation, TI SSP)
Communication status: No. 0-2, 0-5 (for details, refer to Table 38.4 RSPI Communication Status)
When all of the following conditions are satisfied.

- The next transmit data is not set in the transmission buffer (SPTXn)
- The transmission shift register is empty (RSPIA does not perform serial transfer)
- When the SSL negation delay is completed

(5) Transmit-receive slave mode or transmit-only slave mode (clock synchronous operation)
Communication status: No. 0-3, 0-6 (for details, refer to Table 38.4 RSPI Communication Status)
When all of the following conditions are satisfied.

- The next transmit data is not set in the transmission buffer (SPTXn)
- The transmission shift register is empty (RSPIA does not perform serial transfer)
- The last even edge of RSPCK of the frame was detected (SPCMDm.CPHA = 1)

(6) Receive-only slave mode (SPI operation, Motorola SPI)
Communication status: No. 0-7 (for details, refer to Table 38.4 RSPI Communication Status)

- SSL00 input was negated after getting frames for SPDCR.SPFC[1:0] set value in the receive buffer (SPRX)

(7) Receive-only slave mode (SPI operation, TI SSP)
Communication status: No. 0-8 (for details, refer to Table 38.4 RSPI Communication Status)

- SSL negation delay is completed after getting frames for SPDCR.SPFC[1:0] set value in the receive buffer (SPRX).

(8) Receive-only slave mode (clock synchronous operation)
Communication status: No. 0-9 (for details, refer to Table 38.4 RSPI Communication Status)

- The last even edge of RSPCK of the last frame received for SPFC[1:0] set value was detected (SPCMDm.CPHA = 1)

[Clearing condition]
(1) Transmit-receive master mode or transmit-only master mode
Communication status: No. 1-1 to 1-6 (for details, refer to Table 38.4 RSPI Communication Status)
When any of the following conditions are satisfied.

- The next transmit data is set in the transmission buffer (SPTXn)
- When 1 is written to the SPSCLR.SPCFC bit

(2) Transmit-only master mode

Communication status: No. 1-7 to 1-9 (for details, refer to Table 38.4 RSPI Communication Status)
When any of the following conditions are satisfied.

- When 1 is written to the SPRMCR.START bit while the SPCR.SPE bit is 1
- When 1 is written to the SPSCLR.SPCFC bit

(3) Transmit-receive slave mode or transmit-only slave mode

Communication status: No. 0-1 to 0-6 (for details, refer to Table 38.4 RSPI Communication Status)
When any of the following conditions are satisfied.

- The next transmit data is set in the transmission buffer (SPTXn)
- When 1 is written to the SPSCLR.SPCFC bit

(4) Receive-only slave mode (SPI operation)

Communication status: No. 0-7, 0-8 (for details, refer to Table 38.4 RSPI Communication Status)
When any of the following conditions are satisfied.

- The SSL00 input assertion of next data is detected
- When 1 is written to the SPSCLR.SPCFC bit

(5) Receive-only slave mode (clock synchronous operation)

Communication status: No. 0-9 (for details, refer to Table 38.4 RSPI Communication Status)
When any of the following conditions are satisfied.

- The first edge of RSPCK of the next data was detected
- When 1 is written to the SPSCLR.SPCFC bit

SPRF Flag (Receive Buffer Full Flag)

Indicates whether the receive buffer (SPRX) in the RSPI data register has valid data.

[Setting condition]

When the number of data stored in the receive FIFO is greater than the number of frames set in the SPFCR.RTRG[1:0] bits in transmit-receive mode or receive-only mode.
Note that the SPRF flag does not become 1 when the OVRF flag is 1 (Refer to section 38.3.10, Error Detection, for details).

[Clearing condition]

When any of the following conditions are satisfied.

- At the final access when receive data is read from SPDR register (SPRXn, n = 0 to 3) in one processing routine using DTC/DMAC
- When 1 is written to the SPSCLR.SPRFC bit
- When 1 is written to the SPFCLR.FCLR bit

38.2.17 RSPI Transmit FIFO Status Register (SPTFSR)

Address(es): RSPIA0.SPTFSR 000E 2858h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	FREE[2:0]		
Value after reset:	0	0	0	0	0	1	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	FREE[2:0]	Transmit FIFO Free Level	b2 b0 0 0 0: Number of empty stages 0 0 0 1: Number of empty stages 1 0 1 0: Number of empty stages 2 0 1 1: Number of empty stages 3 1 0 0: Number of empty stages 4	R
b7 to b3	—	Reserved	These bits are read as 0.	R

FREE[2:0] Bits (Transmit FIFO Free Level)

Indicate the number of empty transmit FIFO stages. By setting the SPCR.SPE bit to 0, the FREE[2:0] bits will be 100b.

38.2.18 RSPI Receive FIFO Status Register (SPRFSR)

Address(es): RSPIA0.SPRFSR 000E 285Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	FILL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	FILL[2:0]	Receive FIFO Fill Level	b2 b0 0 0 0: Number of stored stages 0 0 0 1: Number of stored stages 1 0 1 0: Number of stored stages 2 0 1 1: Number of stored stages 3 1 0 0: Number of stored stages 4	R
b7 to b3	—	Reserved	These bits are read as 0.	R

FILL[2:0] Bits (Receive FIFO Fill Level)

Indicate the number of stored receive FIFO stages. By setting the SPCR.SPE bit to 0, the FILL[2:0] bits will be 000b.

38.2.19 RSPI Status Clear Register (SPSCLR)

Address(es): RSPIA0.SPSCLR 000E 286Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SPRFC	SPCFC	SPTEFC	UDRFC	PERFC	MODFC	—	OVRFC	RRDYFC	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b7	RRDYFC	RSPI Receive Data Ready Flag Clear	By writing 1, the RSPI receive data ready flag can be cleared. The read value is always 0.	W
b8	OVRFC	Overrun Error Flag Clear	By writing 1, the overrun error flag can be cleared. The read value is always 0.	W
b9	—	Reserved	This bit is read as 0. The write value should be 0.	R
b10	MODFC	Mode Fault Error Flag Clear	By writing 1, the mode fault error flag can be cleared. The read value is always 0.	W*1
b11	PERFC	Parity Error Flag Clear	By writing 1, the parity error flag can be cleared. The read value is always 0.	W
b12	UDRFC	Underrun Error Flag Clear	By writing 1, the underrun error flag can be cleared. The read value is always 0.	W*1, *2
b13	SPTEFC	Transmit Buffer Empty Flag Clear	By writing 1, the transmit buffer empty flag can be cleared. The read value is always 0.	W
b14	SPCFC	Communication End Flag Clear	By writing 1, the communication end flag can be cleared. The read value is always 0.	W
b15	SPRFC	Receive Buffer Full Flag Clear	By writing 1, the receive buffer full flag can be cleared. The read value is always 0.	W

Note 1. Before setting the MODFC and UDRFC, make sure that the SPSR.MODF and UDRF flags are set to 1.
Note 2. When the UDRF flag is cleared, the MODF flag is cleared at the same time (MODFC = 1).

The SPSCLR register is used to clear the status flags in the SPSR register that indicates the operating status of RSPIA.

38.2.20 RSPI FIFO Clear Register (SPFCLR)

Address(es): RSPIA0.SPFCLR 000E 286Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	FCLR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	FCLR	FIFO Clear	By writing 1, the pointer in the FIFO and the stored data are initialized. The read value is always 0.	W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R

The SPFCLR register is used to clear the FIFO. If the value of the SPFCLR register is changed while the SPCR.SPE bit is 1, the operation after the change is not guaranteed.

FCLR Bit (FIFO Clear)

Writing 1 to this bit initializes the pointer and stored data in the transmit/receive FIFO.

38.3 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.

38.3.1 Overview of RSPI Operations

The RSPIA is capable of serial transfers in the following five modes.

(1) Slave mode (SPI operation)
(2) Single-master mode (SPI operation)
(3) Multi-master mode (SPI operation)
(4) Slave mode (clock synchronous operation)
(5) Master mode (clock synchronous operation).

A particular mode of the RSPIA can be selected by using the MSTR, MODFEN, SPMS, and FRFS bits in SPCR. Table 38.6 lists the relationship between RSPI modes and SPCR settings, and a description of each mode.

Table 38.6 Relationship between RSPI Modes and SPCR Settings and Description of Each Mode

	SPI Operation			Clock Synchronous Operation	
Mode	**Slave**	**Single-Master**	**Multi-Master**	**Slave**	**Master**
MSTR bit setting	0	1	1	0	1
MODFEN bit setting	0 or 1	0	1	0	0
SPMS bit setting	0	0	0	1	1
FRFS bit setting	Enabled	Enabled	Enabled	Disabled	Disabled
RSPCK0 signal	Input	Output	Output/Hi-Z*1	Input	Output
MOSI0 signal	Input	Output	Output/Hi-Z*1	Input	Output
MISO0 signal	Output/Hi-Z*2	Input	Input	Output	Input
SSL00 signal	Input	Output	Input	Hi-Z (not used)	Hi-Z (not used)
SSL01 to SSL03 signals	Hi-Z (not used)	Output	Output/Hi-Z*1	Hi-Z (not used)	Hi-Z (not used)
Output pin mode	CMOS/open-drain				
SSL polarity change function	Supported	Supported	Supported	—	—
Transfer rate	Up to PCLKA/2				
Clock source	RSPCK input	On-chip baud rate generator	On-chip baud rate generator	RSPCK input	On-chip baud rate generator
Clock polarity	Two				
Clock phase	Two*3	Two*3	Two*3	One (CPHA = 1)	Two
First transfer bit	MSB/LSB				
Transmit data length	4 to 32 bits				
Burst transfer	Possible (CPHA = 1)	Possible (CPHA = 0,1)	Possible (CPHA = 0,1)	—	—
RSPCK delay control	Not supported	Supported	Supported	Not supported	Supported
SSL negation delay control	Not supported*4	Supported	Supported	Not supported	Supported
Next-access delay control	Not supported	Supported	Supported	Not supported	Supported
Transfer activation method	SSL input active or RSPCK oscillation	Transmit buffer is written to when a transmit buffer empty interrupt request is generated or the SPTEF flag is 1	Transmit buffer is written to when a transmit buffer empty interrupt request is generated or the SPTEF flag is 1	RSPCK oscillation	Transmit buffer is written to when a transmit buffer empty interrupt request is generated or the SPTEF flag is 1
Sequence control	Not supported	Supported	Supported	Not supported	Supported
Transmit buffer empty detection	Supported*5	Supported	Supported	Supported*5	Supported
Receive buffer full detection	Supported*6				
Overrun error detection	Supported*6	Supported*6, *7	Supported*6, *7	Supported*6	Supported*6, *7
Parity error detection	Supported*6, *8				
Mode fault error detection	Supported (MODFEN = 1)	Not supported	Supported	Not supported	Not supported
Underrun error detection	Supported*5	Not supported	Not supported	Supported*5	Not supported

Note 1. Motorola SPI: When SSL00 is asserted by another master device, the pin becomes Hi-Z.
TI SSP: When the SPCR.SPE bit is 1, the pin becomes Hi-Z from the time SSL00 is asserted by another master device until the end of communication.

Note 2. Motorola SPI: When SSL00 is negated or the SPCR.SPE bit is 0, the pin becomes Hi-Z.
TI SSP: The pin becomes Hi-Z when the RSPIA is not communicating (from the end of communication to the next assertion of SSL00) or when the SPCR.SPE bit is 0.

Note 3. In TI SSP mode, setting the SPCMDm.CPHA bit to 0 is not effective.

Note 4. Available only in TI SSP mode.

Note 5. When RSPIA is receive-only slave mode (refer to Table 38.4 RSPI Communication Status), transmit buffer empty error detection and underrun error detection are not performed.

Note 6. When RSPIA is transmit-only master mode or transmit-only slave mode (refer to Table 38.4 RSPI Communication Status), receiver buffer full detection, overrun error detection, and parity error detection are not performed.

Note 7. When the SPCR.SCKASE bit is 1, overrun error detection does not proceed.

Note 8. When the SPCR.SPPE bit is 0, parity error detection is not performed.

38.3.2 Controlling RSPI Pins

The RSPIA in single-master mode (SPI operation) or multi-master mode (SPI operation) 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 listed in Table 38.7.

Table 38.7 MOSI Signal Value Determination during SSL Negation Period

MOIFE Bit	MOIFV Bit	MOSI Signal Value during SSL Negation Period
0	0, 1	Final data from previous transfer
1	0	Low
1	1	High

38.3.3 RSPI System Configuration Examples

This configuration example describes that the low level of SSL0n signals is the active level.
When connecting and using in a multi-slave or multi-master mode, the transfer format of the connected device should be unified to either Motorola SPI or TI SSP.

38.3.3.1 Single Master/Single Slave (with This MCU Acting as Master)

Figure 38.5 shows a single-master/single-slave RSPI system configuration example when this MCU is used as a master. In the single-master/single-slave configuration, the SSL00 to SSL03 output of this MCU (master) are not used. The SSL input of the SPI slave is fixed to the low level, and the SPI slave is maintained in a select state. In the transfer format corresponding to the case where the SPCMDm.CPHA bit 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 SSL0n output of this MCU should be connected to the SSL input of the slave device.
This MCU (master) drives the RSPCK0 and MOSI0. The SPI slave drives the MISO.

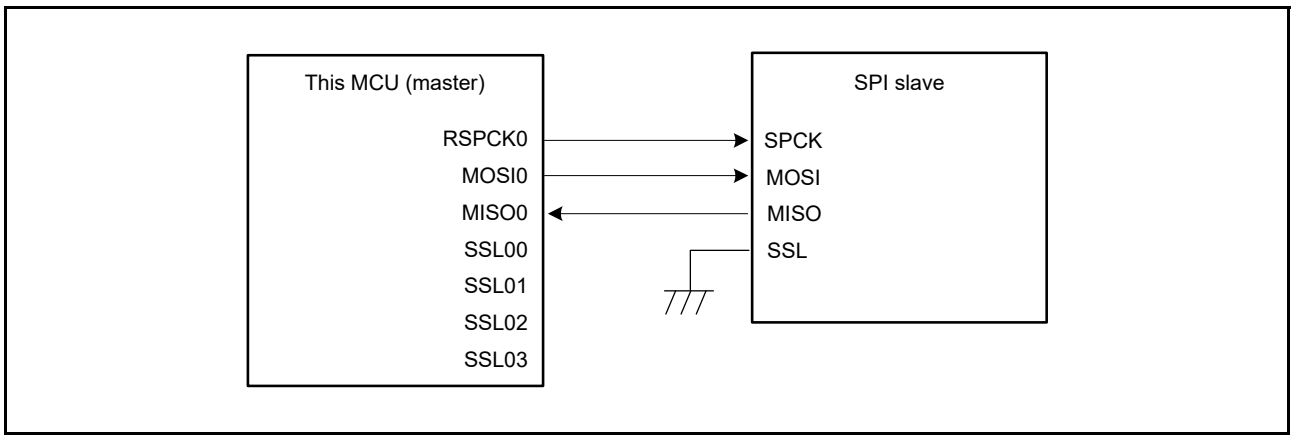


Figure 38.5 Single-Master/Single-Slave Configuration Example (This MCU = Master)

38.3.3.2 Single Master/Single Slave (with This MCU Acting as Slave)

Figure 38.6 shows a single-master/single-slave RSPI system configuration example when this MCU is used as a slave. When this MCU is to operate as a slave, the SSL00 pin is used as SSL input. The SPI master drives the SPCK and MOSI. This MCU (slave) drives the MISO0. When SSL00 is at the non-active level, the pin state is Hi-Z.
In the single-slave configuration in which the SPCMDm.CPHA bit is set to 1, set the SPCR.FRFS bit to 0, and set the SPCR.SPMS bit to 0, the SSL00 input of this MCU (slave) is fixed to the low level, this MCU (slave) is maintained in a select state, and in this manner it is possible to execute serial transfer (Figure 38.7). Note that the communication end event and the communication end interrupt do not output when SSL00 input was fixed as Figure 38.7.

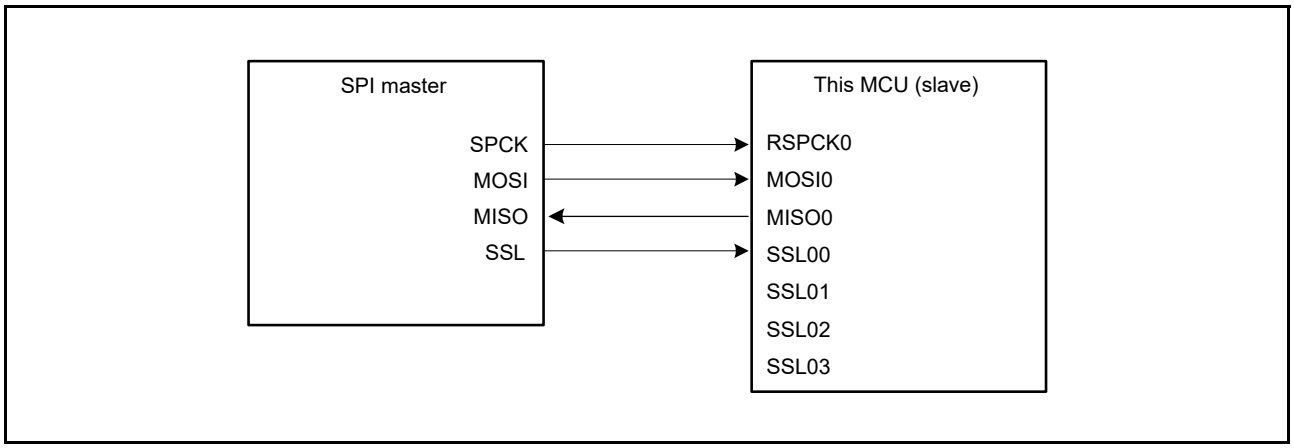


Figure 38.6 Single-Master/Single-Slave Configuration Example (This MCU = Slave, CPHA = 0)

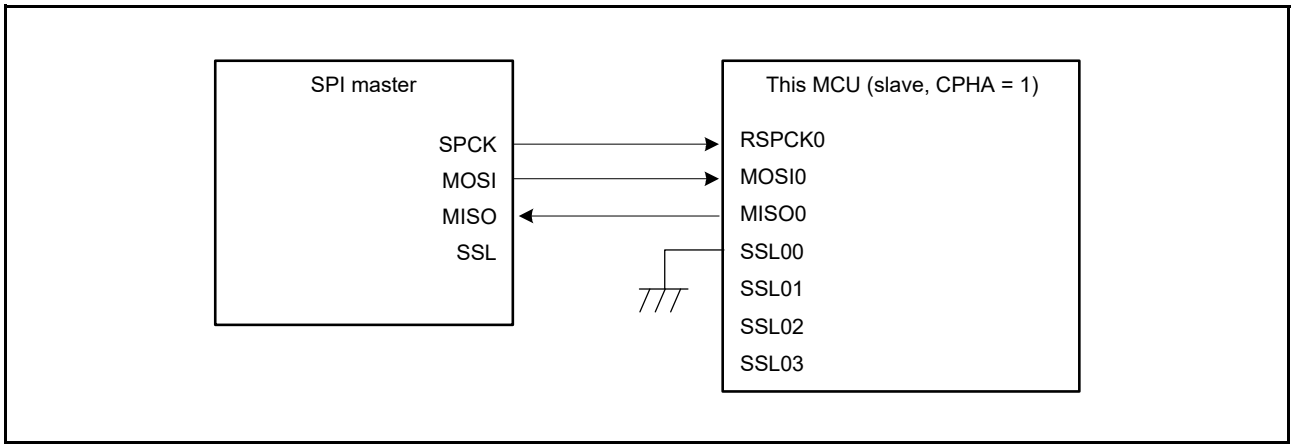


Figure 38.7 Single-Master/Single-Slave Configuration Example (This MCU = Slave, CPHA = 1)

38.3.3.3 Single Master/Multi-Slave (with This MCU Acting as Master)

Figure 38.8 shows a single-master/multi-slave RSPI system configuration example when this MCU is used as a master. In the example of Figure 38.8, the RSPI system is comprised of this MCU (master) and four slaves (SPI slave 0 to SPI slave 3).

The RSPCK0 and MOSI0 outputs of this MCU (master) are connected to the SPCK and MOSI inputs of SPI slave 0 to SPI slave 3. The MISO outputs of SPI slave 0 to SPI slave 3 are all connected to the MISO0 input of this MCU (master). SSL00 to SSL03 outputs of this MCU (master) are connected to the SSL inputs of SPI slave 0 to SPI slave 3, respectively.

This MCU (master) drives RSPCK0, MOSI0, and SSL00 to SSL03. Of the SPI slave 0 to SPI slave 3, the slave that receives low-level input into the SSL input drives MISO.

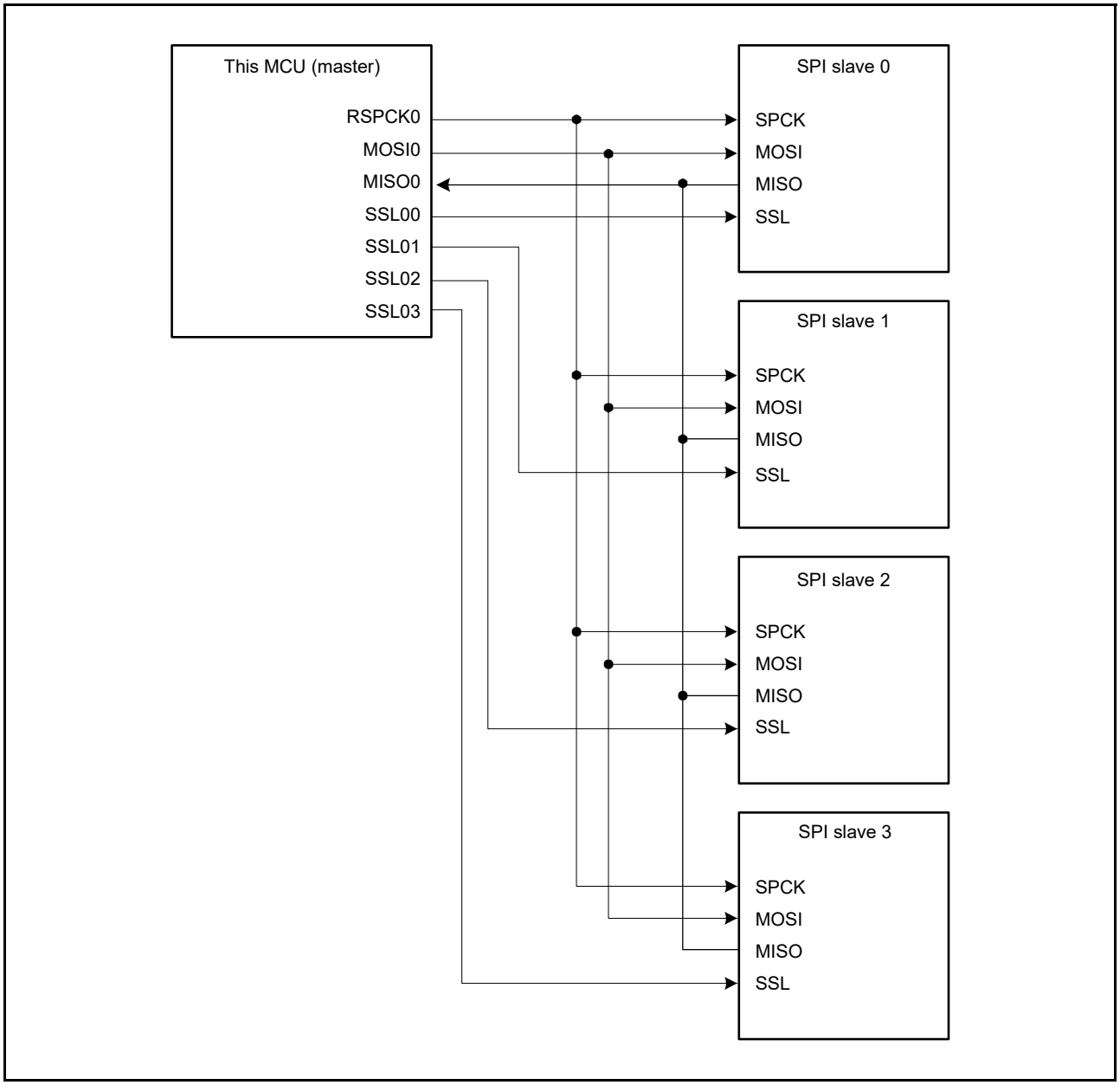


Figure 38.8 Single-Master/Multi-Slave Configuration Example (This MCU = Master)

38.3.3.4 Single Master/Multi-Slave (with This MCU Acting as Slave)

Figure 38.9 shows a single-master/multi-slave RSPI system configuration example when this MCU is used as a slave. In the example of Figure 38.9, the RSPI system is comprised of an SPI master and two MCUs (slave X and slave Y). The SPCK and MOSI outputs of the SPI master are connected to the RSPCK0 and MOSI0 inputs of the MCUs (slave X and slave Y). The MISO0 outputs of the MCUs (slave X and slave Y) are all connected to the MISO input of the SPI master. SSLX and SSLY outputs of the SPI master are connected to the SSL00 inputs of the MCUs (slave X and slave Y), respectively.
The SPI master always drives SPCK, MOSI, SSLX, and SSLY. Of the MCUs (slave X and slave Y), the slave that receives low-level input into the SSL00 input drives MISO0.

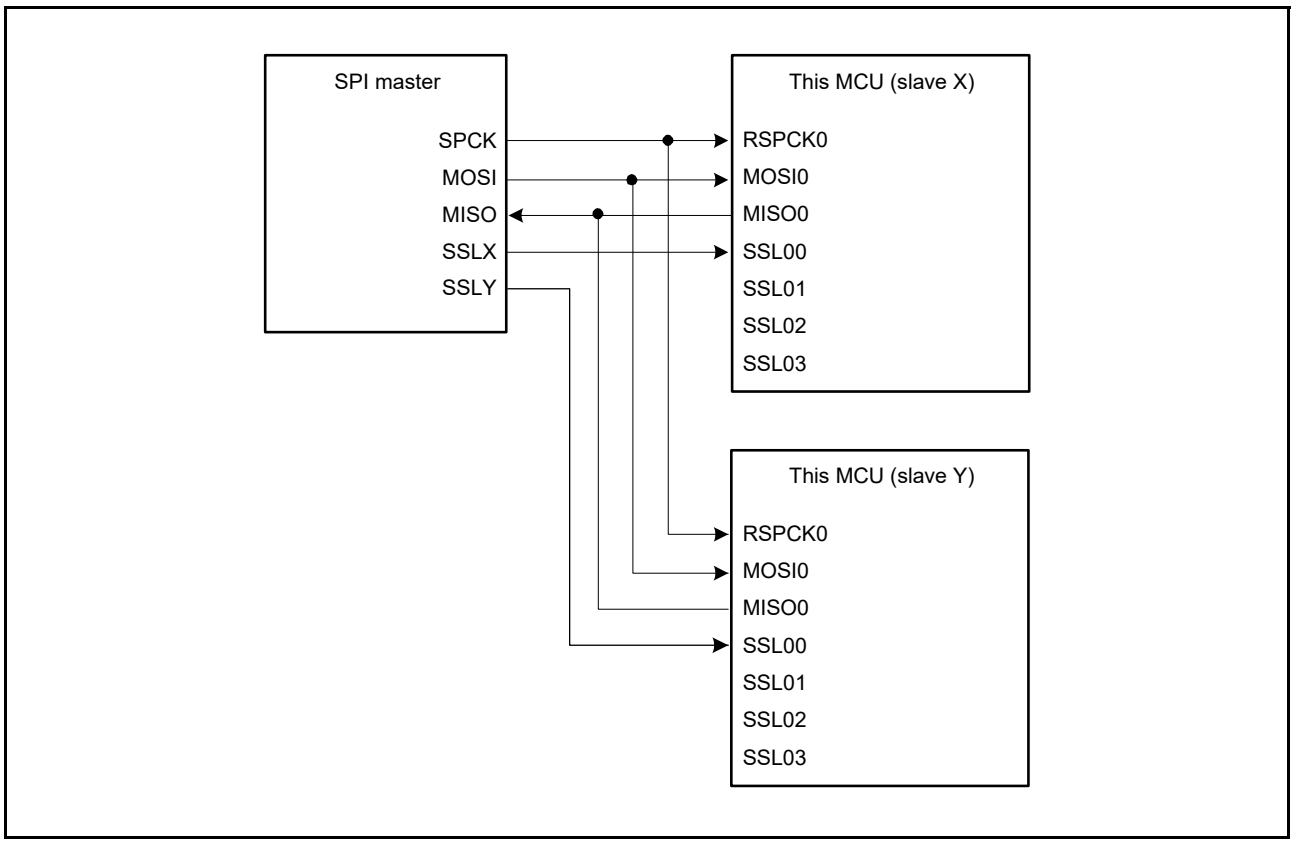


Figure 38.9 Single-Master/Multi-Slave Configuration Example (This MCU = Slave)

38.3.3.5 Multi-Master/Multi-Slave (with This MCU Acting as Master)

Figure 38.10 shows a multi-master/multi-slave RSPI system configuration example when this MCU is used as a master. In the example of Figure 38.10, the RSPI system is comprised of two MCUs (master X and master Y) and two SPI slaves (SPI slave 1 and SPI slave 2).

The RSPCK0 and MOSI0 outputs of the MCUs (master X and master Y) are connected to the SPCK and MOSI inputs of SPI slaves 1 and 2. The MISO outputs of SPI slaves 1 and 2 are connected to the MISO0 inputs of the MCUs (master X and master Y). Any generic port Y output from this MCU (master X) is connected to the SSL00 input of this MCU (master Y). Any generic port X output of this MCU (master Y) is connected to the SSL00 input of this MCU (master X). The SSL01 and SSL02 outputs of the MCUs (master X and master Y) are connected to the SSL inputs of the SPI slaves 1 and 2. In this configuration example, since the system can be comprised solely of SSL00 input, and SSL01 and SSL02 outputs for slave connections, the SSL03 output of this MCU is not required.

This MCU drives RSPCK0, MOSI0, SSL01, and SSL02 when the SSL00 input level is high. When the SSL00 input level is low, this MCU detects a mode fault error, sets RSPCK0, MOSI0, SSL01, and SSL02 to Hi-Z, and releases the RSPI bus right to the other master. Of the SPI slaves 1 and 2, the slave that receives low-level input into the SSL input drives MISO.

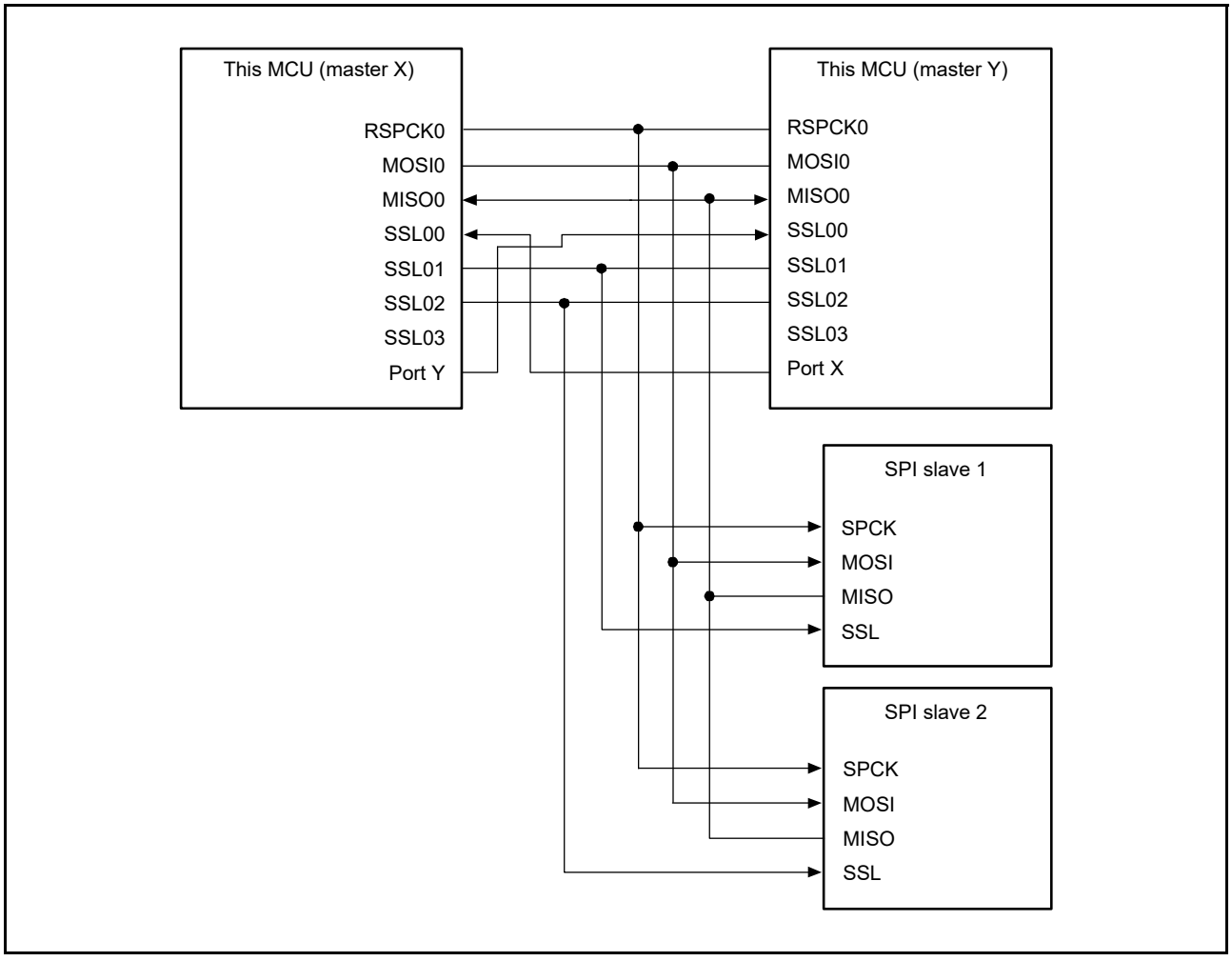


Figure 38.10 Multi-Master/Multi-Slave Configuration Example (This MCU = Master)

When setting TI SSP, enter the following levels for port X and port Y.

- Start of communication: the value of SSLP.SSL0P of the other master
- End of communication: the inverted value of SSLP.SSL0P of the other master

38.3.3.6 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (with This MCU Acting as Master)

Figure 38.11 shows a master (clock synchronous operation)/slave (clock synchronous operation) RSPI system configuration example when this MCU is used as a master. In the master (clock synchronous operation)/slave (clock synchronous operation) configuration, SSL00 to SSL03 of this MCU (master) are not used.
This MCU (master) drives the RSPCK0 and MOSI0. The SPI slave drives the MISO.

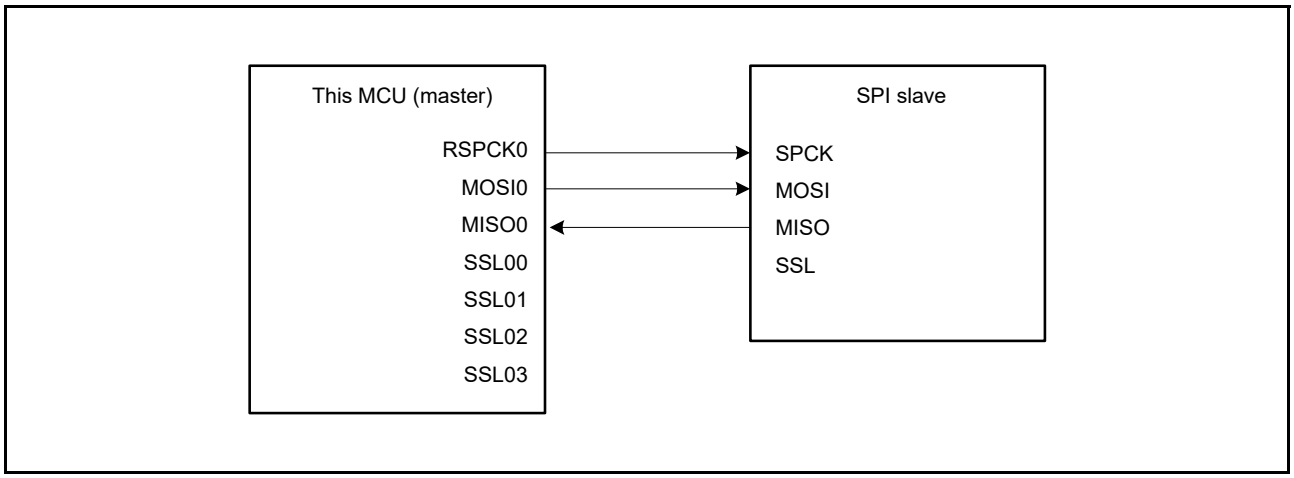


Figure 38.11 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) Configuration Example (This MCU = Master)

38.3.3.7 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (with This MCU Acting as Slave)

Figure 38.12 shows a master (clock synchronous operation)/slave (clock synchronous operation) RSPI system configuration example when this MCU is used as a slave. When this MCU is to operate as a slave (clock synchronous operation), this MCU (slave) drives the MISO0 and the SPI master drives the SPCK and MOSI. In addition, SSL00 to SSL03 of this MCU (slave) are not used.
Only in the single-slave configuration in which the SPCMDm.CPHA bit is set to 1, this MCU (slave) can execute serial transfer.

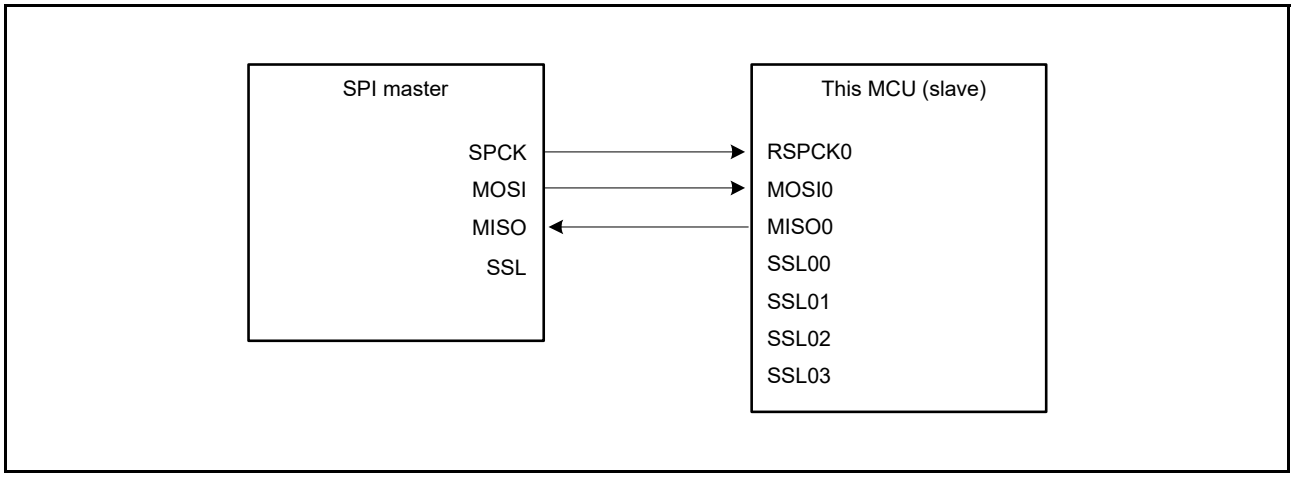


Figure 38.12 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) Configuration Example (This MCU = Slave, CPHA = 1)

38.3.4 Data Format

The RSPIA's data format depends on the settings in RSPI command register m (SPCMDm) (m = 0 to 7) and the parity enable bit in RSPI control register (SPCR.SPPE). Regardless of whether the MSB or LSB is first, the RSPIA treats the range from the LSB bit in the RSPI data register (SPDR) to the selected data length as transmit data.

38.3.4.1 Data Format of One Frame

The format of one frame of data before or after transfer is shown below.

(a) With Parity Disabled

When parity is disabled, transmission or reception of data proceeds with the length in bits selected in the RSPI data length setting bits in RSPI command register m (SPCMDm.SPB[4:0]).

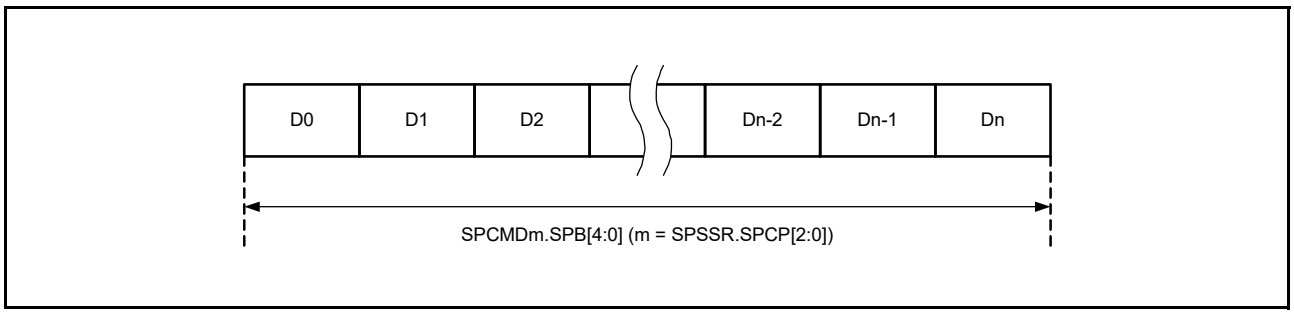


Figure 38.13 Outline of the Data Format (with Parity Disabled)

(b) With Parity Enabled

When parity is enabled, transmission or reception of data proceeds with the length in bits selected in the RSPI data length setting bits in RSPI command register m (SPCMDm.SPB[4:0]). In this case, however, the last bit is a parity bit.

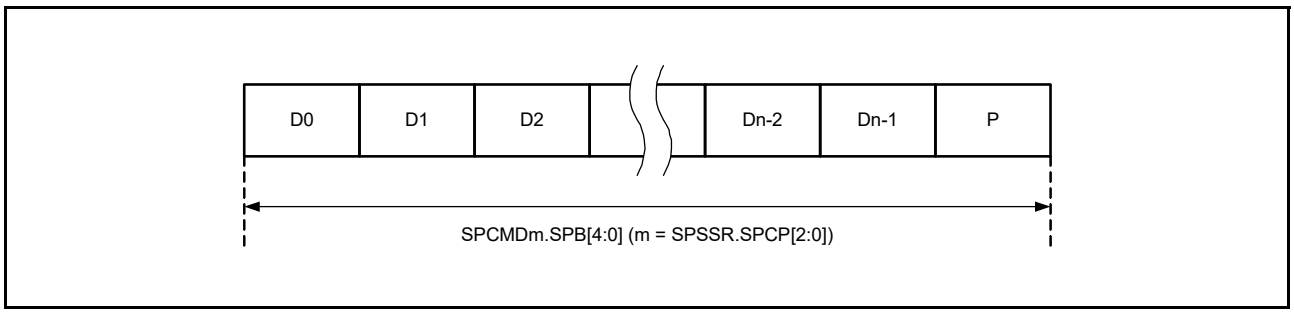


Figure 38.14 Outline of the Data Format (with Parity Enabled)

38.3.4.2 When Parity is Disabled (SPCR.SPPE = 0)

When parity is disabled, data for transmission are copied to the shift register with no prior processing. A description of the connection between the RSPI data register (SPDR) and the shift register in terms of the combination of MSB or LSB first and data length is given below.

(1) MSB First Transfer (32-Bit Data)

Figure 38.15 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity disabled, an RSPI data length of 32 bits, and MSB first selected.

In transmission, bits T31 to T00 from the current stage of the transmit buffer are copied to the shift register. Data for transmission are shifted out from the shift register in order from T31, through T30, and so on to T00.

In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R31 to R00 have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer.

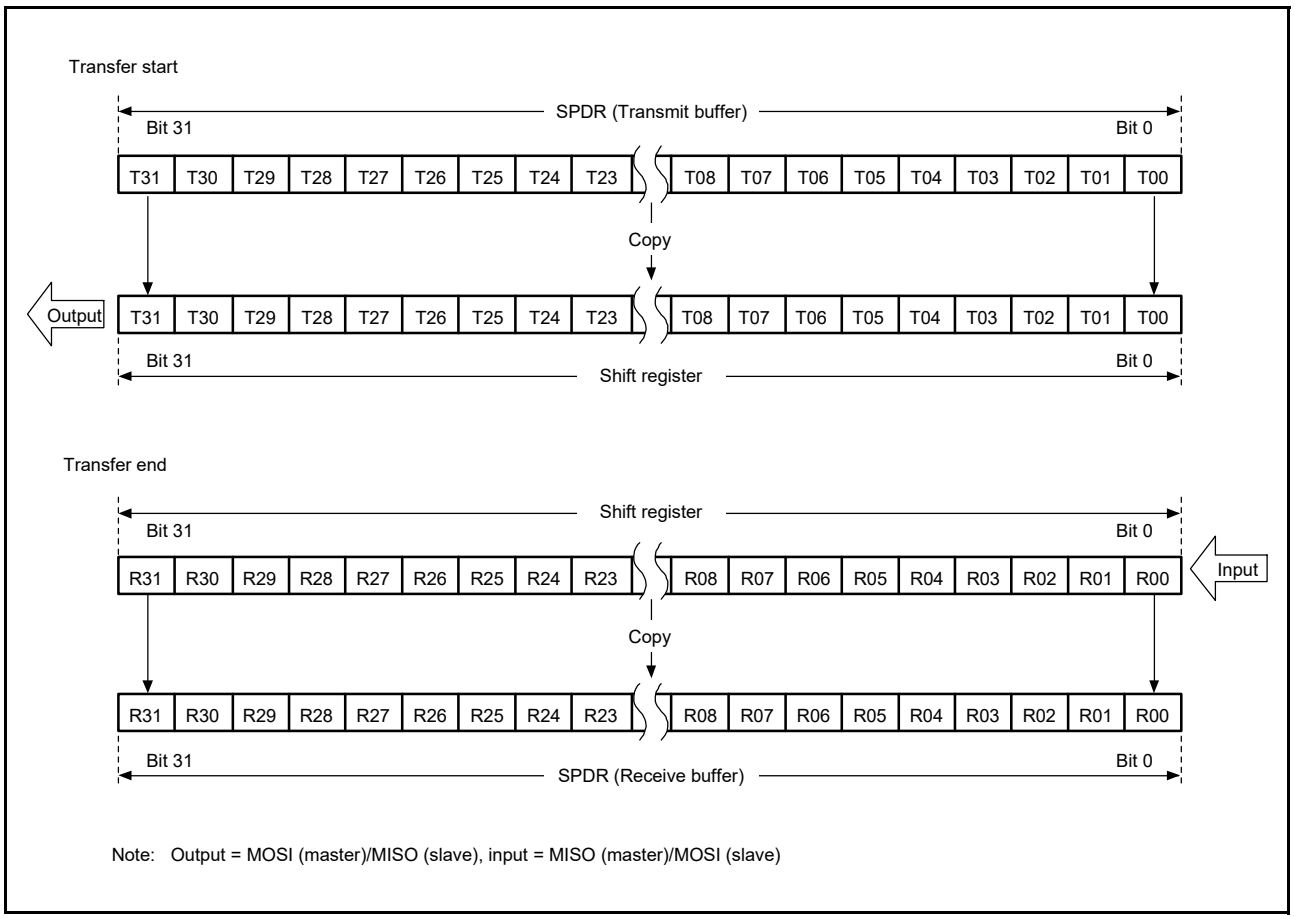


Figure 38.15 MSB First Transfer (32-Bit Data, Parity Disabled)

(2) MSB First Transfer (24-Bit Data)

Figure 38.16 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity disabled, 24 bits as the RSPI data length for an example that is not 32 bits, and MSB first selected.
In transmission, the lower-order 24 bits (T23 to T00) from the current stage of the transmit buffer are copied to the shift register. Data for transmission are shifted out from the shift register in order from T23, through T22, and so on to T00.
In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R23 to R00 have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer.

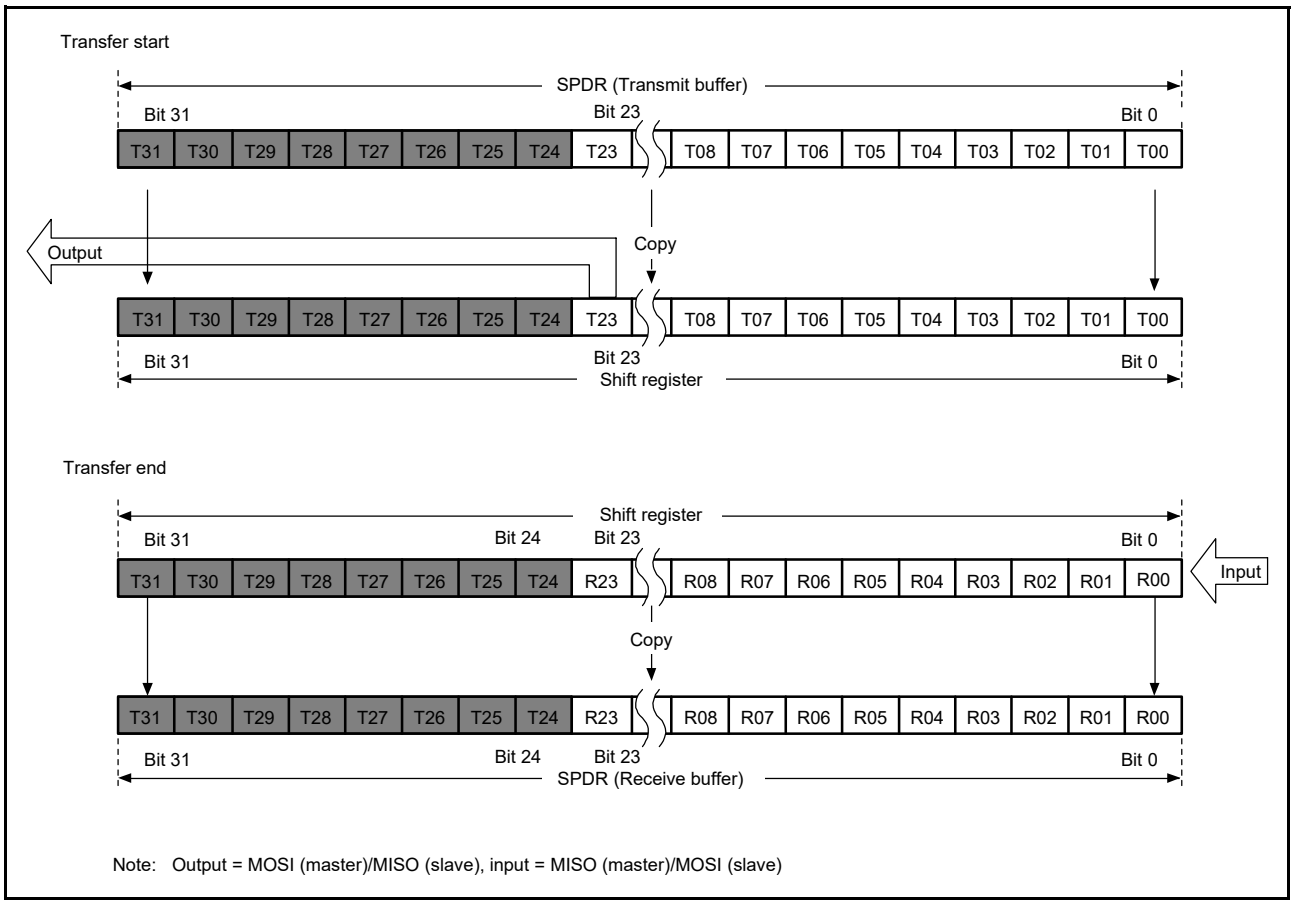


Figure 38.16 MSB First Transfer (24-Bit Data, Parity Disabled)

(3) LSB First Transfer (32-Bit Data)

Figure 38.17 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity disabled, an RSPI data length of 32 bits, and LSB first selected.

In transmission, bits T31 to T00 from the current stage of the transmit buffer are reordered bit by bit to obtain the order T00 to T31 for copying to the shift register. Data for transmission are shifted out from the shift register in order from T00, through T01, and so on to T31.

In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R00 to R31 have been collected after input of the required number of cycles of RSPCK, the values in the shift register are reordered bit by bit to obtain the order R31 to R00 for copying to the receive buffer.

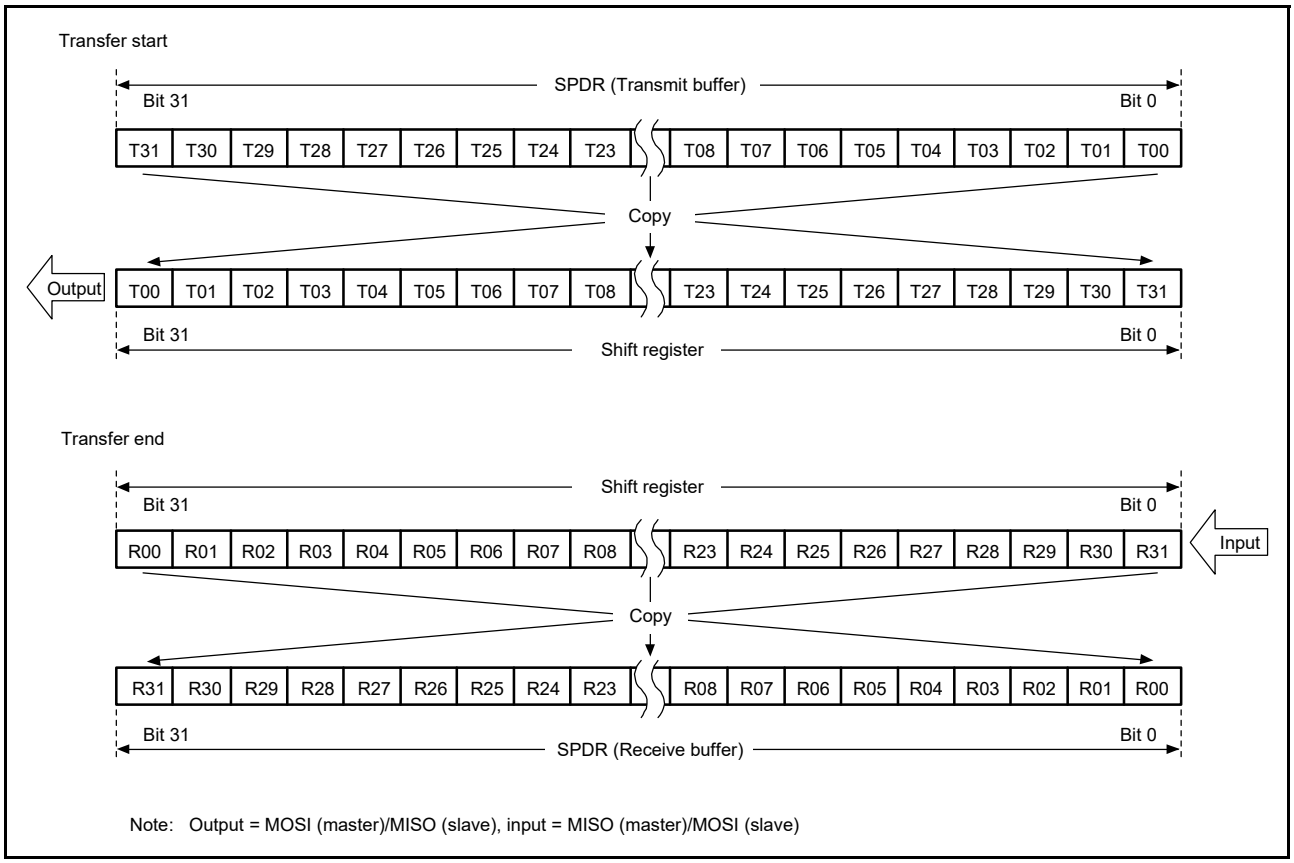


Figure 38.17 LSB First Transfer (32-Bit Data, Parity Disabled)

(4) LSB First Transfer (24-Bit Data)

Figure 38.18 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity disabled, 24 bits as the RSPI data length for an example that is not 32 bits, and LSB first selected.

In transmission, the lower-order 24 bits (T23 to T00) from the current stage of the transmit buffer are reordered bit by bit to obtain the order T00 to T23 for copying to the shift register. Data for transmission are shifted out from the shift register in order from T00, through T01, and so on to T23.

In reception, received data are shifted in bit by bit through bit 8 of the shift register. When bits R00 to R23 have been collected after input of the required number of cycles of RSPCK, the values in the shift register are reordered bit by bit to obtain the order R23 to R00 for copying to the receive buffer.

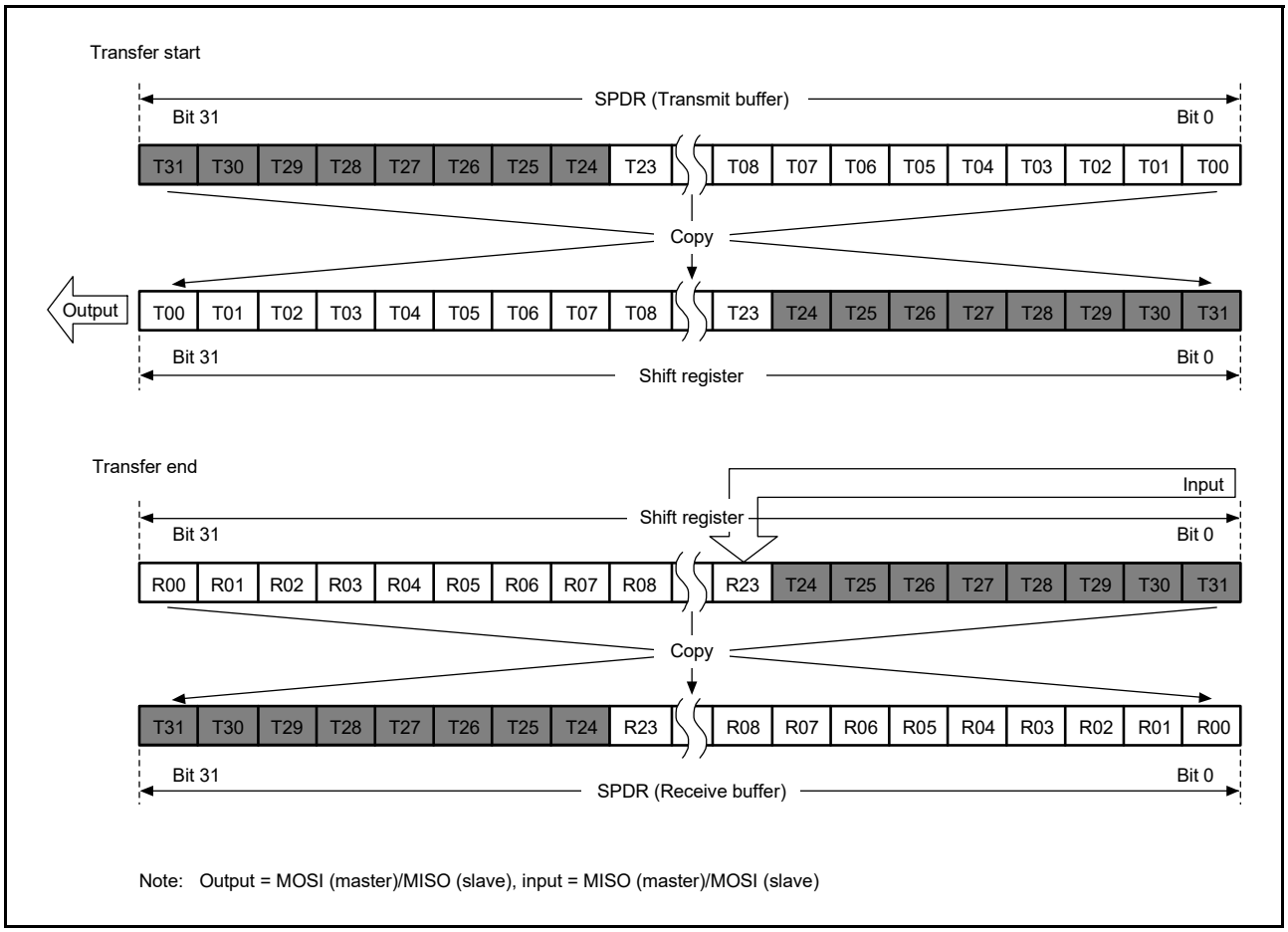


Figure 38.18 LSB First Transfer (24-Bit Data, Parity Disabled)

38.3.4.3 When Parity is Enabled (SPCR.SPPE = 1)

When parity is enabled, the lowest-order bit of the data for transmission becomes a parity bit. Hardware calculates the value of the parity bit.

(1) MSB First Transfer (32-Bit Data)

Figure 38.19 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity enabled, an RSPI data length of 32 bits, and MSB first selected.

In transmission, the value of the parity bit (P) is calculated from bits T31 to T01. This replaces the final bit, T00, and the whole is copied to the shift register. Data are transmitted in the order T31, T30, …, T01, and P.

In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R31 to P have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. On copying of data to the receive buffer, the data from R31 to P are checked to judge the parity error.

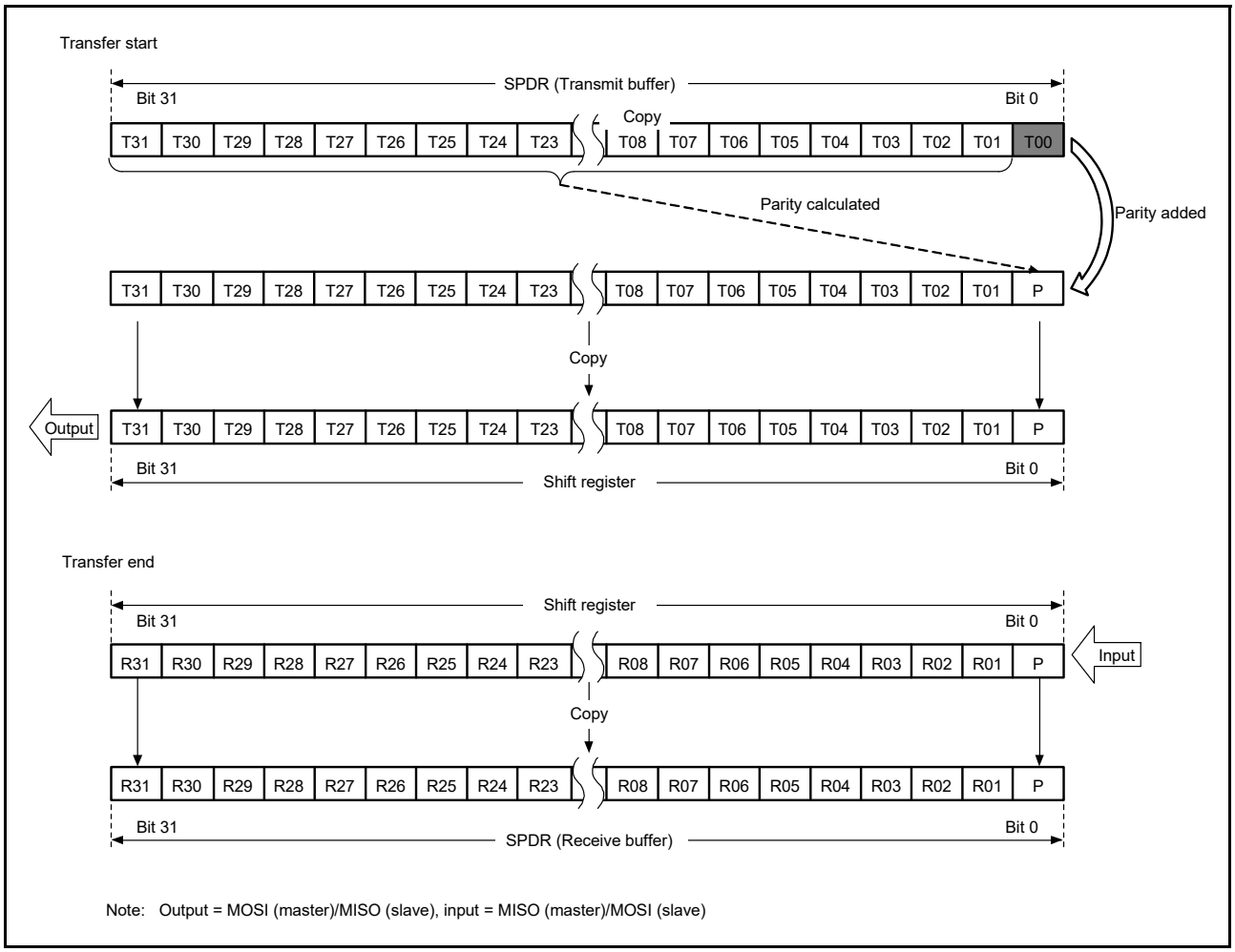


Figure 38.19 MSB First Transfer (32-Bit Data, Parity Enabled)

(2) MSB First Transfer (24-Bit Data)

Figure 38.20 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity enabled, 24 bits as the RSPI data length for an example that is not 32 bits, and MSB first selected.
In transmission, the value of the parity bit (P) is calculated from bits T23 to T01. This replaces the final bit, T00, and the whole is copied to the shift register. Data are transmitted in the order T23, T22, …, T01, and P.
In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R23 to P have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. On copying of data to the receive buffer, the data from R23 to P are checked to judge the parity error.

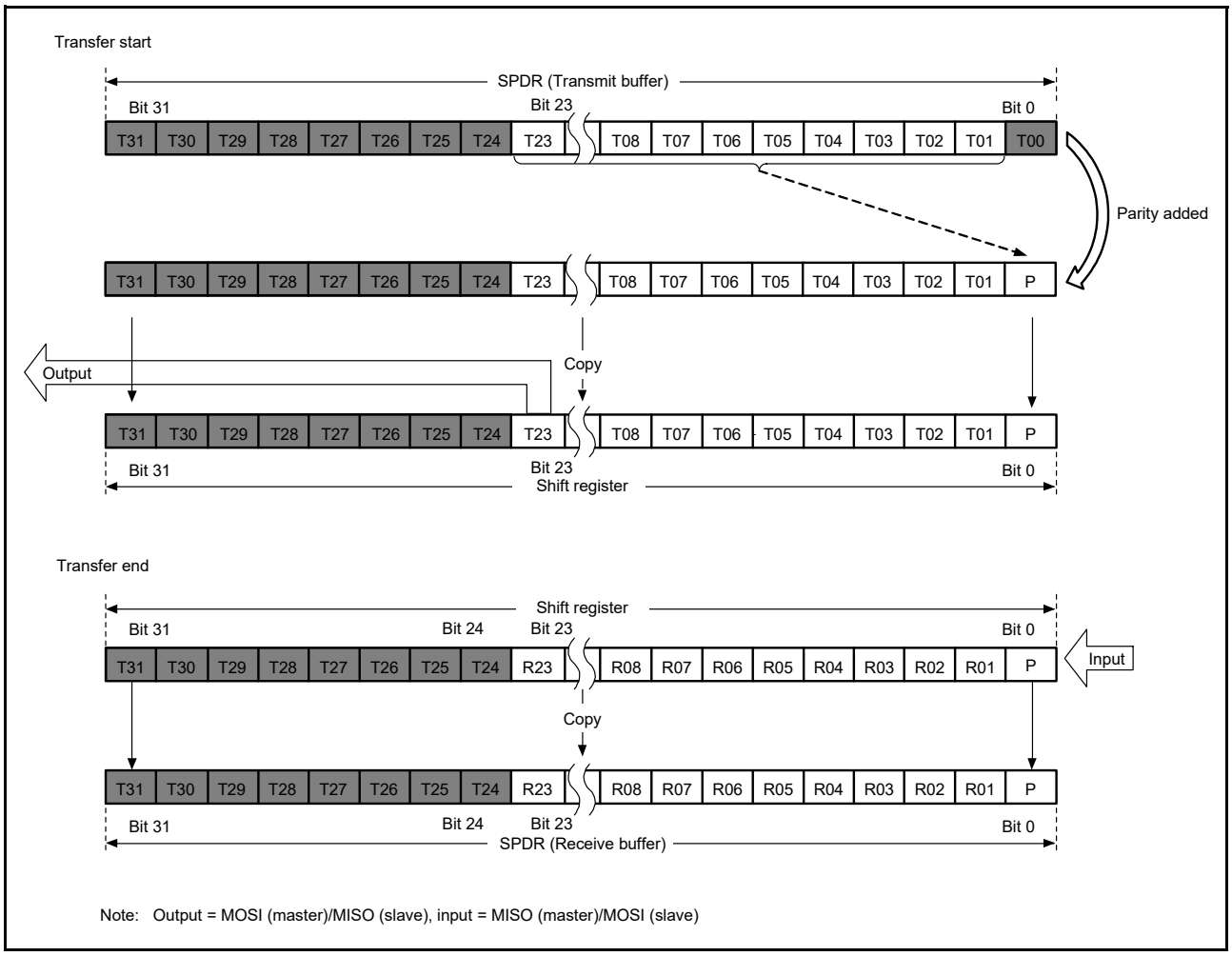


Figure 38.20 MSB First Transfer (24-Bit Data, Parity Enabled)

(3) LSB First Transfer (32-Bit Data)

Figure 38.21 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity enabled, an RSPI data length of 32 bits, and LSB first selected.

In transmission, the value of the parity bit (P) is calculated from bits T30 to T00. This replaces the final bit, T31, and the whole is copied to the shift register. Data are transmitted in the order T00, T01, …, T30, and P.

In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R00 to P have been collected after input of the required number of cycles of RSPCK, the values in the shift register are reordered bit by bit to obtain the order P to R00 for copying to the receive buffer. On copying of data to the receive buffer, the data from R00 to P are checked to judge the parity error.

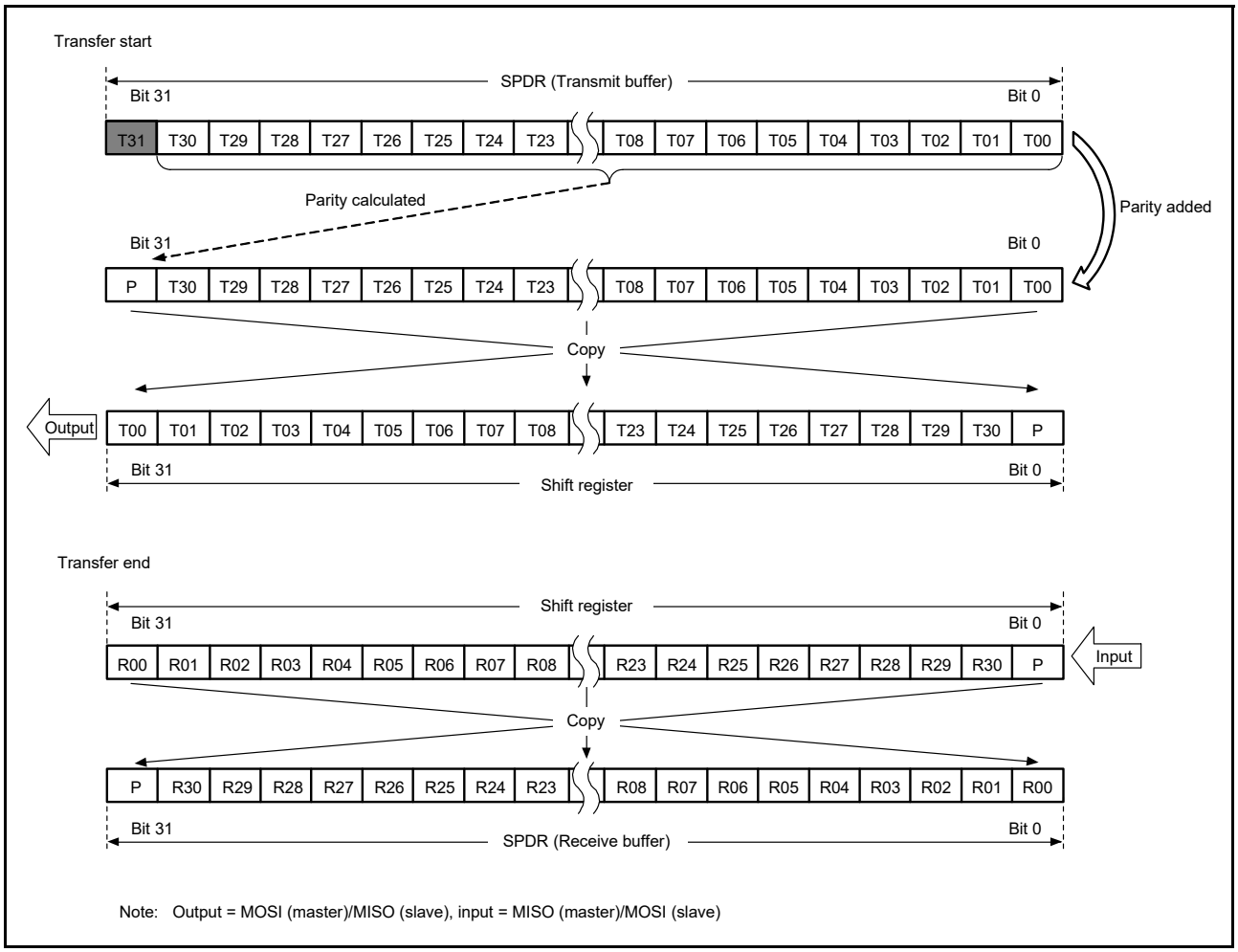


Figure 38.21 LSB First Transfer (32-Bit Data, Parity Enabled)

(4) LSB First Transfer (24-Bit Data)

Figure 38.22 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity enabled, 24 bits as the RSPI data length for an example that is not 32 bits, and LSB first selected.
In transmission, the value of the parity bit (P) is calculated from bits T22 to T00. This replaces the final bit, T23, and the whole is copied to the shift register. Data are transmitted in the order T00, T01, …, T22, and P.
In reception, received data are shifted in bit by bit through bit 8 of the shift register. When bits R00 to P have been collected after input of the required number of cycles of RSPCK, the values in the shift register are reordered bit by bit to obtain the order P (bit 23) to R00 (bit 0) for copying to the receive buffer. On copying of data to the receive buffer, the data from R00 to P are checked to judge the parity error.

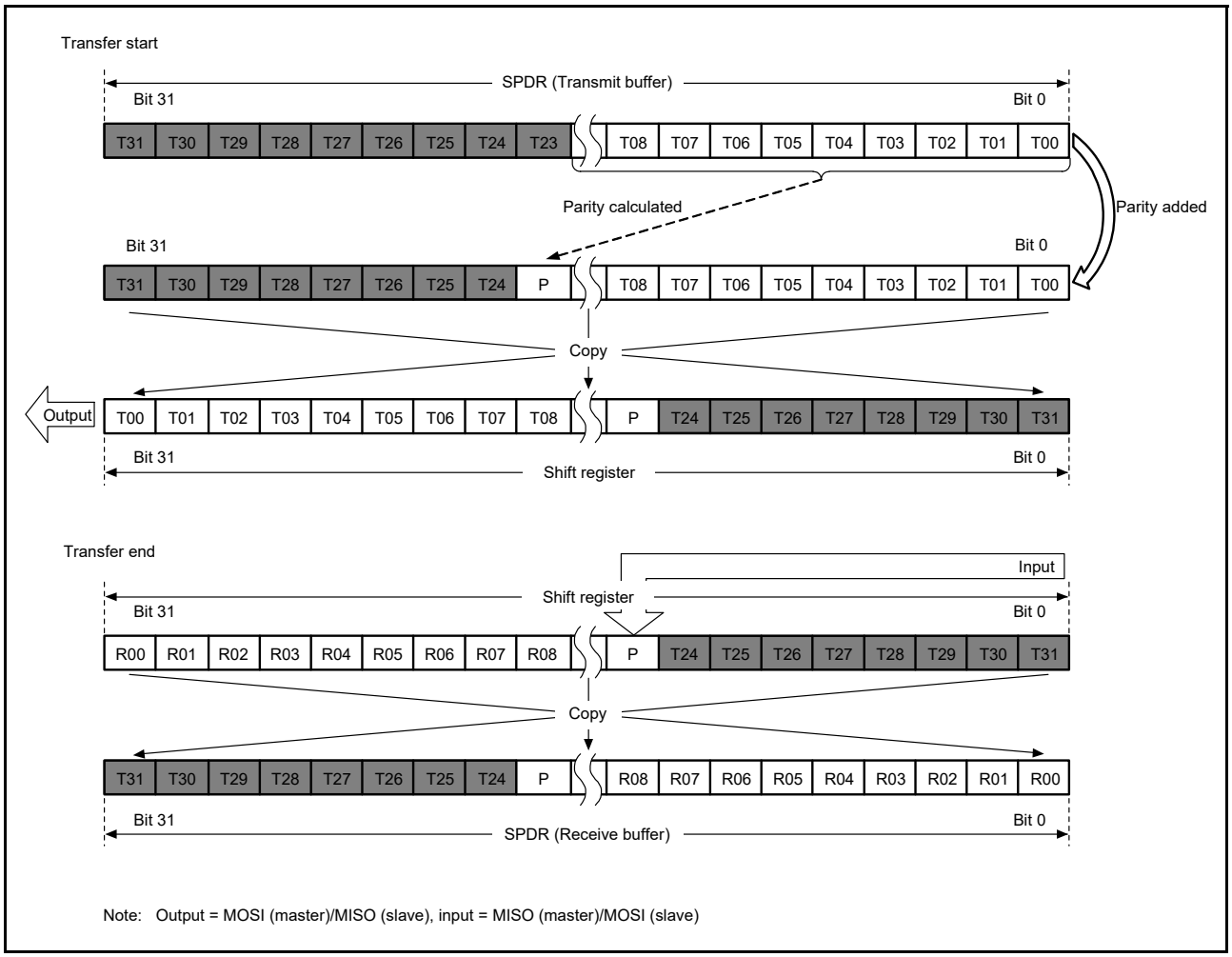


Figure 38.22 LSB First Transfer (24-Bit Data, Parity Enabled)

38.3.4.4 Byte Swap Transmission

When the SPDCR.BYSW bit is 1 (byte swapping of SPDR data enabled), data bytes written in the transmit buffer (SPDR) will be swapped (in 8 bit units) when the data is transferred to the shift register. Figure 38.23 shows data transfer between the SPDR register and the shift register when data length is 32 bits in combination of MSB/LSB first with byte swapping enabled/disabled.

(1) MSB first transfer (when the byte swap is disabled)
Data (Byte 3 [T31 to T24] to Byte 0 [T07 to T00]) in the transmit buffer are copied to the shift register. Bit values in the shift register are shifted and transmitted in the order of T31, T30, …, and T00 as transmit data.

(2) MSB first transfer (when the byte swap is enabled)
Byte values of the transmit buffer (Byte 3 [T31 to T24] to Byte 0 [T07 to T00]) are reversed in byte units and are copied to the shift register in the order of Byte 0 [T07 to T00] to Byte 3 [T31 to T24].
Bit values in the shift register are shifted and transmitted in the order of T07, T06, …, T00, T15, T14, …, T08, T23, T22, …, T16, T31, T30, …, and T24 as transmit data.

(3) LSB first transfer (when the byte swap is disabled)
Bit values of the transmit buffer (Byte 3 [T31 to T24] to Byte 0 [T07 to T00]) are reversed in bit units and are copied to the shift register in the order of Byte 0 [T00 to T07] to Byte 3 [T24 to T31].
Bit values in the shift register are shifted and transmitted in the order of T00, T01, …, and T31 as transmit data.

(4) LSB first transfer (when the byte swap is enabled)
Bit values of each byte of the transmit buffer (Byte 3 [T31 to T24] to Byte 0 [T07 to T00]) are reversed in bit units and are copied to the shift register in the order of Byte 3 [T24 to T31] to Byte 0 [T00 to T07].
Bit values in the shift register are shifted and transmitted in the order of T24, T25, …, T31, T16, T17, …, T23, T08, T09, …, T15, T00, T01, …, and T07 as transmit data.

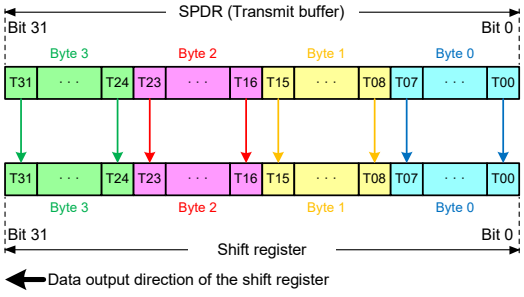
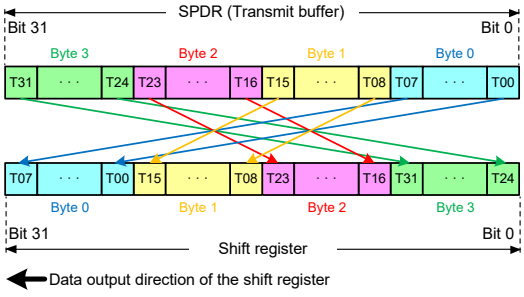
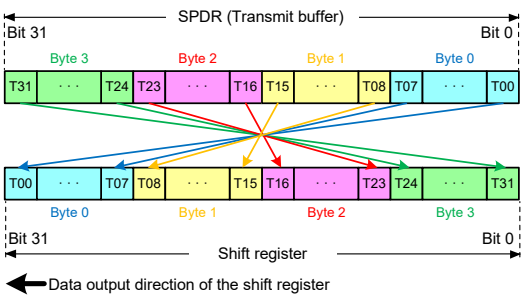
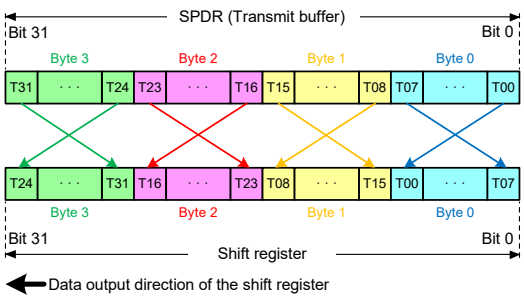
32-bit Byte Swap Specifications	Byte Swap is Disabled (BYSW = 0)	Byte Swap is Enabled (BYSW = 1)
MSB-first (LSBF = 0)		
LSB-first (LSBF = 1)		

Figure 38.23 Transmit Data Handling in MSB/LSB First with Byte Swapping Enabled/Disabled (Data Length = 32 bits)

Figure 38.24 shows data transfer between the SPDR register and the shift register when data length is 16 bits in combination of MSB/LSB first with byte swapping enabled/disabled.

(1) MSB first transfer (when the byte swap is disabled)
Data (Byte 1 [T15 to T08] to Byte 0 [T07 to T00]) in the transmit buffer are copied to the shift register in the order of Byte 1 [T15 to T08] to Byte 0 [T07 to T00]. Bit values in the shift register are shifted and transmitted in the order of T15, T14, …, and T00 as transmit data.

(2) MSB first transfer (when the byte swap is enabled)
Byte values of the transmit buffer (Byte 1 [T15 to T08] to Byte 0 [T07 to T00]) are reversed in byte units and are copied to the shift register in the order of Byte 0 [T07 to T00] to Byte 1 [T15 to T08]. Bit values in the shift register are shifted and transmitted in the order of T07, T06, …, T00, T15, T14, …, T08 as transmit data.

(3) LSB first transfer (when the byte swap is disabled)
Bit values of the transmit buffer (Byte 1 [T15 to T08] to Byte 0 [T07 to T00]) are reversed in bit units and are copied to the shift register in the order of Byte 0 [T00 to T07] to Byte 1 [T15 to T08]. Bit values in the shift register are shifted and transmitted in the order of T00, T01, …, and T15 as transmit data.

(4) LSB first transfer (when the byte swap is enabled)
Bit values of each byte of the transmit buffer (Byte 1 [T15 to T08] to Byte 0 [T07 to T00]) are reversed in bit units and are copied to the shift register in the order of Byte 1 [T08 to T15] to Byte 0 [T00 to T07].
Bit values in the shift register are shifted and transmitted in the order of T08, T09, …, T15, T00, T01, …, and T07 as transmit data.

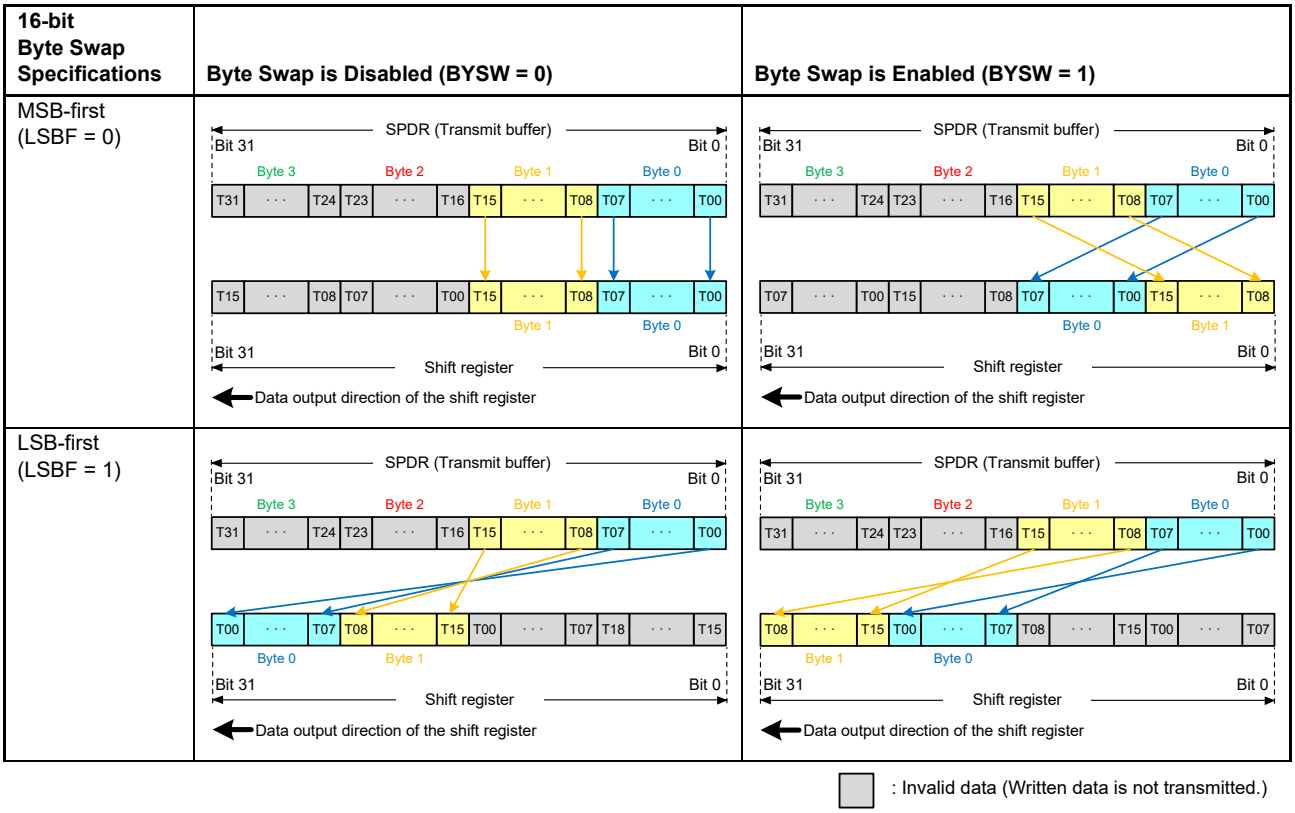


Figure 38.24 Transmit Data Handling in MSB/LSB First with Byte Swapping Enabled/Disabled (Data Length = 16 bits)

Note 1. When using the byte swap, set 16 bits or 32 bits to the data length (SPCMDm.SPB[4:0] setting). If setting the other length (4 to 15 or 17 to 31 bits), the operation is not guaranteed.

Note 2. When the byte swap is enabled, disable the parity function (SPCR.SPPE = 0). If the parity function is enabled (SPCR.SPPE = 1), the operation after the change is not guaranteed.
Note 3. Set the SPDCR.BYSW bit, while the SPCR.SPE bit is 0. If the value of the SPDCR.BYSW bit is changed while the SPCR.SPE bit is 1, the operation after the change is not guaranteed.

38.3.4.5 Byte Swap Reception

When the SPDCR.BYSW bit is 1 (byte swapping of SPDR data enabled), data bytes in the shift register will be swapped (in 8 bit units) when the data is transferred to the receive buffer (SPDR). Figure 38.25 shows data transfer between the shift register and the SPDR register when data length is 32 bits in combination of MSB/LSB first with byte swapping enabled/disabled.

(1) MSB first transfer (when the byte swap is disabled)
The first received data (R31) is stored in bit 0 of the shift register, and received data is shifted in the order of R31, R30, …, and R00. When necessary RSPCK cycles are input and data is stored from Byte 3 [R31 to R24] to Byte 0 [R07 to R00], the shift register value is copied to the receive buffer.

(2) MSB first transfer (when the byte swap is enabled)
The first received data (R07) is stored in bit 0 of the shift register, and received data is shifted in the order of R07, R06, …, R00, R15, R14, …, R08, R23, R22, …, R16, R31, R30, …, and R24. When necessary RSPCK cycles are input and data is stored from Byte 0 [R07 to R00] to Byte 3 [R31 to R24], byte values in the shift register are reversed in byte units and are copied to the receive buffer in the order of Byte 3 [R31 to R24] to Byte 0 [R07 to R00].

(3) LSB first transfer (when the byte swap is disabled)
The first received data (R00) is stored in bit 0 of the shift register, and received data is shifted in the order of R00, R01, …, and R31. When necessary RSPCK cycles are input and data is stored from Byte 0 [R00 to R07] to Byte 3 [R24 to R31], bit values in the shift register are reversed in bit units and are copied to the receive buffer in the order of Byte 3 [R31 to R24] to Byte 0 [R07 to R00].

(4) LSB first transfer (when the byte swap is enabled)
The first received data (R24) is stored in bit 0 of the shift register, and received data is shifted in the order of R24, R25, …, R31, R16, R17, …, R23, R08, R09, …, R15, R00, R01, …, and R07. When necessary RSPCK cycles are input and data is stored from Byte 3 [R24 to R31] to Byte 0 [R00 to R07], bit values of each byte in the shift register are reversed in bit units and are copied to the receive buffer in the order of Byte 3 [R31 to R24] to Byte 0 [R07 to R00].

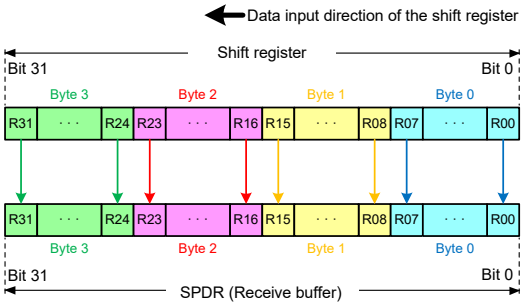
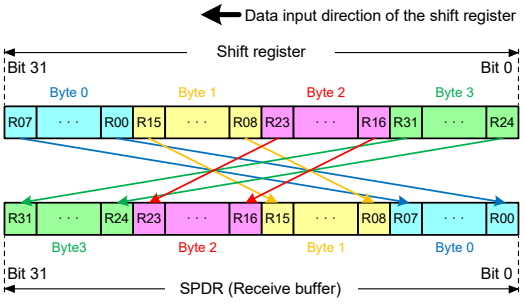
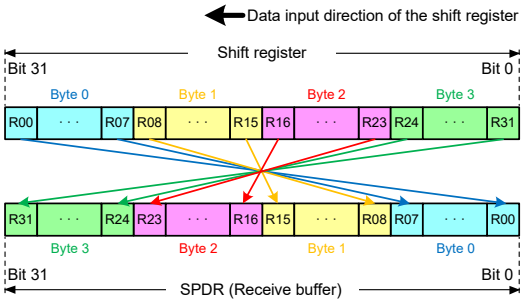
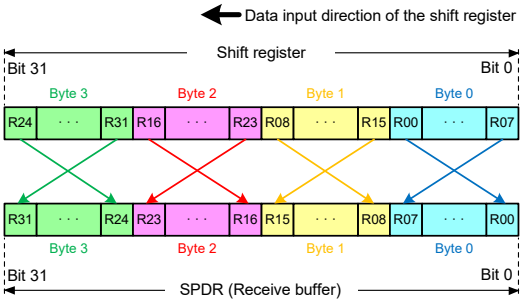
32-bit Byte Swap Specifications	Byte Swap is Disabled (BYSW = 0)	Byte Swap is Enabled (BYSW = 1)
MSB-first (LSBF = 0)		
LSB-first (LSBF = 1)		

Figure 38.25 Receive Data Handling in MSB/LSB First with Byte Swapping Enabled/Disabled (Data Length = 32 bits)

Figure 38.26 shows data transfer between the shift register and the SPDR register when data length is 16 bits in combination of MSB/LSB first with byte swapping enabled/disabled.

(1) MSB first transfer (when the byte swap is disabled)
The first received data (R15) is stored in bit 0 of the shift register, and received data is shifted in the order of R15, R14, …, and R00. When necessary RSPCK cycles are input and data is stored from Byte 3 [R31 to R24] to Byte 0 [R07 to R00], the shift register value is copied to the receive buffer.

(2) MSB first transfer (when the byte swap is enabled)
The first received data (R07) is stored in bit 0 of the shift register, and received data is shifted in the order of R07, R06, …, R00, R15, R14, …, and R08. When necessary RSPCK cycles are input and data is stored from Byte 0 [R07 to R00] to Byte 1 [R15 to R08], byte values in the shift register are reversed in byte units and are copied to the receive buffer in the order of Byte 3 [R31 to R24] to Byte 0 [R07 to R00].

(3) LSB first transfer (when the byte swap is disabled)
The first received data (R00) is stored in bit 15 of the shift register, and received data is shifted in the order of R00, R01, …, R07, R08, R09, …, and R15. When necessary RSPCK cycles are input and data is stored from Byte 0 [R00 to R07] to Byte 1 [R08 to R15], bit values in the shift register are reversed in bit units and are copied to the receive buffer in the order of Byte 3 [R31 to R24] to Byte 0 [R07 to R00].

(4) LSB first transfer (when the byte swap is enabled)
The first received data (R08) is stored in bit 15 of the shift register, and received data is shifted in the order of R08, R09, …, R15, R00, R01, …, and R15. When necessary RSPCK cycles are input and data is stored from Byte 1 [R08 to R15] to Byte 0 [R00 to R07], bit values of each byte in the shift register are reversed in bit units and are copied to the receive buffer in the order of Byte 3 [R31 to R24] to Byte 0 [R07 to R00].

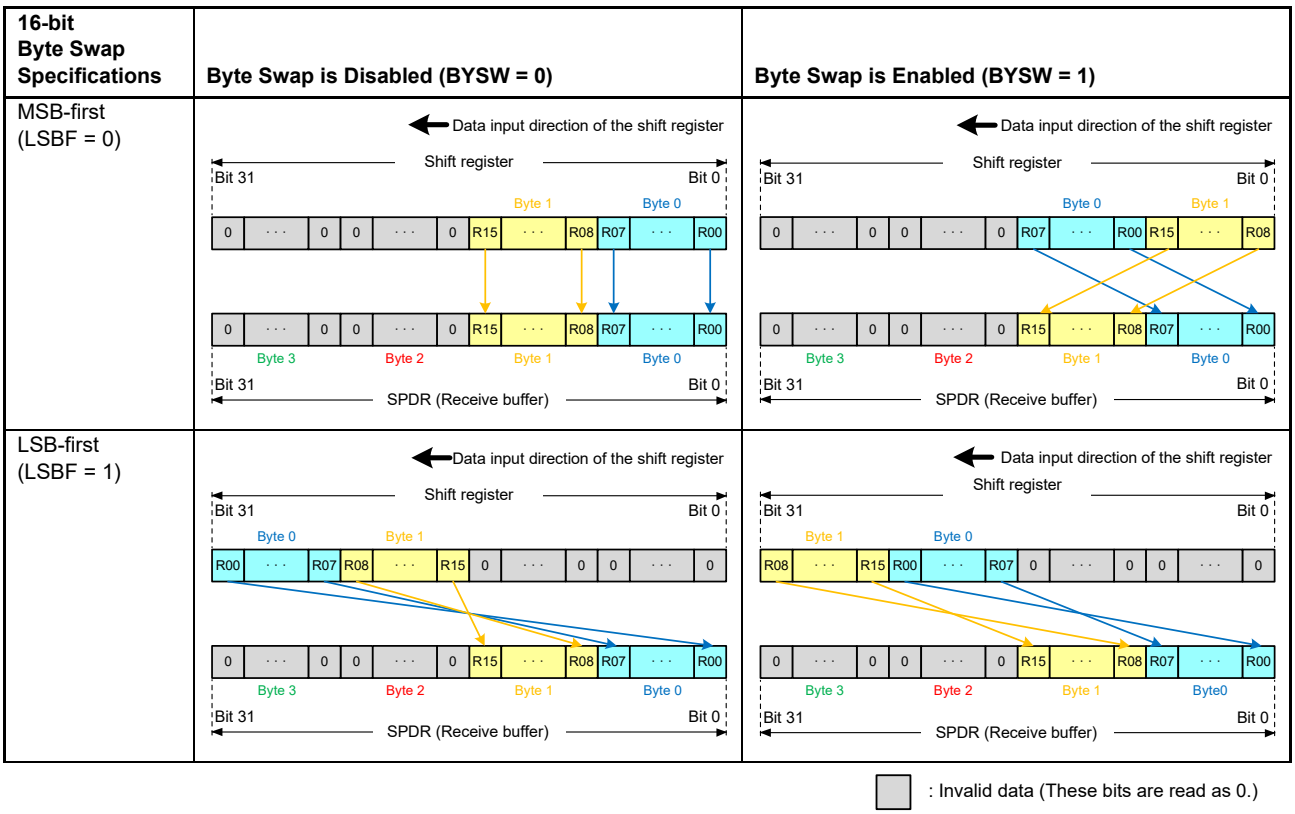


Figure 38.26 Receive Data Handling in MSB/LSB First with Byte Swapping Enabled/Disabled (Data Length = 16 bits)

Note 1. When using the byte swap, set 16 bits or 32 bits to the data length (SPCMDm.SPB[4:0] setting). If setting the other length (4 to 15 or 17 to 31 bits), the operation is not guaranteed.
Note 2. When the byte swap is enabled, disable the parity function (SPCR.SPPE = 0). If the parity function is enabled (SPCR.SPPE = 1), the operation after the change is not guaranteed.
Note 3. Set the SPDCR.BYSW bit, while the SPCR.SPE bit is 0. If the value of the SPDCR.BYSW bit is changed while the SPCR.SPE bit is 1, the operation after the change is not guaranteed.

38.3.5 Transfer Format (Frame Format)

38.3.5.1 CPHA = 0

Figure 38.27 shows a sample transfer format for the serial transfer of 8-bit data when the SPCMDm.CPHA bit is 0. Note that clock synchronous operation (the SPCR.SPMS bit is 1) should not performed when the RSPIA operates in slave mode (SPCR.MSTR = 0) and the CPHA bit is 0. In Figure 38.27, RSPCK0 (CPOL = 0) indicates the RSPCK0 signal waveform when the SPCMDm.CPOL bit is 0; RSPCK0 (CPOL = 1) indicates the RSPCK0 signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the RSPIA fetches serial transfer data into the shift register. The I/O directions of the signals depend on the RSPIA settings. For details, refer to section 38.3.2, Controlling RSPI Pins.

When the SPCMDm.CPHA bit is 0, the driving of valid data to the MOSI0 and MISO0 signals commences at an SSL0n signal assertion timing. The first RSPCK0 signal change timing that occurs after the SSL0n signal assertion becomes the first transfer data fetch timing. After this timing, data is sampled at every 1 RSPCK cycle. The change timing for MOSI0 and MISO0 signals is 1/2 RSPCK cycles after the transfer data fetch timing. The CPOL bit setting does not affect the RSPCK0 signal operation timing; it only affects the signal polarity.

t1 denotes a period from an SSL0n signal assertion to RSPCK0 oscillation (RSPCK delay). t2 denotes a period from the termination of RSPCK0 oscillation to an SSL0n signal negation (SSL negation delay). t3 denotes a period in which SSL0n 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 RSPIA of this MCU is in master mode, refer to section 38.3.13.1, Master Mode Operation.

[Motorola SPI]

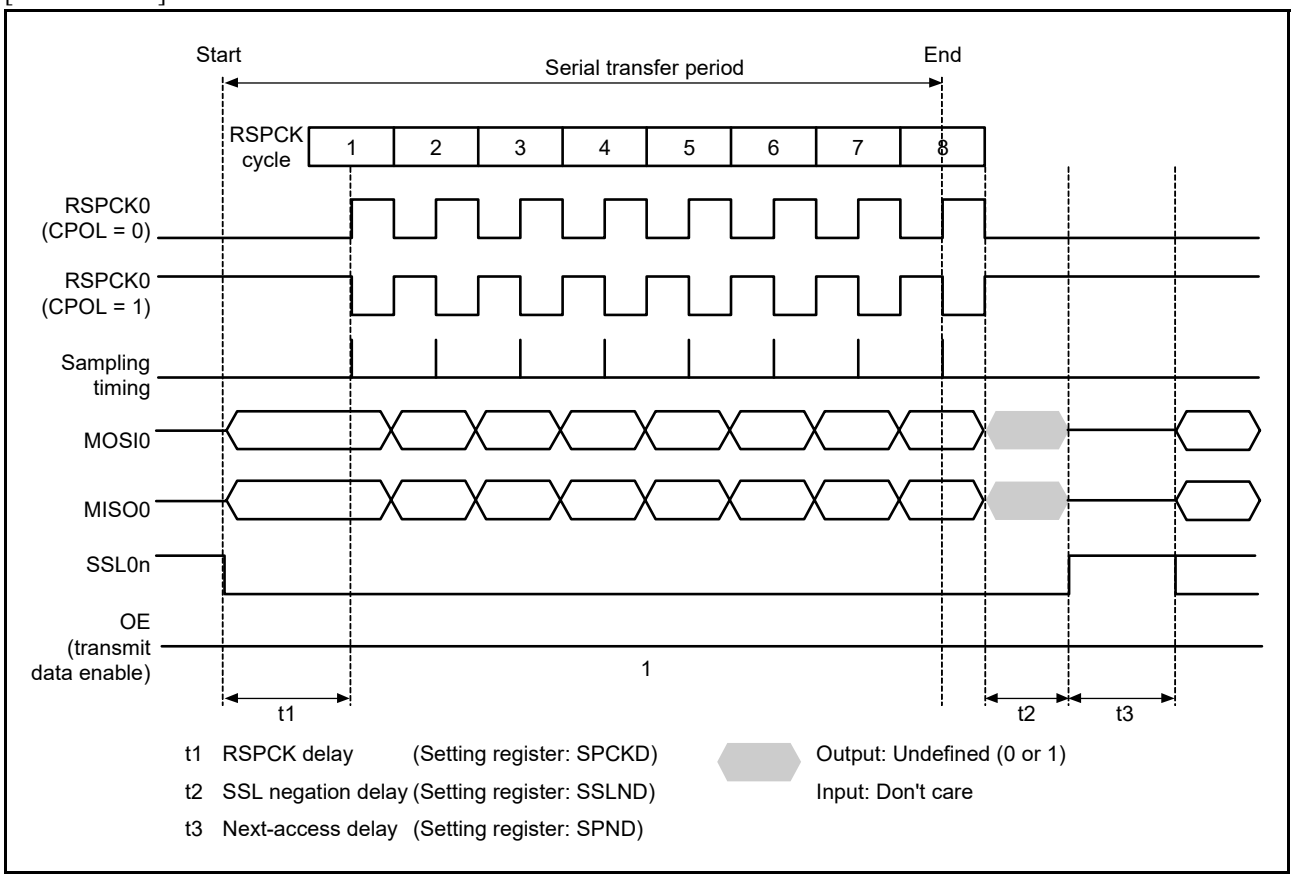


Figure 38.27 RSPI Transfer Format (CPHA = 0, FRFS = 0)

[TI SSP]

In TI SSP mode, setting the CPHA bit to 0 is not effective.

38.3.5.2 CPHA = 1

Figure 38.28 and Figure 38.29 show sample transfer formats for the serial transfer of 8-bit data when the SPCMDm.CPHA bit is 1. However, when the SPCR.SPMS bit is 1, the SSL0n signals are not used, and only the three signals RSPCK0, MOSI0, and MISO0 handle communications. In Figure 38.28, RSPCK0 (CPOL = 0) indicates the RSPCK0 signal waveform when the SPCMDm.CPOL bit is 0; RSPCK0 (CPOL = 1) indicates the RSPCK0 signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the RSPIA fetches serial transfer data into the shift register. The I/O directions of the signals depend on the RSPI mode (master or slave). For details, refer to section 38.3.2, Controlling RSPI Pins.

When the SPCMDm.CPHA bit is 1, the driving of invalid data to the MISO0 signal commences at an SSL0n signal assertion timing. The output of valid data to the MOSI0 and MISO0 signals commences at the first RSPCK0 signal change timing that occurs after the SSL0n signal assertion. After this timing, data is updated at every 1 RSPCK cycle. The transfer data fetch timing is 1/2 RSPCK cycles after the data update timing. The SPCMDm.CPOL bit setting does not affect the RSPCK0 signal operation timing; it only affects 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 RSPIA of this MCU is in master mode, refer to section 38.3.13.1, Master Mode Operation.

[Motorola SPI]

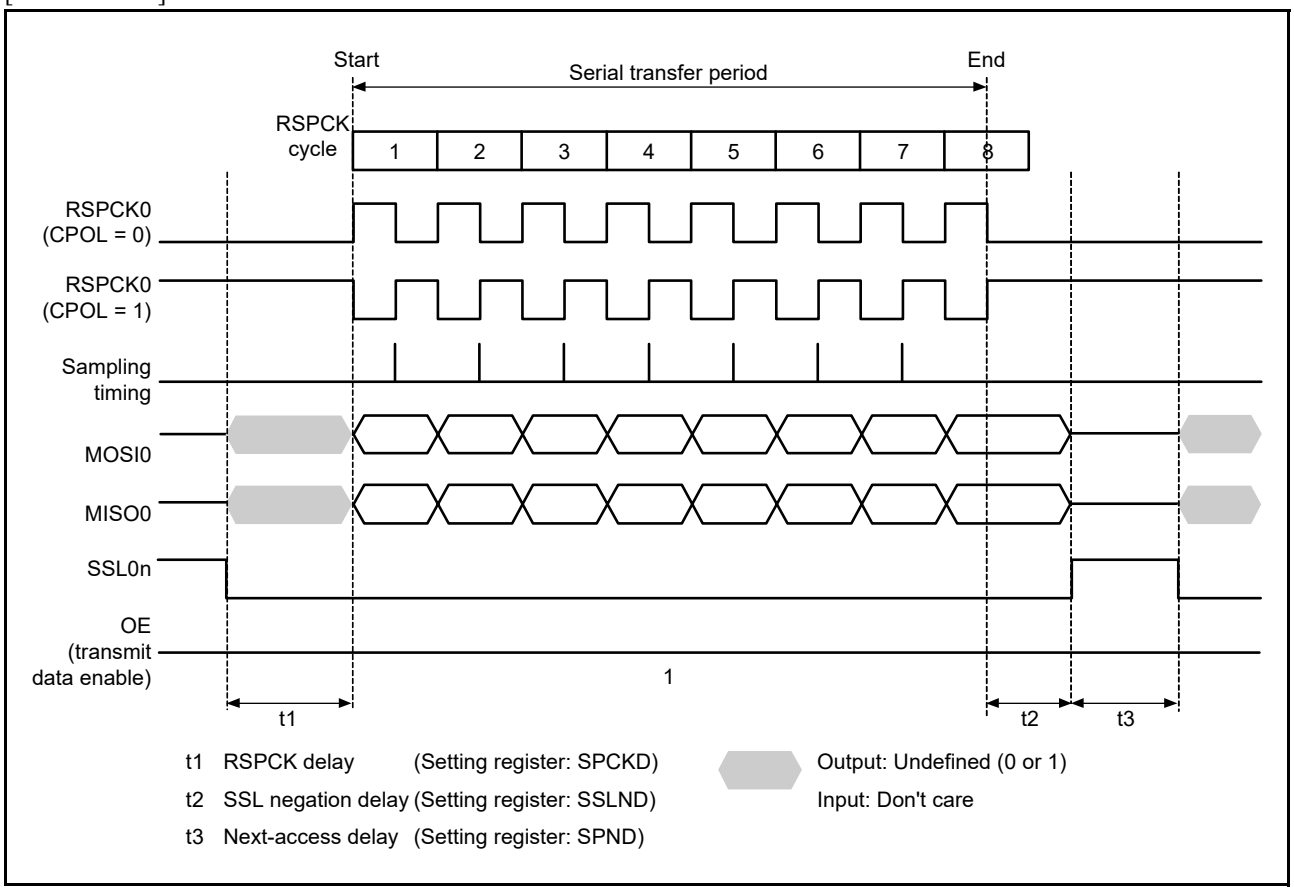


Figure 38.28 RSPI Transfer Format (CPHA = 1, FRFS = 0)

[TI SSP]

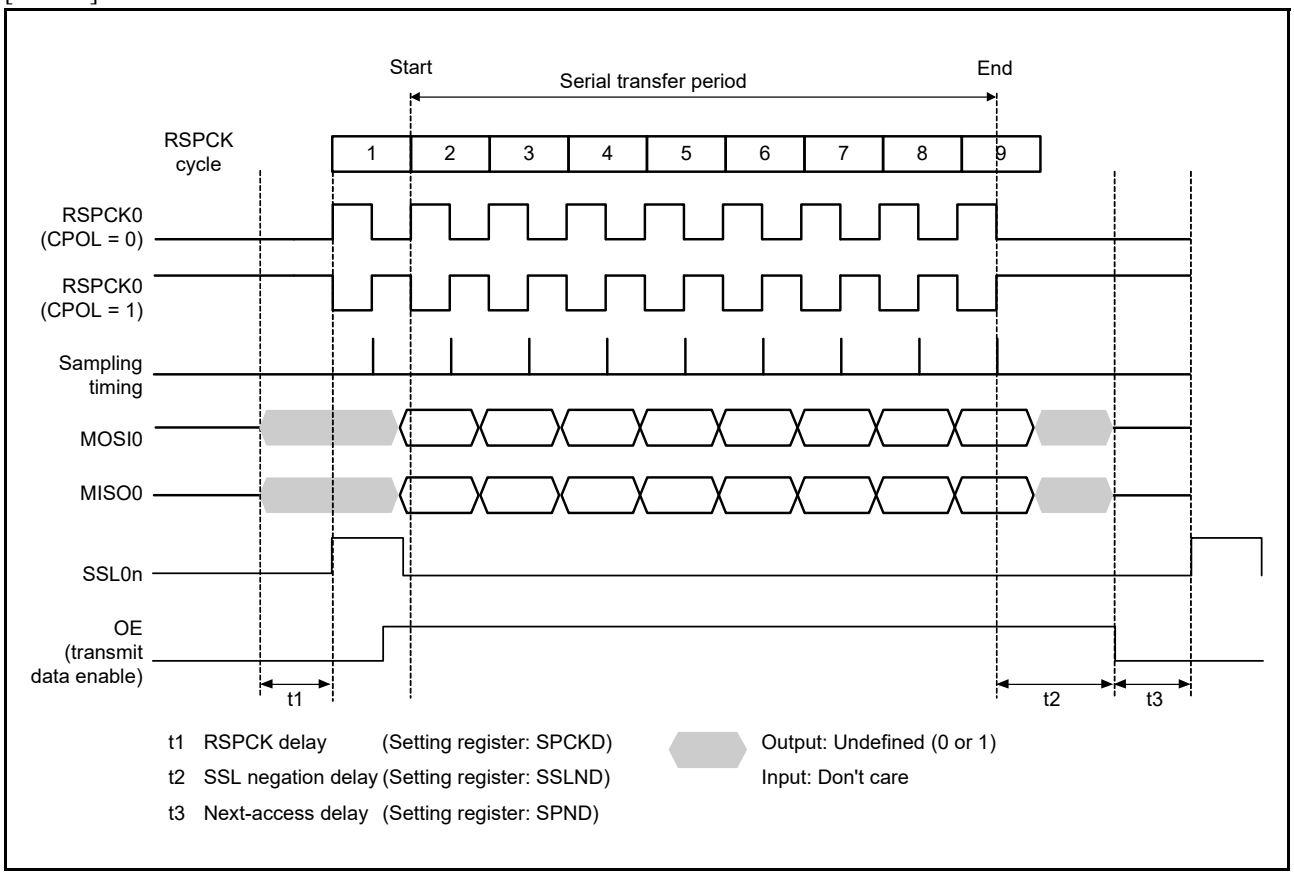


Figure 38.29 RSPI Transfer Format (CPHA = 1, FRFS = 1)

38.3.6 Communication Mode

Transmit-receive mode, transmit-only mode, and receive-only mode can be selected by the communication mode select bit (SPCR.CMMD[1:0]).

The SPDR access shown in Figure 38.30, Figure 38.31 and Figure 38.32 indicate the condition of access to the SPDR register, where W denotes a write cycle.

38.3.6.1 Transmit-Receive Mode (SPCR.CMMD[1:0] = 00b)

Figure 38.30 shows an example of operation when the communication mode select bits (SPCR.CMMD[1:0]) are set to 00b. In the example in Figure 38.30, the RSPIA performs an 8-bit serial transfer in which SPFCR.TTRG[1:0] = 00b, SPFCR.RTRG[1:0] = FIFO stage – 1, SPCMDm.CPHA = 1, and SPCMDm.CPOL = 0. The numbers given under the RSPCK0 waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

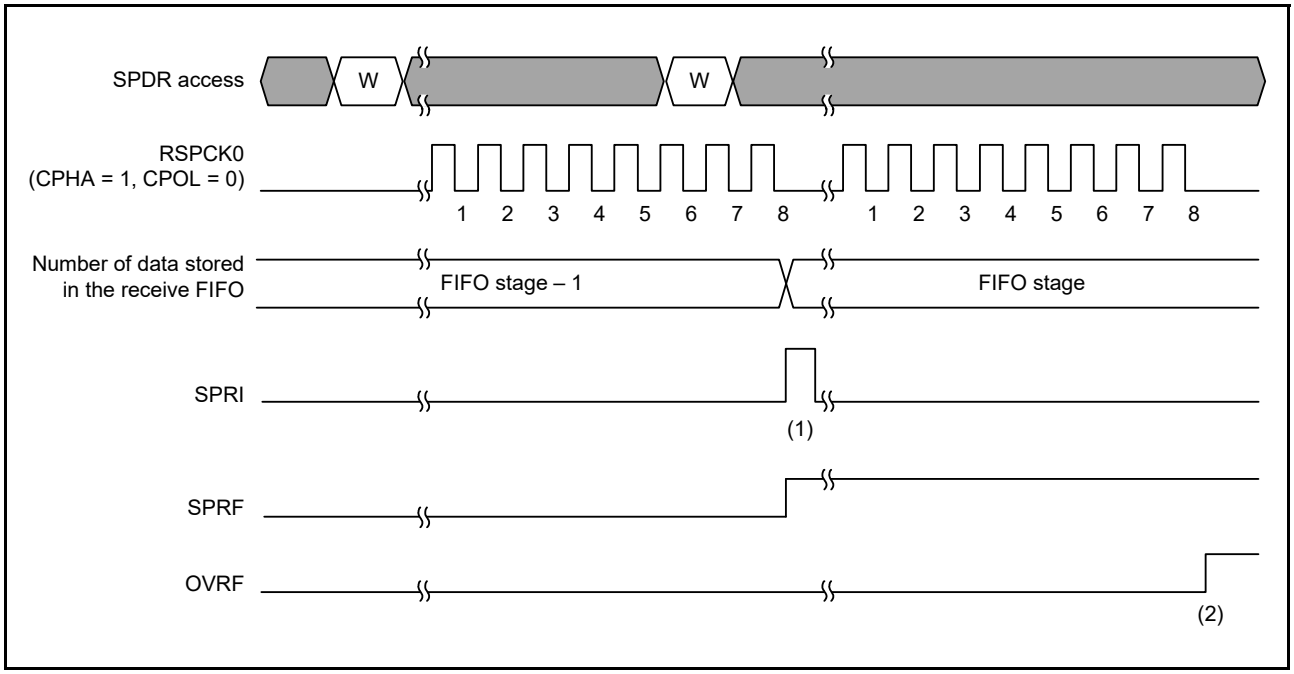


Figure 38.30 Operation Example of SPCR.CMMD[1:0] = 00b

The operation of the flags at timings shown in steps (1) and (2) in the figure is described below.

(1) When a serial transfer ends while the number of data stored in the SPDR receive buffer matches the number of frames set in SPFCR.RTRG[1:0] bits, the RSPIA generates a receive buffer full interrupt request (SPRI) (sets the SPSR.SPRF flag to 1) and copies the received data in the shift register to the receive buffer.

(2) When a serial transfer ends with data for the number of FIFO stages is stored in the SPDR receive buffer, the RSPIA sets the SPSR.OVRF flag to 1 and discards the received data in the shift register.

When transmit-receive mode (CMMD[1:0] = 00b) is selected, reception occurs simultaneously with transmit operations. As such, the SPRF and OVRF flags in the SPSR register become 1 at the timing described in (1) and (2), respectively, according to the state of the receive buffer.

38.3.6.2 Transmit-Only Mode (SPCR.CMMD[1:0] = 01b)

Figure 38.31 shows an example of operation when the communication mode select bits (SPCR.CMMD[1:0]) are set to 01b. In the example in Figure 38.31, the RSPIA performs an 8-bit serial transfer in which SPFCR.TTRG[1:0] = 00b, SPFCR.RTRG[1:0] = 00b, SPCMDm.CPHA = 1, and SPCMDm.CPOL = 0. The numbers given under the RSPCK0 waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

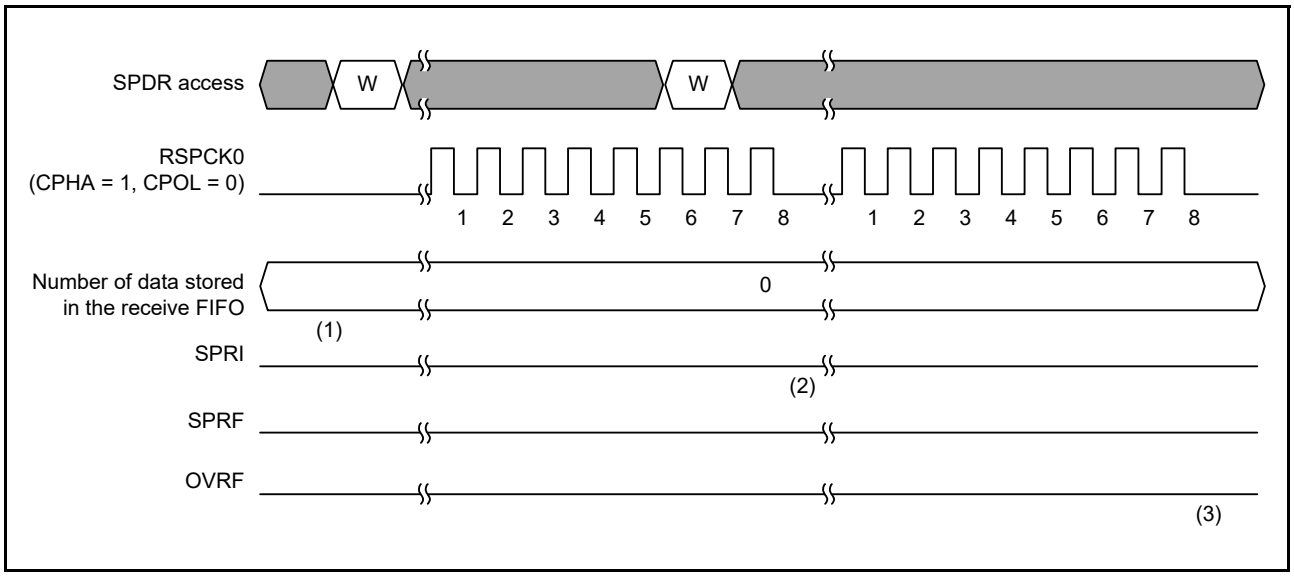


Figure 38.31 Operation Example of SPCR.CMMD[1:0] = 01b

The operation of the flags at timings shown in steps (1) to (3) in the figure is described below.

(1) Make sure there is no data left in the receive buffer and the SPSR.SPRF, OVRF flags are 0 before entering the mode of transmit operations only (SPCR.CMMD[1:0] = 01b).

(2) When a serial transfer ends without received data in the receive FIFO of SPDR, if transmit-only mode is selected (SPCR.CMMD[1:0] = 01b), the SPRF flag remains 0 and the RSPIA does not copy the data from the shift register to the receive buffer.

(3) Since the receive buffer of SPDR does not hold data that was received in the previous serial transfer, even when a serial transfer ends, the SPSR.OVRF flag retains the value of 0, and the data in the shift register is not copied to the receive buffer.

When transmit-only mode (SPCR.CMMD[1:0] = 01b) is selected, the RSPIA transmits data but does not receive data. Therefore, the SPSR.SPRF, OVRF flags remain 0 at the timings of (1) to (3).

38.3.6.3 Receive-Only Mode (SPCR.CMMD[1:0] = 10b)

Figure 38.32 shows an example of operation when the communication mode select bits (SPCR.CMMD[1:0]) are set to 10b. In the example in Figure 38.32, the RSPIA performs an 8-bit serial transfer in which SPFCR.TTRG[1:0] = 00b, SPFCR.RTRG[1:0] = FIFO stage – 1, SPCMDm.CPHA = 1, and SPCMDm.CPOL = 0. The numbers given under the RSPCK0 waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

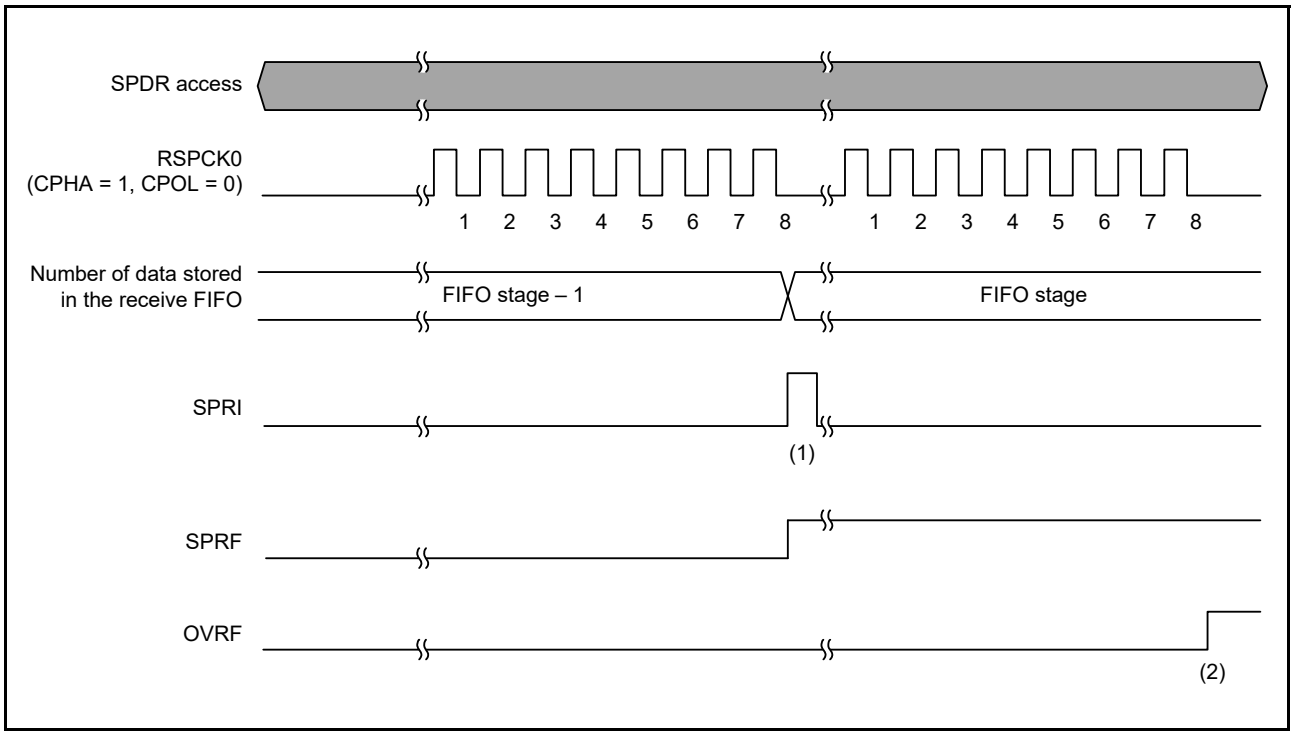


Figure 38.32 Operation Example of SPCR.CMMD[1:0] = 10b

The operation of the flags at timings shown in steps (1) to (2) in the figure is described below.

(1) When a serial transfer ends while the number of data stored in the SPDR receive buffer matches the number of frames set in SPFCR.RTRG[1:0] bits, the RSPIA generates a receive buffer full interrupt request (SPRI) (sets the SPSR.SPRF flag to 1) and copies the received data in the shift register to the receive buffer.

(2) When a serial transfer ends with data for the number of FIFO stages is stored in the SPDR receive buffer, the RSPIA sets the SPSR.OVRF flag to 1 and discards the received data in the shift register.

38.3.7 Transmit Buffer Empty/Receive Buffer Full Interrupts

Figure 38.33 shows an example of operation of the transmit buffer empty interrupt (SPTI) and the receive buffer full interrupt (SPRI). The SPDR register access shown in Figure 38.33 indicate the conditions of access to the SPDR register, where W denotes a write cycle, and R a read cycle. In the examples in Figure 38.33, the RSPIA performs an 8-bit serial transfer in which the SPCR.CMMD[1:0] bits are 00b, SPFCR.TTRG[1:0] bits are 00b, the SPFCR.RTRG[1:0] bits are 00b, the SPCMDm.CPHA bit is 1, and the SPCMDm.CPOL bit is 0. The numbers given under the RSPCK0 waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

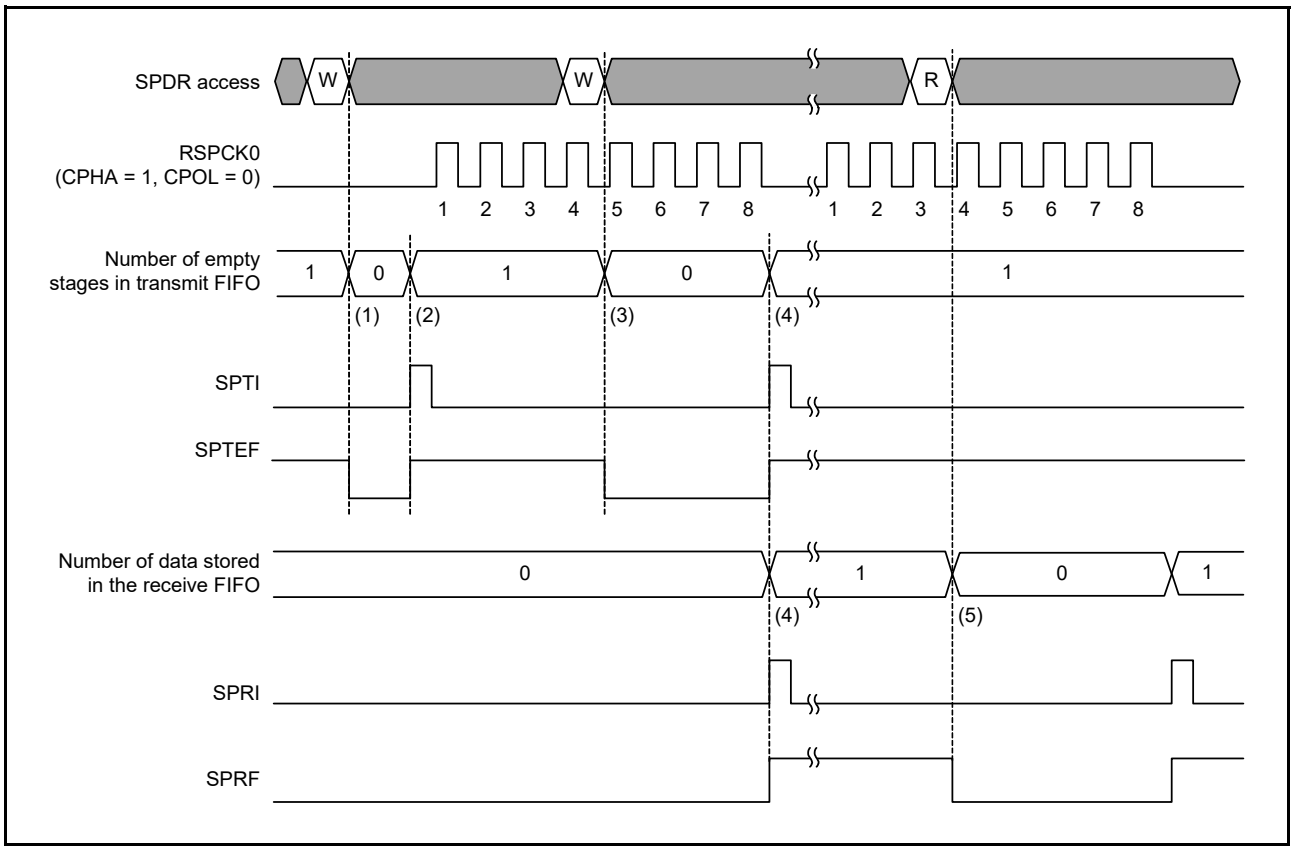


Figure 38.33 Operation Example of SPTI and SPRI Interrupts

The operation of the interrupts at timings shown in steps (1) to (5) in the figure is described below.

(1) If transmit data is written to SPDR when the next transmit data is not set in the transmit FIFO, the RSPIA writes data to the transmit buffer. When transmit data is written to SPDR in one processing routine using DTC/DMAC, the SPSR.SPTEF flag becomes 0 at the last access.

(2) If the shift register is empty, the RSPIA copies the data from the transmit buffer to the shift register. If the number of empty stage of the transmit FIFO is greater than the threshold value, the RSPIA generates a transmit buffer empty interrupt request (SPTI) and sets the SPSR.SPTEF flag to 1. How a serial transfer is started depends on the mode of the RSPIA. For details, refer to section 38.3.13, SPI Operation, and section 38.3.14, Clock Synchronous Operation.

(3) When transmit data is written to SPDR in the transmit buffer empty interrupt routine or in the transmit buffer empty detecting process by polling the SPTEF flag, the data is transferred to the transmit buffer. When transmit data is written to SPDR in one processing routine using DTC/DMAC, the SPSR.SPTEF flag becomes 0 at the last access. Because the data being transmitted is stored in the shift register, the RSPIA does not copy the data from the transmit buffer to the shift register.

(4) If the serial transfer ends (data sampling clock edge in the final bit is detected) when the data stored in the receive FIFO of SPDR < the number of FIFO stages, the RSPIA copies the receive data from the shift register to the receive buffer. At this time, if the data stored in the receive FIFO is greater than the threshold value, the RSPIA generates a

receive buffer full interrupt request (SPRI), and sets the SPSR.SPRF flag to 1. Since the shift register becomes empty upon completion of the serial transfer, when the next transmit data had been set in the transmit FIFO before the serial transfer ended, the RSPIA sets the SPSR.SPTEF flag to 1 and copies the data from the transmit 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 RSPIA determines that the shift register is empty, thus data transfer from the transmit buffer to the shift register is enabled.

(5) When SPDR is read in the receive buffer full interrupt (SPRI) routine or in the receive buffer full detecting process by polling the SPRF flag, the receive data can be read. If the received data is read from SPDR in one processing routine using DTC/DMAC, the SPRF flag becomes 0 at the last access.

If transmit data is written to SPDR while no empty stages in the transmit FIFO, the RSPIA does not update the data in the transmit buffer. Transmit data should be written to SPDR in the transmit buffer empty interrupt request routine or in the transmit buffer empty detecting process by polling the SPTEF flag.

When setting the SPCR.SPE bit to 0 (RSPI function disabled), if the SPCR.SPTIE is 1, a transmit buffer empty interrupt request will occur. However, transmit buffer empty interrupt is inhibited by disabling transmit buffer empty interrupt (the SPTIE bit is 0) at the same time as SPE bit setting.

When serial transfer ends while data is stored in the receive FIFO for the number of FIFO stages, the RSPIA does not copy data from the shift register to the receive buffer, and detects an overrun error (refer to section 38.3.10, Error Detection). To prevent a receive data overrun error, read the received data before the next serial transfer ends.

Transmit buffer empty interrupt and receive buffer full interrupt or the corresponding IRn.IR flags (where n is the interrupt vector number) in the ICU can be used to confirm the states of the transmit and receive buffers. The status of the transmit and receive buffers can be also confirmed by the SPTEF and SPRF flags.

38.3.8 Idle Interrupts

When the SPSSR.SPCP[2:0] bits becomes 000b, if the next transmit data is not set, the SPSR.IDLNF flag is set to 0 and an idle interrupt request is generated. An interrupt request is also made by setting the SPCR.SPE bit to 0.

[Motorola SPI]

Figure 38.34 shows an example of idle interrupt operation during normal operation.

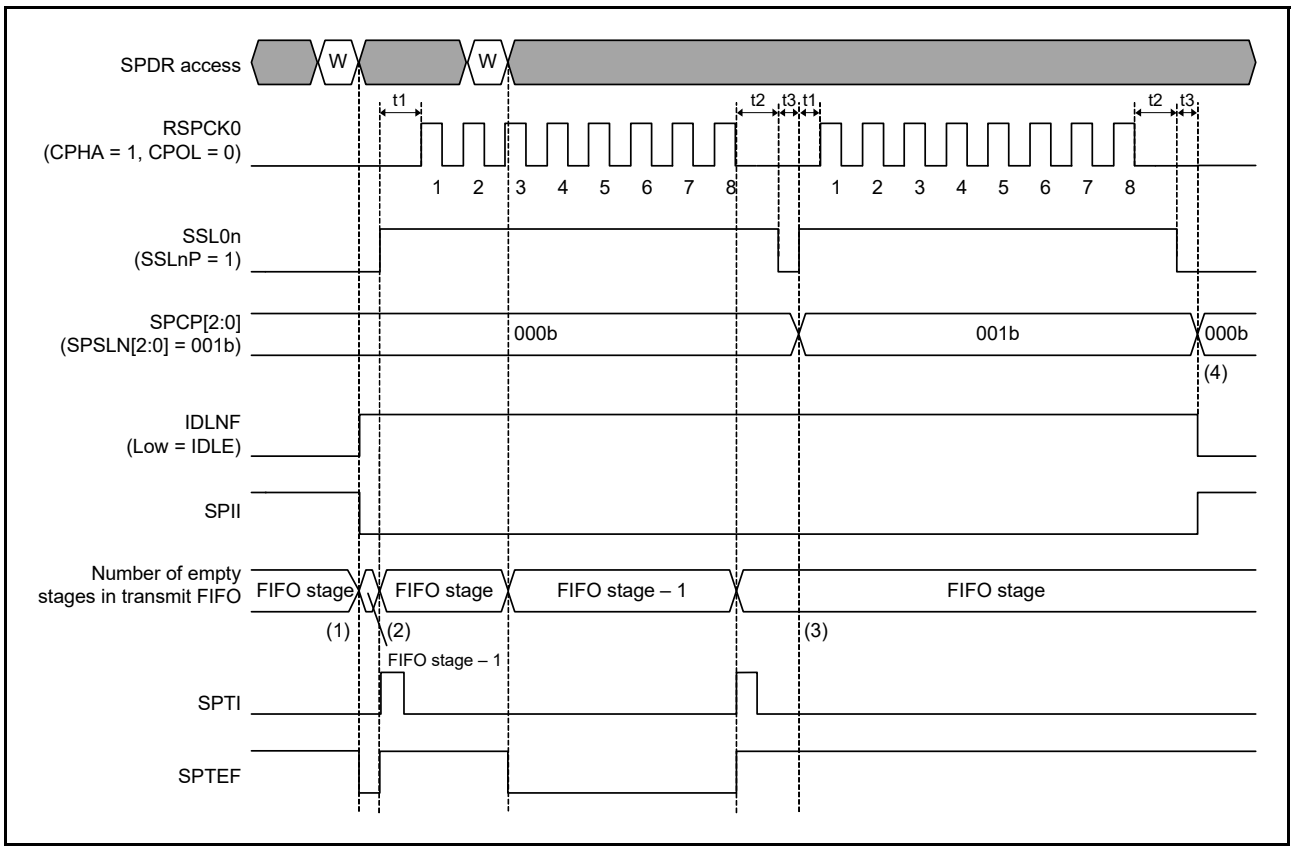


Figure 38.34 Operation Example of Idle Interrupts (Master Mode, Motorola SPI)

(1) At the start of transmission, if the next transmit data is not set in the transmission buffer, the SPSR.IDLNF flag becomes 0 (IDLE). Writing transmit data sets the SPSR.IDLNF flag to 1 (BUSY). When the SPCR.SPIIE bit is set to 1 before transmit data is written, interrupt processing is required before transmission start. For this reason, set the SPCR.SPIIE bit to 0 before starting transmission.

(2) After transmission has started, the SPSR.IDLNF flag remains 1 (BUSY) regardless of the transmit buffer state.

(3) The SPSSR.SPCP[2:0] bits change the command to the next one at the end of t3 cycle. When the next command is not 000b, the SPSR.IDLNF flag remains unchanged even when the next transmit data has not been written.

(4) The SPSR.IDLNF flag becomes 0 (IDLE) at the end of t3 cycle because the next command is 000b and the next transmit data is not present. When the SPCR.SPIIE bit is 1, an SPII interrupt is output.

[TI SSP]

Figure 38.35 shows an example of idle interrupt operation during normal operation.

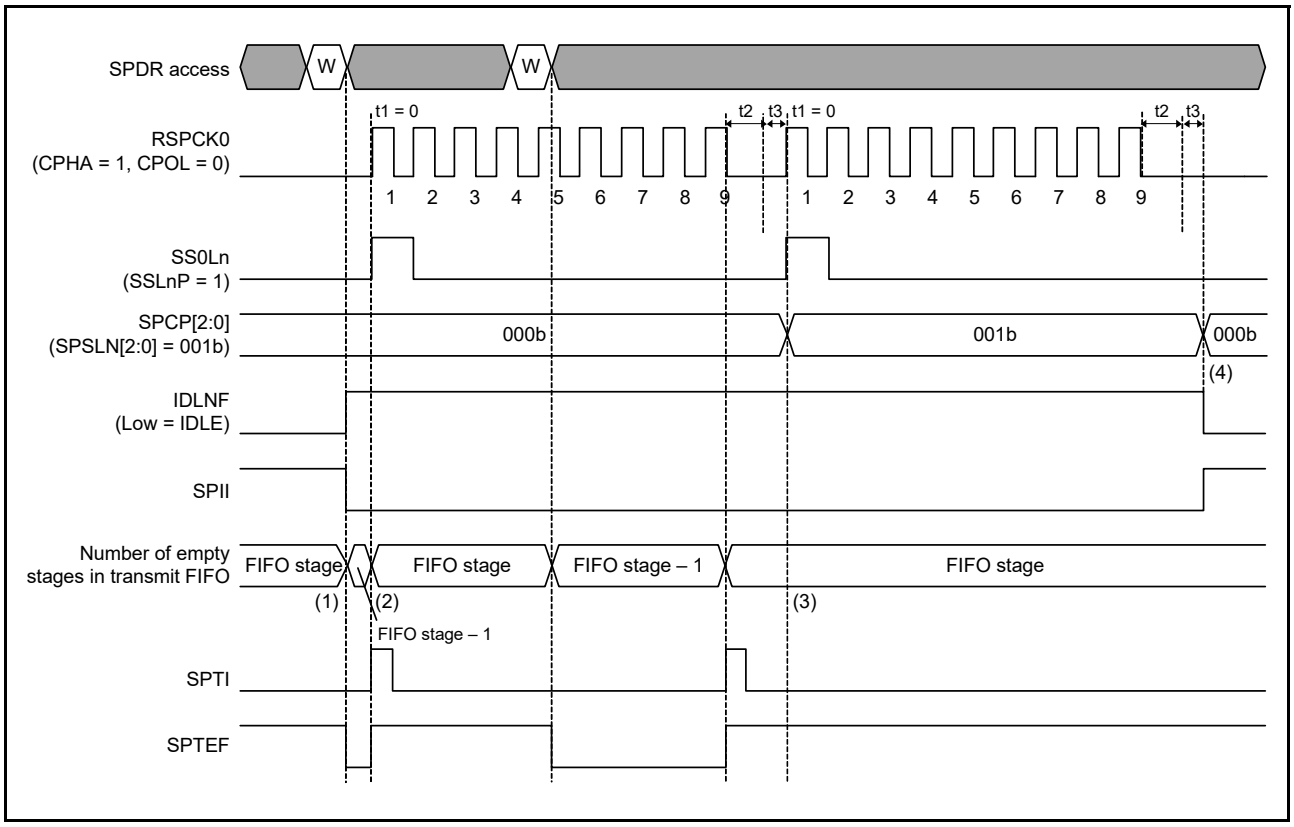


Figure 38.35 Operation Example of Idle Interrupts (Master Mode, TI SSP)

(1) At the start of transmission, if the next transmit data is not set in the transmission buffer, the SPSR.IDLNF flag becomes 0 (IDLE). Writing transmit data sets the SPSR.IDLNF flag to 1 (BUSY). When the SPCR.SPIIE bit is set to 1 before transmit data is written, interrupt processing is required before transmission start. For this reason, set the SPCR.SPIIE bit to 0 before starting transmission.

(2) After transmission has started, the SPSR.IDLNF flag remains 1 (BUSY) regardless of the transmit buffer state.

(3) The SPSSR.SPCP[2:0] bits change the command to the next one at the end of t3 cycle. When the next command is not 000b, the SPSR.IDLNF flag remains unchanged even when the next transmit data has not been written.

(4) The SPSR.IDLNF flag becomes 0 (IDLE) the end of t3 cycle because the next command is 000b and the next transmit data is not present. When the SPCR.SPIIE bit is 1, an SPII interrupt is output.

An idle interrupt during slave mode operation depends only on the value of the SPCR.SPE bit.

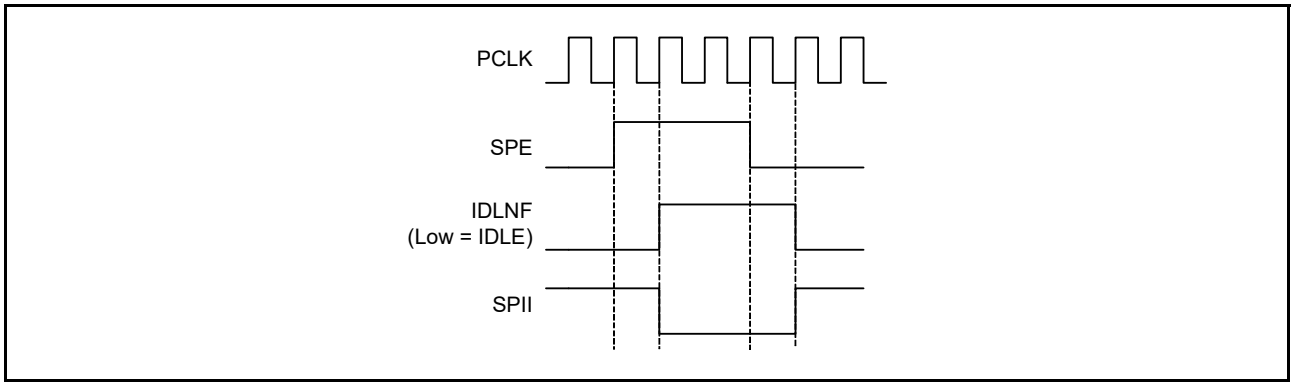


Figure 38.36 Operation Example of Idle Interrupts (Slave Mode)

38.3.9 Communication End Interrupts

38.3.9.1 Transmit-Receive Master Mode or Transmit-Only Master Mode

Refer to the description of the SPCF Flag (Communication End Flag) in section 38.2.16, RSPI Status Register (SPSR) for setting/clearing conditions of the communication end flag during transmit-receive master mode or transmit-only master mode.

[Motorola SPI]

Figure 38.37 shows an example of the SPCI interrupt operation during transmit-receive master mode/transmit-only master mode.

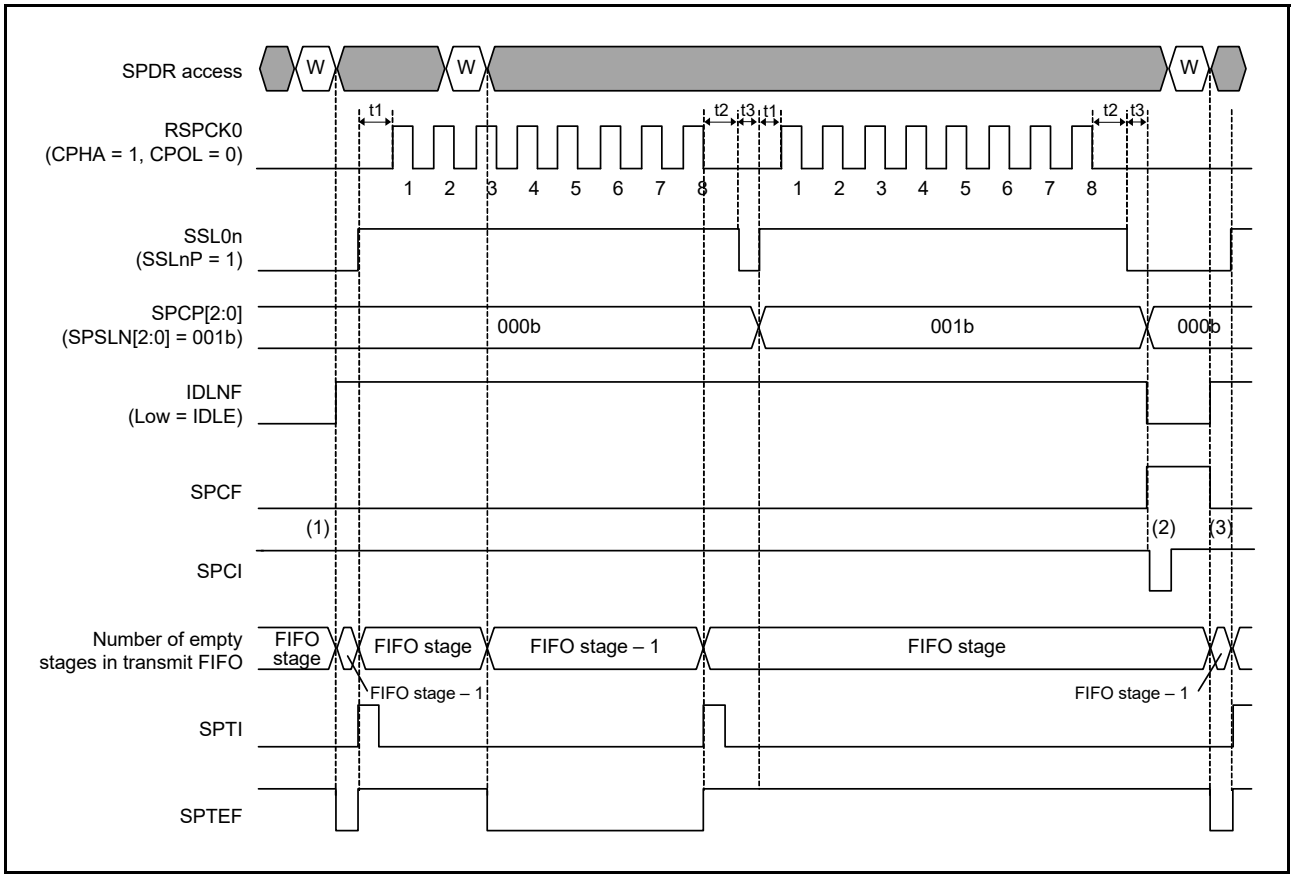


Figure 38.37 Operation Example of SPCI Interrupts (Transmit-Receive Master Mode/Transmit-Only Master Mode, Motorola SPI)

(1) The SPCF flag is 0 and the level of the SPCI interrupt is 1 before starting communication. And these have kept during communication.

(2) The SPCF flag becomes 1 (communication end) at the end of t3 cycle, because the next command is 000b and there is no next transmit data. And then the SPCI interrupt outputs with PCLK 1 cycle width if the SPCR.SPCIE bit is 1.

(3) The SPCF flag becomes 0 when the next transmit data is written to the transmit buffer (SPTX). Or when 1 is written to the SPSCLR.SPCFC bit, then the SPCF flag becomes 0.

[TI SSP]

Figure 38.38 shows an example of the SPCI interrupt operation during transmit-receive master mode/transmit-only master mode.

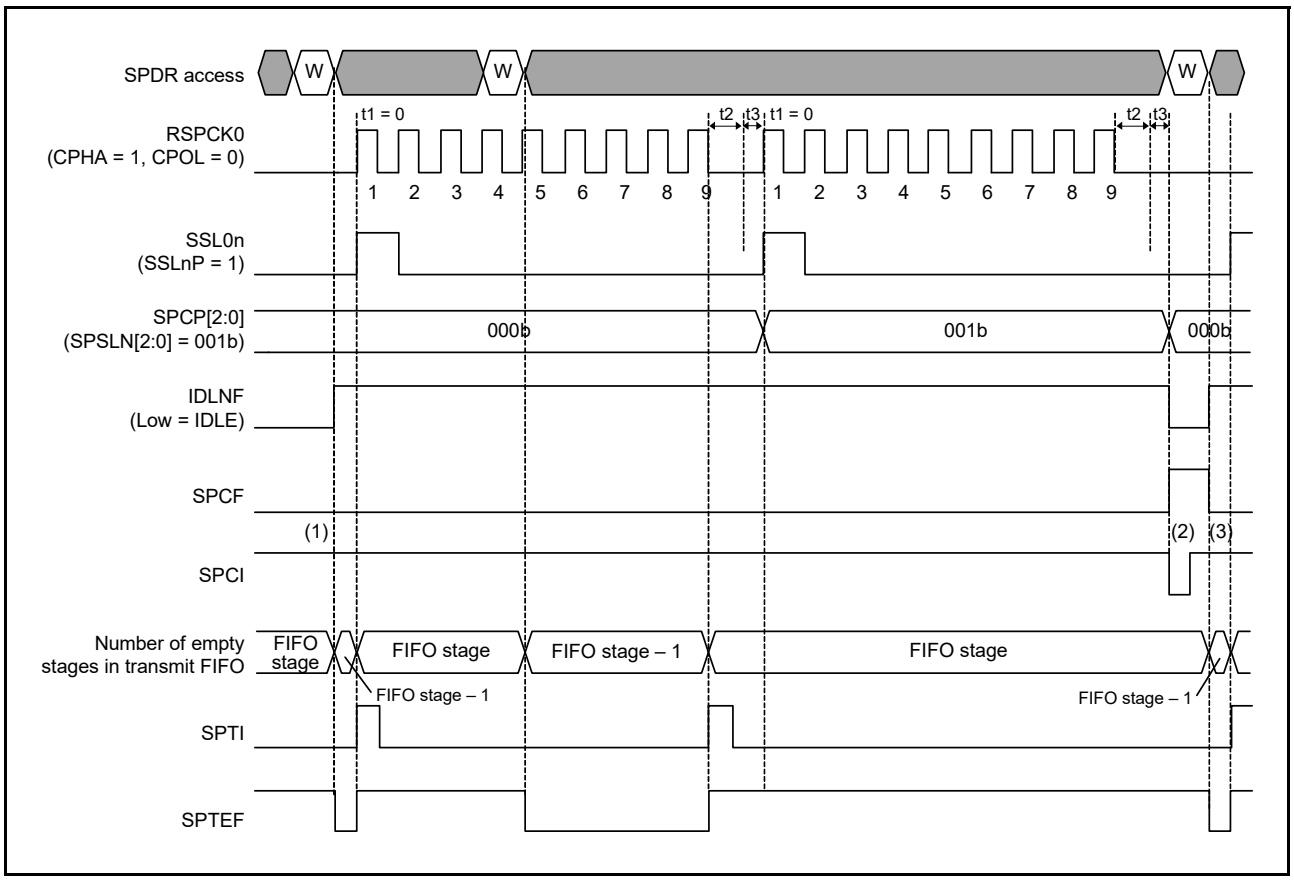


Figure 38.38 Operation Example of SPCI Interrupts (Transmit-Receive Master Mode/Transmit-Only Master Mode, TI SSP)

(1) The SPCF flag is 0 and the level of the SPCI interrupt is 1 before starting communication. And these have kept during communication.

(2) The SPCF flag becomes 1 (communication end) at the end of t3 cycle, because the next command is 000b and there is no next transmit data. And then the SPCI interrupt outputs with PCLK 1 cycle width if the SPCR.SPCIE bit is 1.

(3) The SPCF flag becomes 0 when the next transmit data is written to the transmit buffer (SPTX). Or when 1 is written to the SPSCLR.SPCFC bit, then the SPCF flag becomes 0.

38.3.9.2 Receive-Only Master Mode

Refer to the description of the SPCF Flag (Communication End Flag) in section 38.2.16, RSPI Status Register (SPSR) for setting/clearing conditions of the communication end flag during receive-only master mode.

Figure 38.39 shows an example of the SPCI interrupt operation during receive-only master mode when the SPRMCR.RFC[4:0] bits are 00000b.

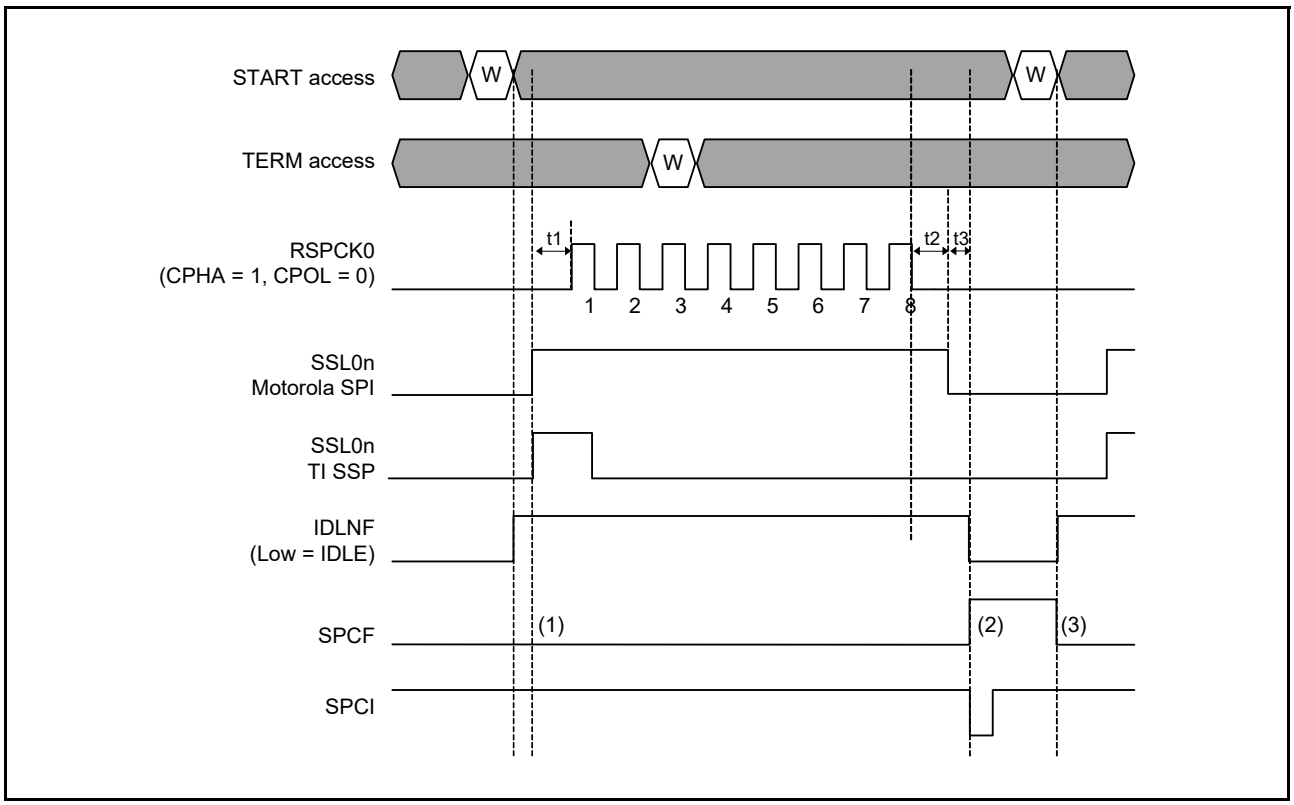


Figure 38.39 Operation Example of SPCI Interrupts (RFC[4:0] = 00000b) (Receive-Only Master Mode)

(1) The SPCF flag is 0 and the level of the SPCI interrupt is 1 before starting communication. And these have kept during communication.

(2) When the SPRMCR.TERM bit is set to 1 during communication, the SPCF flag becomes 1 (communication end) at the end of t3 cycle. And then the SPCI interrupt outputs with PCLK 1 cycle width if the SPCR.SPCIE bit is 1.

(3) The SPCF flag becomes 0 when 1 is written to the SPRMCR.START bit. Or when 1 is written to the SPSCLR.SPCFC bit, then the SPCF flag becomes 0.

Figure 38.40 shows an example of the SPCI interrupt operation during receive-only master mode when the SPRMCR.RFC[4:0] bits are not 00000b.

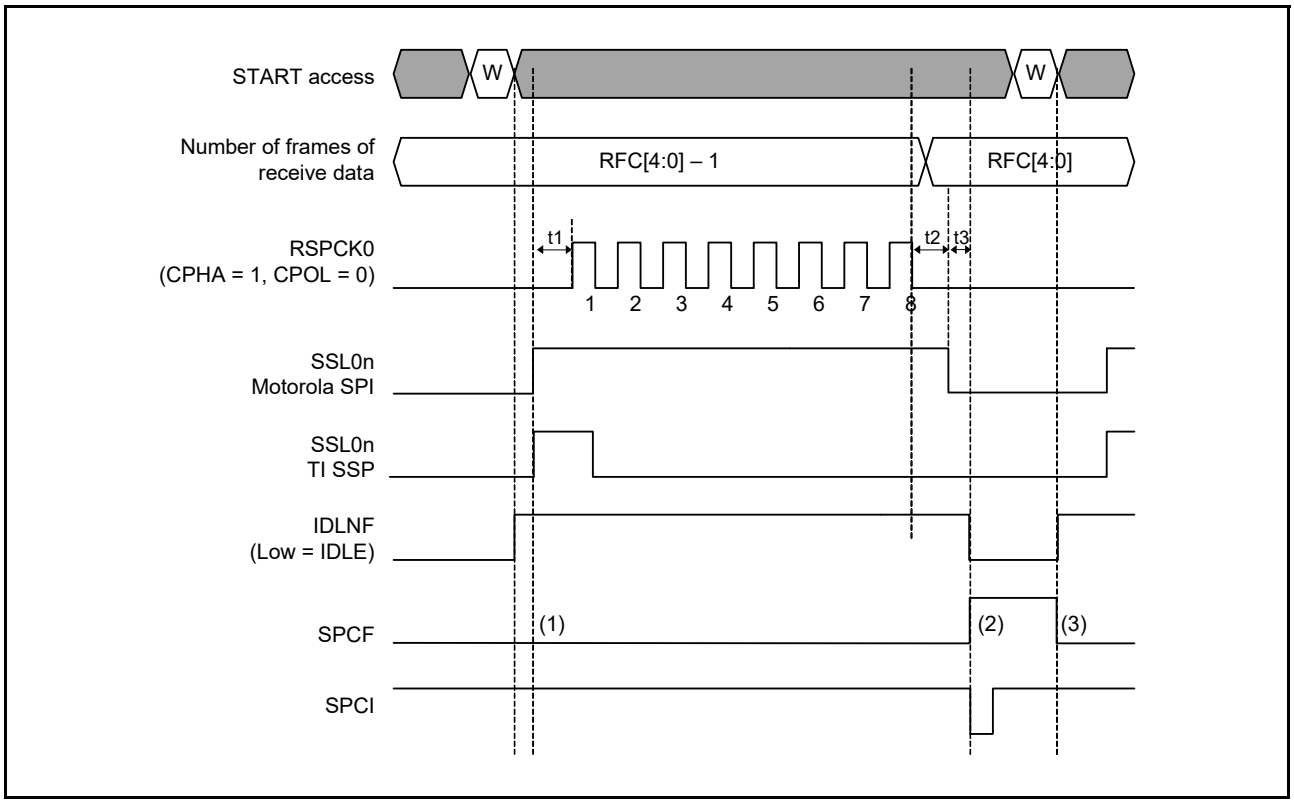


Figure 38.40 Operation Example of SPCI Interrupts (RFC[4:0] ≠ 00000b) (Receive-Only Master Mode)

(1) The SPCF flag is 0 and the level of the SPCI interrupt is 1 before starting communication. And these have kept during communication.
(2) The SPCF flag becomes 1 (communication end) at the end of t3 cycle after receiving the number of frames set by the SPRMCR.RFC[4:0] bits. And then the SPCI interrupt outputs with PCLK 1 cycle width if the SPCR.SPCIE bit is 1.
(3) The SPCF flag becomes 0 when 1 is written to the SPRMCR.START bit. Or when 1 is written to the SPSCLR.SPCFC bit, then the SPCF flag becomes 0.

In slave mode operation, the output timing of the SPCI interrupt is deferent due to the value of the SPCR.SPMS bit (RSPI mode select bit). And the clear timing of the communication end interrupt is deferent due to the communication mode (transmit-receive, transmit-only, or receive-only).

38.3.9.3 Transmit-Receive Slave Mode or Transmit-Only Slave Mode on SPI Operation (4-Wire Method)

Refer to the description of the SPCF Flag (Communication End Flag) in section 38.2.16, RSPI Status Register (SPSR) for setting/clearing conditions of the communication end flag during transmit-receive slave mode or transmit-only slave mode on SPI operation (4-wire method).

[Motorola SPI]

Figure 38.41 shows an example of the SPCI interrupt operation during transmit-receive slave mode/transmit-only slave mode on SPI operation.

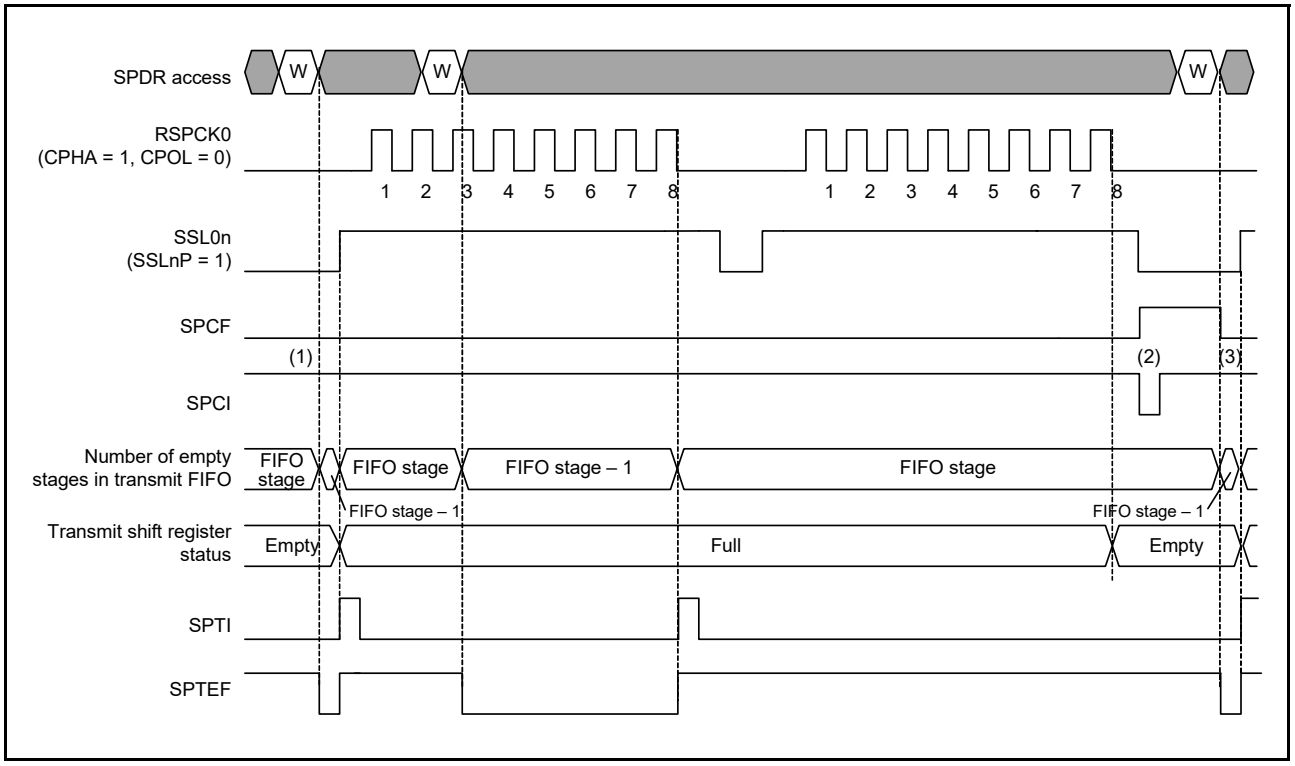


Figure 38.41 Operation Example of SPCI Interrupts (Transmit-Receive Slave Mode/Transmit-Only Slave Mode, Motorola SPI)

(1) The SPCF flag is 0 and the level of the SPCI interrupt is 1 before starting communication. And these have kept during communication.

(2) The SPCF flag becomes 1 (communication end) at the timing of SSL0n negation, when the next transmit data is not set in the transmit FIFO and the transmit shift register is empty. And then the SPCI interrupt outputs with PCLK 1 cycle width if the SPCR.SPCIE bit is 1.

(3) The SPCF flag becomes 0 when the next transmit data is written to the transmit buffer (SPTX). Or when 1 is written to the SPSCLR.SPCFC bit, then the SPCF flag becomes 0.

[TI SSP]

Figure 38.42 shows an example of the SPCI interrupt operation during transmit-receive slave mode/transmit-only slave mode on SPI operation.

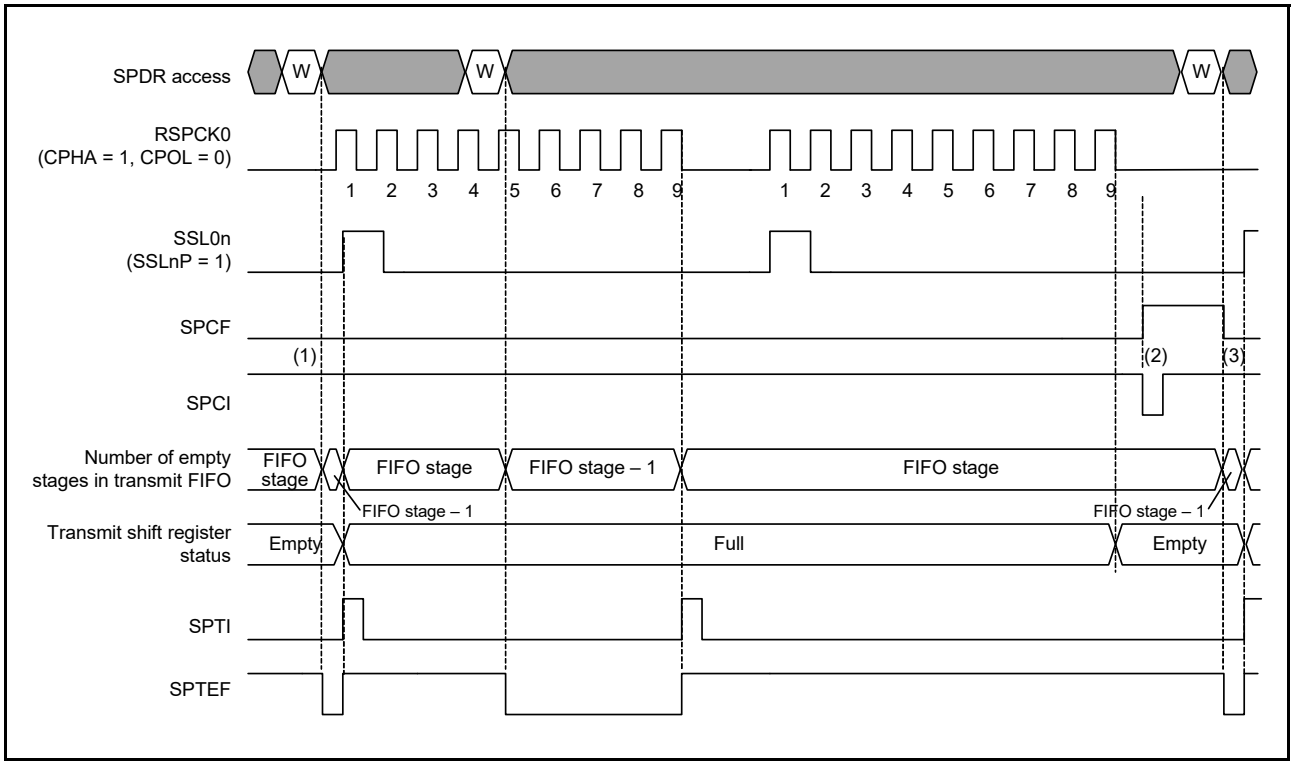


Figure 38.42 Operation Example of SPCI Interrupts (Transmit-Receive Slave Mode/Transmit-Only Slave Mode, TI SSP)

(1) The SPCF flag is 0 and the level of the SPCI interrupt is 1 before starting communication. And these have kept during communication.

(2) The SPCF flag becomes 1 (communication end) when the next transmit data is not set in the transmit FIFO, the transmit shift register is empty, and SSL negation delay is completed. And then the SPCI interrupt outputs with PCLK 1 cycle width if the SPCR.SPCIE bit is 1.

(3) The SPCF flag becomes 0 when the next transmit data is written to the transmit buffer (SPTX). Or when 1 is written to the SPSCLR.SPCFC bit, then the SPCF flag becomes 0.

38.3.9.4 Receive-Only Slave Mode on SPI Operation (4-Wire Method)

Refer to the description of the SPCF Flag (Communication End Flag) in section 38.2.16, RSPI Status Register (SPSR) for setting/clearing conditions of the communication end flag during receive-only slave mode on SPI operation (4-wire method).

[Motorola SPI]

Figure 38.43 shows an example of the SPCI interrupt operation during receive-only slave mode on SPI operation.

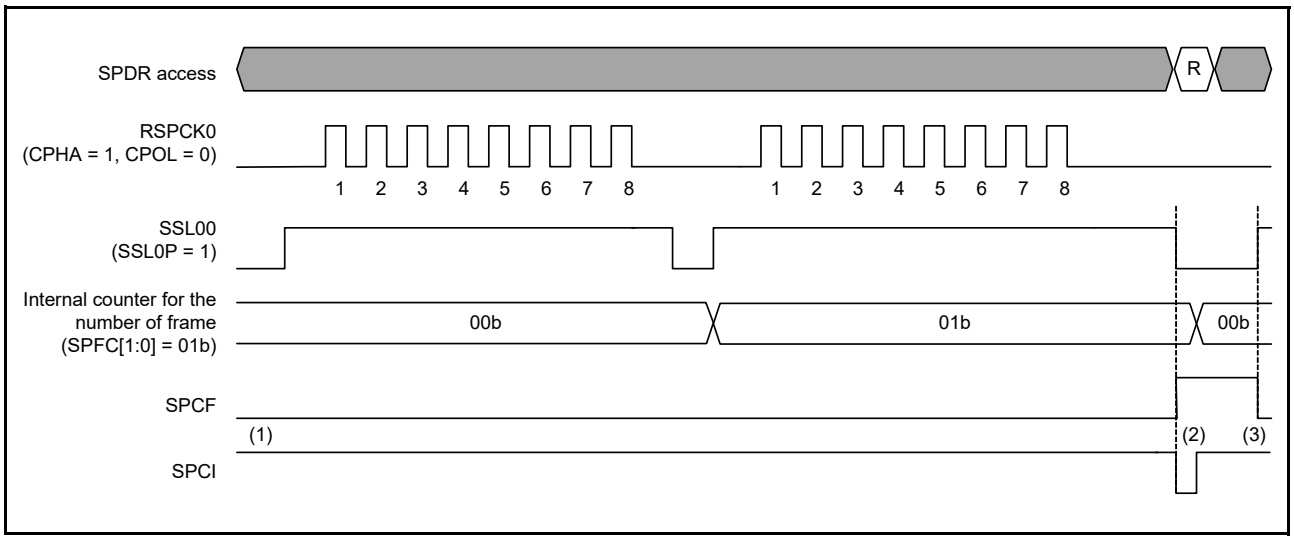


Figure 38.43 Operation Example of SPCI Interrupts (Receive-Only Slave Mode, Motorola SPI)

(1) The SPCF flag is 0 and the level of the SPCI interrupt is 1 before starting communication. And these have kept during communication.

(2) After the frames of the value set in the SPDCR.SPFC[1:0] bits are stored in the receive buffer, the SPCF flag becomes 1 (communication end) at the timing of SSL00 negation. And then the SPCI interrupt outputs with PCLK 1 cycle width if the SPCIE bit is 1.

(3) The SPCF flag becomes 0 at the SSL00 assertion when the next transmission start. Or when 1 is written to the SPSCLR.SPCFC bit, then the SPCF flag becomes 0.

[TI SSP]

Figure 38.44 shows an example of the SPCI interrupt operation during receive-only slave mode on SPI operation.

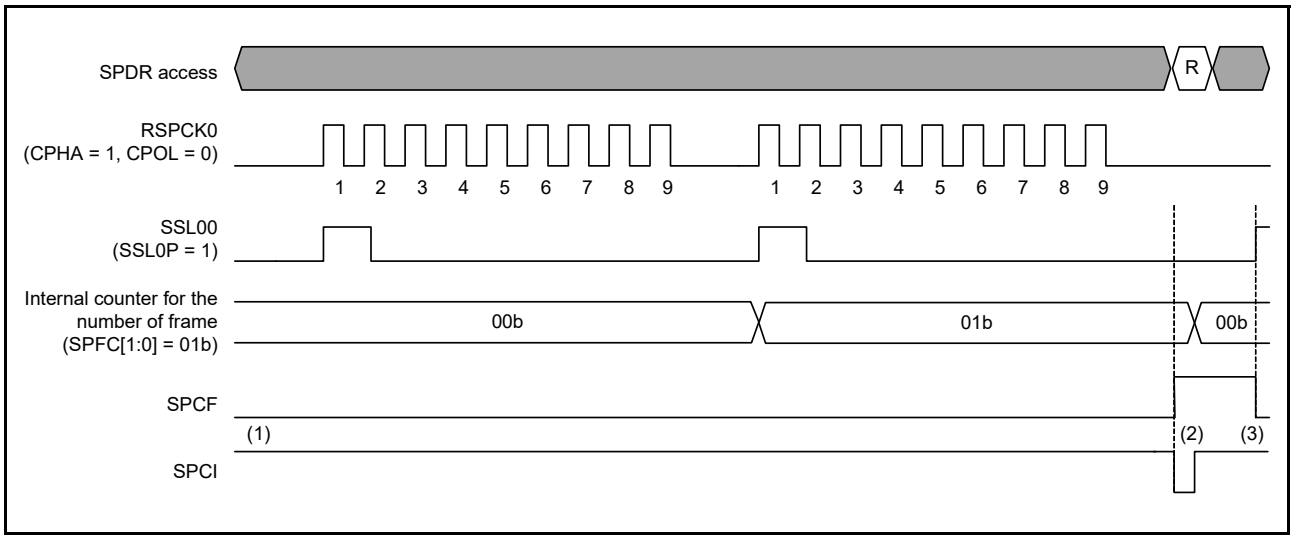


Figure 38.44 Operation Example of SPCI Interrupts (Receive-Only Slave Mode, TI SSP)

(1) The SPCF flag is 0 and the level of the SPCI interrupt is 1 before starting communication. And these have kept during communication.

(2) After the frames of the value set in the SPDCR.SPFC[1:0] bits are stored in the receive buffer, the SPCF flag becomes 1 (communication end) at the timing when SSL00 negation delay is completed. And then the SPCI interrupt outputs with PCLK 1 cycle width if the SPCIE bit is 1.

(3) The SPCF flag becomes 0 at the SSL00 assertion when the next transmission start. Or when 1 is written to the SPSCLR.SPCFC bit, then the SPCF flag becomes 0.

38.3.9.5 Transmit-Receive Slave Mode or Transmit-Only Slave Mode on Clock Synchronous Operation (3-Wire Method)

Refer to the description of the SPCF Flag (Communication End Flag) in section 38.2.16, RSPI Status Register (SPSR) for setting/clearing conditions of the communication end flag during transmit-receive slave mode or transmit-only slave mode on clock synchronous operation (3-wire method).

Figure 38.45 shows an example of the SPCI interrupt operation during transmit-receive slave mode/transmit-only slave mode on clock synchronous operation.

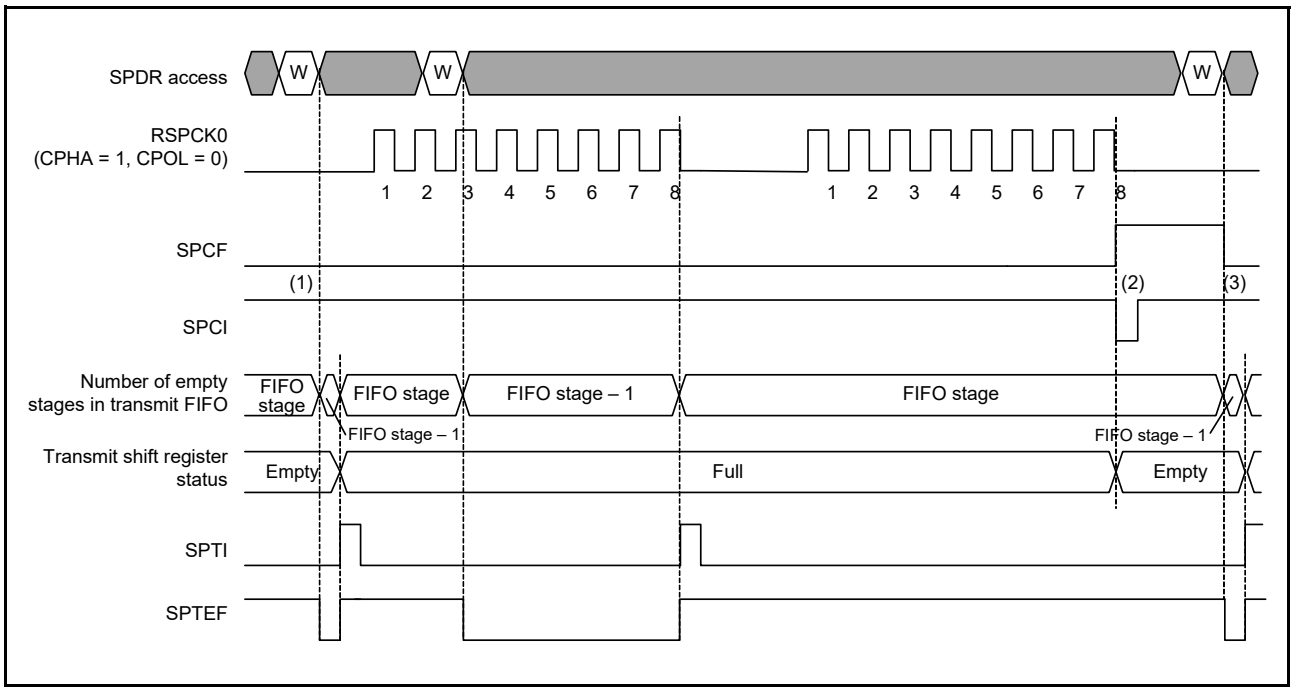


Figure 38.45 Operation Example of SPCI Interrupts (Transmit-Receive Slave Mode/Transmit-Only Slave Mode, Clock Synchronous Operation)

(1) The SPCF flag is 0 and the level of the SPCI interrupt is 1 before starting communication. And these have kept during communication.

(2) The SPCF flag becomes 1 (communication end) at the sampling timing of the final data bit of RSPCK0, when the next transmit data is not set in the transmit buffer and the transmit shift register is empty. And then the SPCI interrupt outputs with PCLK 1 cycle width if the SPCR.SPCIE bit is 1.

(3) The SPCF flag becomes 0 when the next transmit data is written to the transmit buffer (SPTX). Or when 1 is written to the SPSCLR.SPCFC bit, then the SPCF flag becomes 0.

38.3.9.6 Receive-Only Slave Mode on Clock Synchronous Operation (3-Wire Method)

Refer to the description of the SPCF Flag (Communication End Flag) in section 38.2.16, RSPI Status Register (SPSR) for setting/clearing conditions of the communication end flag during receive-only slave mode on clock synchronous operation (3-wire method).

Figure 38.46 shows an example of the SPCI interrupt operation during receive-only slave mode on clock synchronous operation.

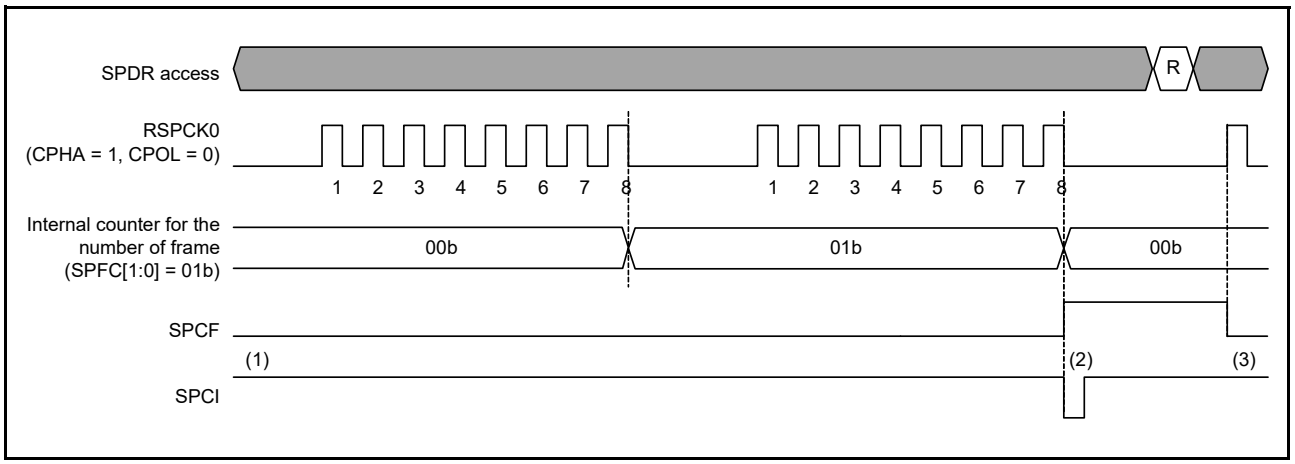


Figure 38.46 Operation Example of SPCI Interrupts (Receive-Only Slave Mode, Clock Synchronous Operation)

(1) The SPCF flag is 0 and the level of the SPCI interrupt is 1 before starting communication. And these have kept during communication.

(2) The SPCF flag becomes 1 (communication end) at the sampling timing of the final data bit of RSPCK0, when the final frame of the value set in the SPDCR.SPFC[1:0] bits is received. And then the SPCI interrupt outputs with PCLK 1 cycle width if the SPCR.SPCIE bit is 1.

(3) The SPCF flag becomes 0 at the first edge of RSPCK0 for the next transmission. Or when 1 is written to the SPSCLR.SPCFC bit, then the SPCF flag becomes 0.

38.3.10 Error Detection

In the normal serial transfer of the RSPIA, the data written to the transmit buffer of SPDR is transmitted, and the received data can be read from the receive buffer of SPDR. If access is made to SPDR, depending on the status of the transmit/receive buffer or the status of the RSPIA 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 RSPIA detects the event as an underrun error, overrun error, parity error, or mode fault error. Table 38.8 lists the relationship between non-normal transfer operations and the RSPIA's error detection function.

Table 38.8 Relationship between Non-Normal Transfer Operations and Error Detection Function of RSPIA

	Occurrence Condition	RSPI Operation	Error Detection
1	The SPDR register is written while no empty stages in the transmit FIFO.	• The contents of the transmit buffer are kept. • Missing write data.	None
2	The SPDR register is read while no empty stages in the receive FIFO.	If reception has completed, then the received data is output to the bus. Otherwise, data received previously is output instead.	None
3	Serial transfer is started in transmit-receive slave mode or transmit-only slave mode when transmit data is not ready to output.	• Serial transfer is suspended • Transmit/receive data is missing • The MISO0 output signal is disabled • RSPI function is disabled	Underrun error Refer to section 38.3.10.4, Underrun Error for underrun error.
4	Serial transfer terminates when data is stored in the receive FIFO with no empty FIFO stage.	• The contents of the receive FIFO are kept • Missing receive data	Overrun error Refer to section 38.3.10.1, Overrun Error for overrun error.
5	An incorrect parity bit is received with the parity function enabled in the transmit-receive mode or receive-only mode	The parity error flag is asserted.	Parity error Refer to section 38.3.10.2, Parity Error for parity error.
6	The SSL00 input signal is asserted when the serial transfer is idle in multi-master mode.	• Driving of the RSPCK0, MOSI0, SSL01 to SSL03 output signals is stopped. • RSPI function is disabled.	Mode fault error Refer to section 38.3.10.3, Mode Fault Error for mode fault error.
7	The SSL00 input signal is asserted during serial transfer in multi-master mode.	• Serial transfer is suspended. • Missing transmit/receive data. • Driving of the RSPCK0, MOSI0, SSL01 to SSL03 output signals is stopped. • RSPI function is disabled.	
8	The SSL00 input signal is negated during serial transfer in slave mode. [Motorola SPI]	• Serial transfer is suspended. • Missing transmit/receive data. • Driving of the MISO0 output signal is stopped. • RSPI function is disabled.	
9	The SSL00 input signal is asserted during serial transfer in slave mode. [TI SSP]	• Serial transfer is suspended. • Missing transmit/receive data. • Driving of the MISO0 output signal is stopped. • RSPI function is disabled.	
10	After data is stored in the receive FIFO with RDRIS = 1, the number of stored data is less than the threshold value and no receive data is written for the set value of the SPDRCSR register.	• The receive data ready flag is asserted.	Receive data ready Refer to section 38.3.11, Received Data Ready Detection for receive data ready

On operation 1 in Table 38.8, the RSPIA does not detect an error. To prevent data omission during the writing to SPDR, the SPDR register should be written when a transmit buffer empty interrupt request occurs or while the SPSR.SPTEF flag is 1.

Likewise, the RSPIA does not detect an error on operation 2. To prevent extraneous data from being read, the SPDR register should be read when a receive buffer full interrupt request occurs or while the SPSR.SPRF flag is 1.

For the transmit and receive interrupts, refer to section 38.3.7, Transmit Buffer Empty/Receive Buffer Full Interrupts.

38.3.10.1 Overrun Error

If a serial transfer ends while no empty stages in the receive FIFO, the RSPIA detects an overrun error, and sets the SPSR.OVRF flag to 1. When the OVRF flag is 1, the RSPIA 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 set the SPSR.OVRF flag to 0, issue a system reset or write 1 to the SPSRC.OVRFC bit.

Figure 38.47 shows an example of operations of the SPSR.SPRF and OVRF flags. The SPSCLR and SPDR accesses shown in Figure 38.47 indicate the condition of accesses to SPSCLR and SPDR, respectively, where W denotes a write cycle, and R a read cycle. In the example in Figure 38.47, the RSPIA performs an 8-bit serial transfer in which the SPCMDm.CPHA bit is 1 and the SPCMDm.CPOL bit is 0. The numbers given under the RSPCK0 waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

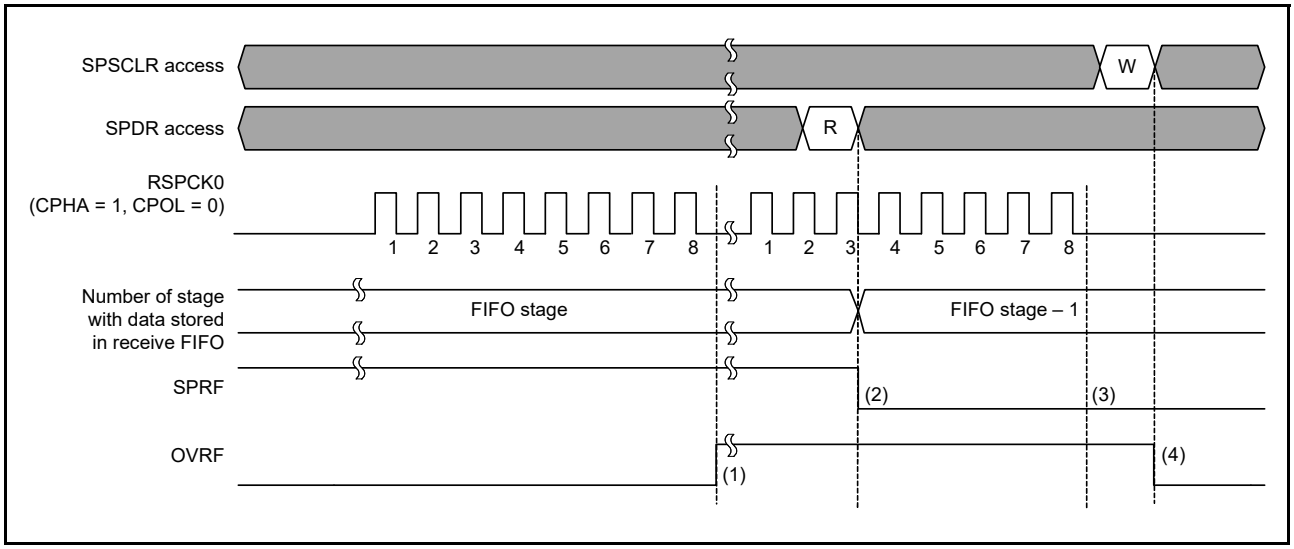


Figure 38.47 Operation Example of SPRF and OVRF Flags

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 while no empty stages in the receive FIFO, the RSPIA detects an overrun error, and sets the OVRF flag to 1. The RSPIA does not copy the data in the shift register to the receive buffer. Even if the SPPE bit is 1, parity errors are not detected. In master mode, the RSPIA copies the pointer value to SPCMDm register to the SPSSR.SPECM[2:0] bits.

(2) When SPDR is read, the RSPIA outputs the data in the receive buffer. At this time the SPRF flag becomes 0 at the last access when the received data is read from the SPDR register in one processing routine using DTC/DMAC. Even if the receive buffer becomes empty, the OVRF flag does not become 0.

(3) If the serial transfer ends with the OVRF flag being 1 (an overrun error occurs), the RSPIA does not copy the data in the shift register to the receive buffer (the SPRF flag remains 0). A receive buffer full interrupt is not generated. Even if the SPPE bit is 1, parity errors are not detected. When in master mode, the RSPIA does not update the SPSSR.SPECM[2:0] bits. When in an overrun error state and the RSPIA does not copy the received data from the shift register to the receive buffer, upon termination of the serial transfer, the RSPIA determines that the shift register is empty; in this manner, data transfer from the transmit buffer to the shift register is enabled.

(4) When 1 is written to the SPSCLR.OVRFC bit, the OVRF flag is set to 0.

The occurrence of an overrun can be checked either by reading the SPSR register or by using an error interrupt and reading the SPSR register. When executing a serial transfer without using an error interrupt, measures should be taken to ensure the early detection of overrun errors, such as reading the SPSR register immediately after SPDR is read. When the RSPIA is used in master mode, the pointer value to the SPCMDm register at the occurrence of the error can be checked by reading the SPSSR.SPECM[2:0] bits. If an overrun error occurs and the OVRF flag is set to 1, normal reception operations cannot be performed until the OVRF flag is set to 0.

When the RSPCK auto-stop function is enabled in master mode, an overrun error does not occur. Figure 38.48 and Figure 38.49 show the clock stop waveform when a serial transfer continues while no empty stages in the receive FIFO in master mode.

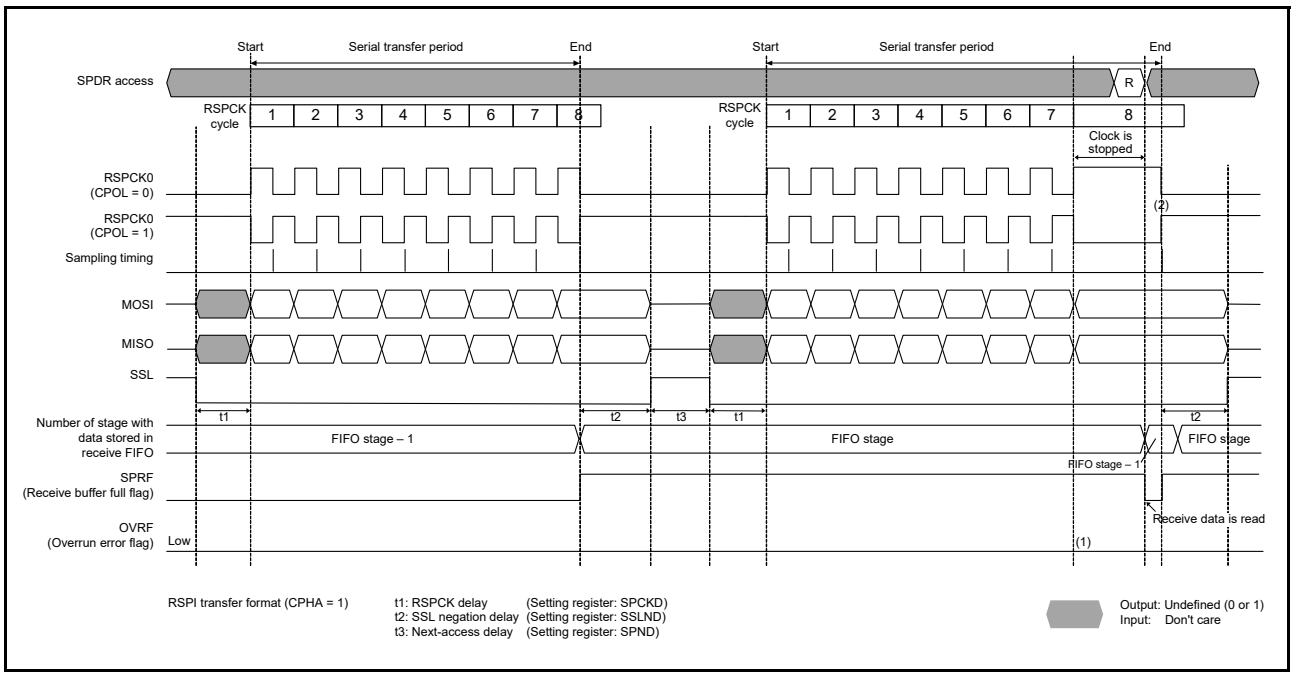


Figure 38.48 Clock Stop Waveform When a Serial Transfer Continues While No Empty Stages in the Receive FIFO in Master Mode (CPHA = 1)

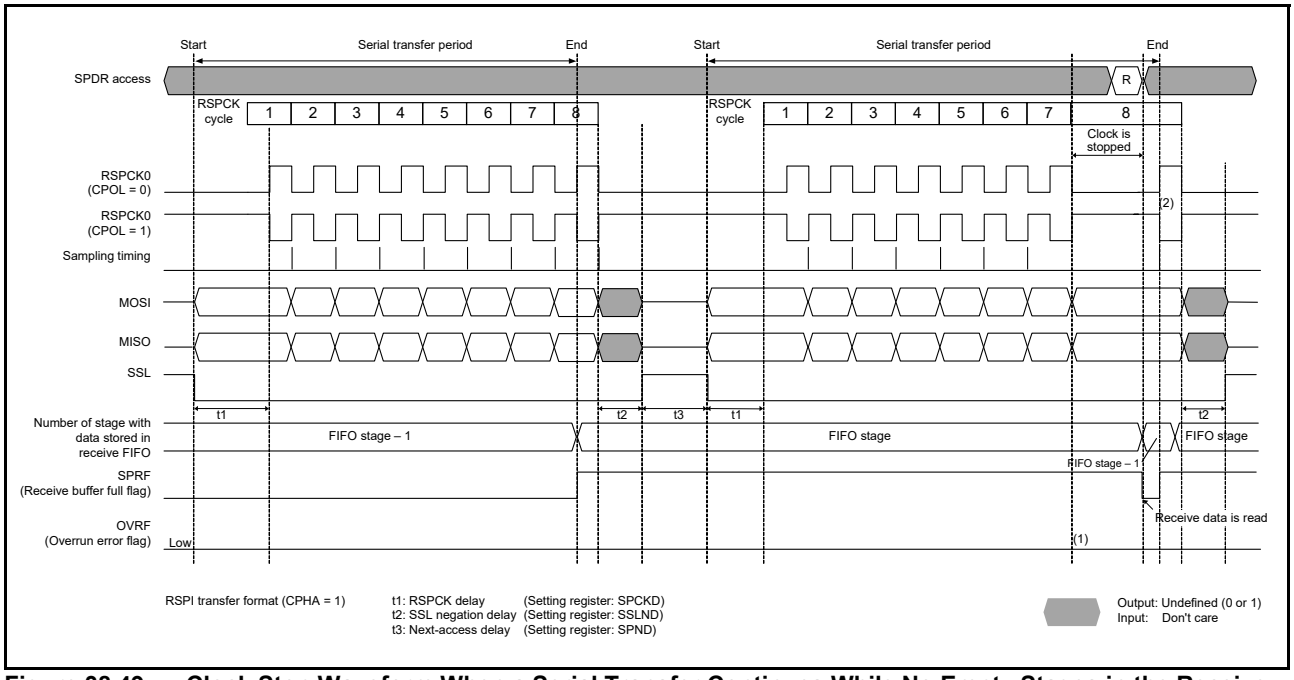


Figure 38.49 Clock Stop Waveform When a Serial Transfer Continues While No Empty Stages in the Receive FIFO in Master Mode (CPHA = 0)

The operation of the flags at the timings shown in steps (1) and (2) in the figure is described below.

(1) When no empty stages in the receive FIFO, an overrun error does not occur because the RSPCK0 is stopped.

(2) If SPDR is read while the clock is stopped, data in the receive buffer can be read. The RSPCK0 restarts after reading the receive buffer.

When the RSPCK auto-stop function is enabled for transfer with no delay of between frames during burst transfer in master mode, an overrun error does not occur. Figure 38.50 and Figure 38.51 show the clock stop waveform when there is no delay between frames during burst transfer and a serial transfer continues while no empty stages in the receive FIFO in master mode.

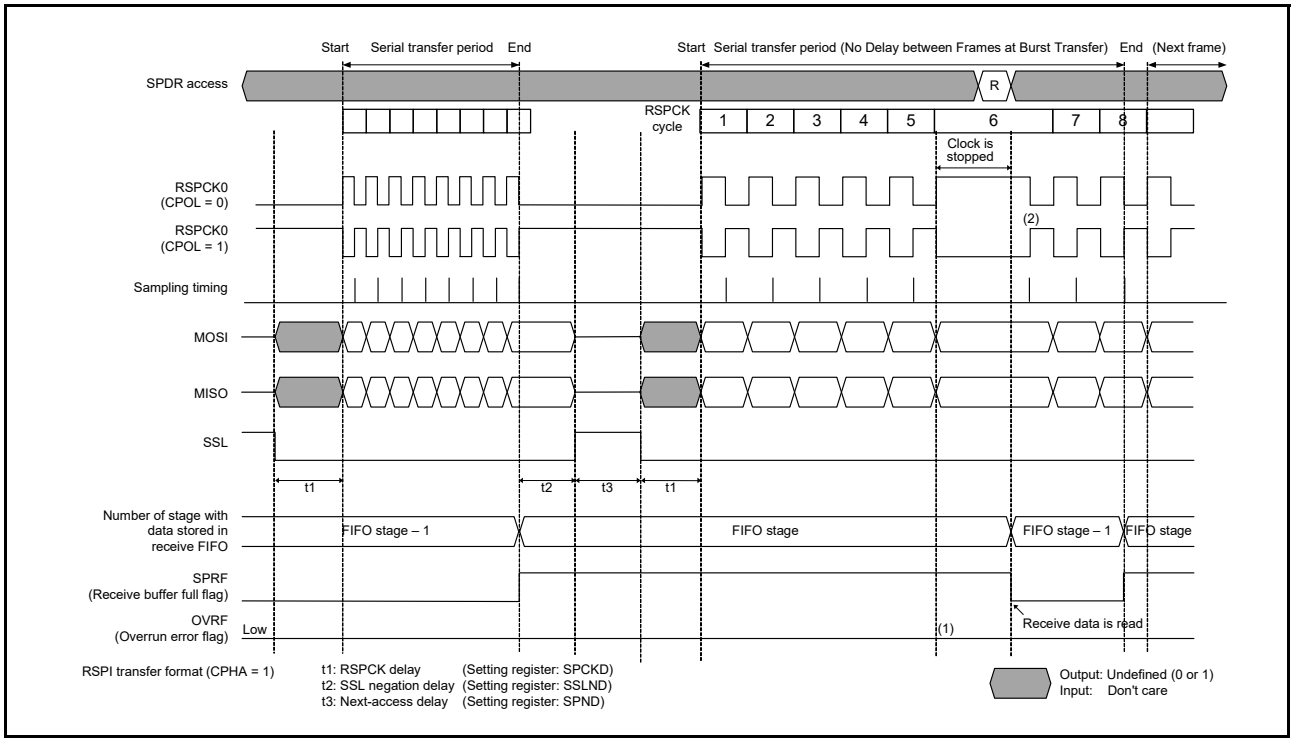


Figure 38.50 Clock Stop Waveform When a Serial Transfer Continues While No Empty Stages in the Receive FIFO in Master Mode (No Delay between Frames at Burst Transfer, CPHA = 1)

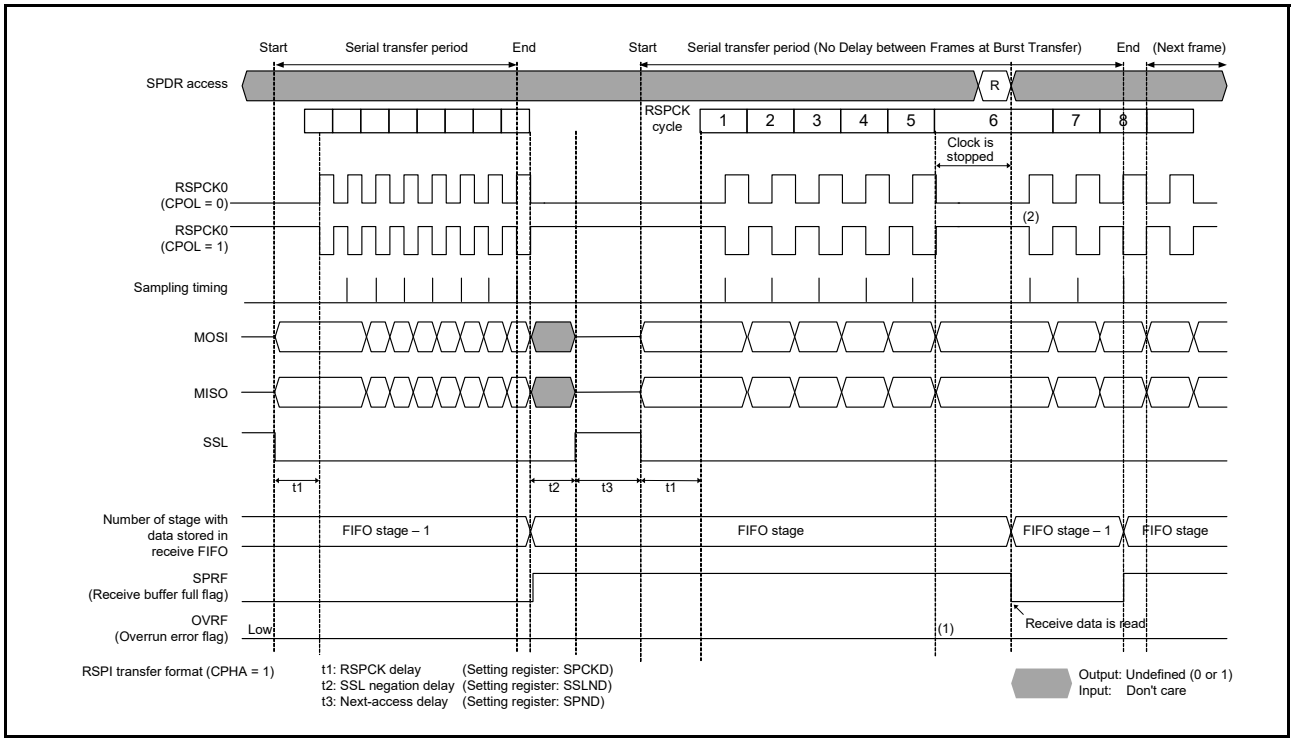


Figure 38.51 Clock Stop Waveform When a Serial Transfer Continues While No Empty Stages in the Receive FIFO in Master Mode (No Delay between Frames at Burst Transfer, CPHA = 0)

The operation of the flags at the timings shown in steps (1) and (2) in the figure is described below.

(1) When no empty stages in the receive FIFO, an overrun error does not occur because the RSPCK0 is stopped.

(2) If SPDR is read while the clock is stopped, data in the receive buffer can be read. The RSPCK0 restarts after reading the receive buffer.

38.3.10.2 Parity Error

If the SPCR.SPPE bit set to 1, when serial transfer in transmit-receive mode or receive-only mode terminates, the RSPIA checks whether there are parity errors. Upon detecting a parity error in the received data, the RSPIA sets the SPSR.PERF flag to 1. Since the RSPIA does not copy the data in the shift register to the receive buffer when the SPSR.OVRF flag is set to 1, parity error detection is not performed for the received data. To set the PERF flag to 0, issue a system reset or write 1 to the SPSCLR.PERFC bit.

Figure 38.52 shows an example of operation of the OVRF and PERF flags. The SPSR access shown in Figure 38.52 indicates the condition of access to SPSR register, where W denotes a write cycle, and R a read cycle. In the example of Figure 38.52, serial communications is performed while the SPCR.SPPE bit is 1. The RSPIA performs an 8-bit serial transfer in which the SPCMDm.CPHA bit is 1 and the SPCMDm.CPOL bit is 0. The numbers given under the RSPCK0 waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

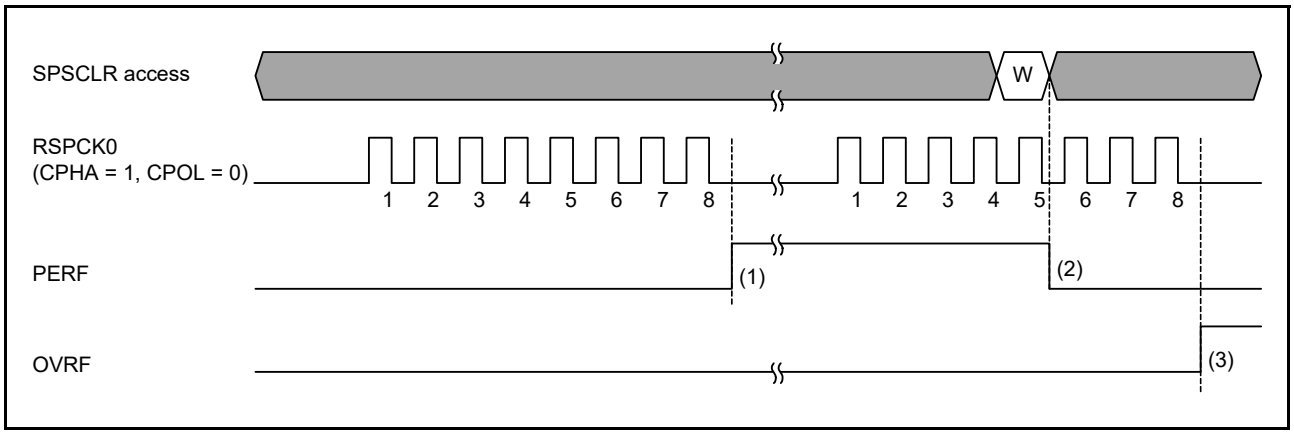


Figure 38.52 Operation Example of PERF Flag

The operation of the flags at the timing shown in steps (1) to (3) in the figure is described below.

(1) If a serial transfer terminates with the RSPIA not detecting an overrun error, the RSPIA copies the data in the shift register to the receive buffer. The RSPIA judges the received data at this timing, and sets the PERF flag to 1 if a parity error is detected. In master mode, the RSPIA copies the pointer value to SPCMDm register to the SPSSR.SPECM[2:0] bits.

(2) When 1 is written to the SPSCLR.PERFC bit, the PERF flag is set to 0.

(3) When the RSPIA detects an overrun error and serial transfer is terminated, the data in the shift register is not copied to the receive buffer. The RSPIA does not perform parity error detection at this timing.

The occurrence of a parity error can be checked either by reading the SPSR register or by using an error interrupt and reading the SPSR register. To use an error interrupt, set the SPCR.SPEIE bit to 1. When executing a serial transfer without using an error interrupt, measures should be taken to ensure the early detection of parity errors, such as reading the SPSR register. When the RSPIA is used in master mode, the pointer value to SPCMDm register at the occurrence of the error can be checked by reading the SPSSR.SPECM[2:0] bits.

38.3.10.3 Mode Fault Error

The RSPIA operates in multi-master mode when the SPCR.MSTR bit is 1, the SPCR.SPMS bit is 0, and the SPCR.MODFEN bit is 1. If the active level is input with respect to the SSL00 input signal of the RSPIA in multi-master mode, the RSPIA detects a mode fault error irrespective of the status of the serial transfer, and sets the SPSR.MODF flag to 1. Upon detecting the mode fault error, the RSPIA copies the value of the pointer to SPCMDm to the SPSSR.SPECM[2:0] bits. The active level of the SSL00 signal is determined by the SSLP.SSL0P bit.
When the MSTR bit is 0, the RSPIA operates in slave mode. The RSPIA detects a mode fault error if the MODFEN bit of the RSPIA in slave mode is 1, and the SPMS bit is 0, and the following conditions are satisfied during the serial transfer period (from the time the driving of valid data is started to the time the final valid data is fetched).

- In the Motorola SPI mode, when the SSL00 input signal is negated during the serial transfer period
- In the TI SSP mode, when the SSL00 input signal is asserted during the serial transfer period
  However, during a burst transfer, no error is detected even if the SSL00 input signal is asserted during the last bit of frame.

Upon detecting a mode fault error, the RSPIA stops driving of the output signals and sets the SPCR.SPE bit to 0. When the SPE bit is cleared, the RSPI function is disabled (refer to section 38.3.12, Initializing RSPIA). In the case of multi-master configuration, detection of a mode fault error is used to stop driving of the output signals and the RSPI function, which allows the master right to be released.
The occurrence of a mode fault error can be checked either by reading the SPSR register or by using an error interrupt and reading the SPSR register. Detecting mode fault errors without utilizing the error interrupt requires polling of the SPSR register. When using the RSPIA in master mode, the pointer value to SPCMDm register at the occurrence of the error can be checked by reading the SPSSR.SPECM[2:0] bits.
When the MODF flag is 1, writing of the value 1 to the SPE bit is ignored by the RSPIA. To enable the RSPI function after the detection of a mode fault error, set the MODF flag to 0.

38.3.10.4 Underrun Error

If RSPIA operates in slave mode (the SPCR.MSTR bit is 0) and the communication mode select bits (CMMD[1:0]) in the RSPI control register (SPCR) is set to 00b or 01b, when serial transfer is started before transmit data output is ready with the SPCR.SPE bit is 1 (RSPI function is enabled), the RSPIA detects an underrun error, and sets the SPSR.MODF and SPSR.UDRF flags to 1.
Upon detecting an underrun error, the RSPIA stops driving of the output signals and sets the SPCR.SPE bit to 0. Clearing of the SPE bit disables the RSPI function. (Refer to section 38.3.12, Initializing RSPIA).
The occurrence of an underrun error can be checked either by reading the SPSR register or by using an error interrupt and reading the SPSR register. Detecting underrun errors without utilizing the error interrupt requires polling of the SPSR register.
When the MODF flag is 1, writing of the value 1 to the SPE bit is ignored by the RSPIA. To enable the RSPI function after the detection of an underrun error, set the MODF flag to 0.

38.3.11 Received Data Ready Detection

When the SPCR.CMMD[1:0] bits are 00b, 01b or 11b, and the value of the SPDRCSR register is not 00h, after receiving data in the receive FIFO during communication (SPE = 1), the SPSR.RRDYF flag is set to 1 when the received data is not stored even after the number of received FIFOs is equal to or less than the threshold value and the value set in the SPDRCSR register has elapsed.

When the receive data ready is detected, the interrupt and event link output can be selected as SPRI or SPEI with the SPCR.RDRIS bit.

Figure 38.53 shows an example of received data ready detection operation.

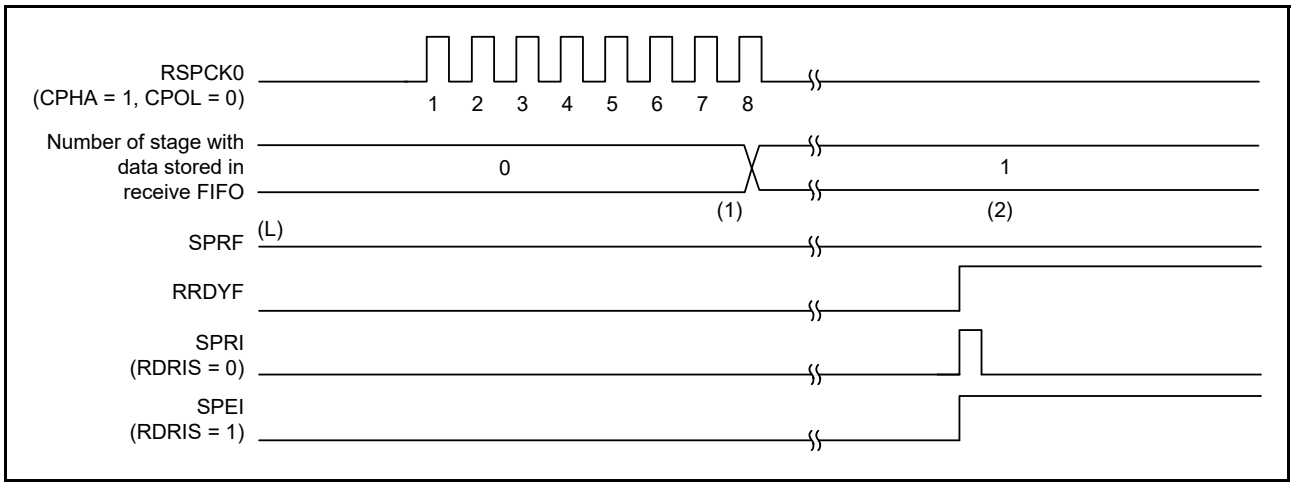


Figure 38.53 Received Data Ready Detection

The operation at the timing shown in steps (1) and (2) in the figure is described below.

(1) Store the received data to the receive FIFO. The SPSR.SPRF flag is 0, because the number of data stored in the receive FIFO is equal to or less than the value of the SPFCR.RTRG[1:0] bits.

(2) Set the RRDYF flag and assert SPRI or SPEI because data is not written to the receive FIFO for the value of the SPDRCSR register from (1).

38.3.12 Initializing RSPIA

If 0 is written to the SPCR.SPE bit or the RSPIA sets the SPE bit to 0 because of the detection of a mode fault error or an underrun error, the RSPIA disables the RSPI function, and initializes some of the module functions. When a system reset is generated, the RSPIA initializes all of the module functions. The following describes initialization by the clearing of the SPCR.SPE bit and initialization by a system reset.

38.3.12.1 Initialization by Clearing the SPE Bit

When the SPCR.SPE bit is set to 0, the RSPIA performs the following initialization:

- Suspending any serial transfer that is being executed
- Stopping the driving of output signals (Hi-Z) in slave mode
- Initializing the internal state of the RSPIA
- Set the SPSR.SPTEF flag to 1

Initialization by the clearing of the SPE bit does not initialize the control bits of the RSPIA. For this reason, the RSPIA can be started in the same transfer mode as prior to the initialization if the SPE bit is set to 1 again.
The SPSR.SPCF, SPRF, OVRF, MODF, PERF, and UDRF flags are not initialized, nor is the value of the RSPI sequence status register (SPSSR) initialized. For this reason, even after the RSPIA is initialized, data from the receive buffer, the communication end status, and the error status during RSPI transfer can be read to check them.
The SPSR.SPTEF flag is initialized to 1. Therefore, if the SPCR.SPTIE bit is set to 1 after RSPIA initialization, a transmit buffer empty interrupt is generated. When the CPU initializes the RSPIA, in order to disable any transmit buffer empty interrupt, 0 should be written to the SPTIE bit simultaneously with the writing of 0 to the SPE bit.

38.3.12.2 System Reset

The initialization by a system reset completely initializes the RSPIA through the initialization of all bits for controlling the RSPIA, initialization of the status bits, and initialization of data registers, in addition to the requirements described in section 38.3.12.1, Initialization by Clearing the SPE Bit.

38.3.13 SPI Operation

38.3.13.1 Master Mode Operation

The only difference between single-master mode operation and multi-master mode operation lies in mode fault error detection (refer to section 38.3.10, Error Detection). When operating in single-master mode, the RSPIA does not detect mode fault errors whereas the RSPIA running in multi-master mode does detect mode fault errors. This section explains operations that are common to single-master mode and multi-master mode.

(1) Starting a Serial Transfer

The RSPIA updates the data in the transmit buffer (SPTXn, n = 0 to 3) when data is written to the RSPI data register (SPDR) with the transmit buffer being empty (data for the next transfer is not set). When the shift register is empty the RSPIA copies data from the transmit buffer to the shift register and starts serial transfer. Upon copying transmit data to the shift register, the RSPIA 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.
For details on the RSPI transfer format, refer to section 38.3.5, Transfer Format (Frame Format). The polarity of the SSL0n output pins depends on the SSLP register settings.

(2) Terminating a Serial Transfer

[Except receive-only operation in master mode]
Irrespective of the SPCMDm.CPHA bit, the RSPIA terminates the serial transfer after transmitting an RSPCK0 edge corresponding to the final sampling timing. If the number of data stored in the receive FIFO is less than the number of FIFO stages, after the termination of serial transfer, the RSPIA copies data from the shift register to the receive buffer of the SPDR register.
It should be noted that the final sampling timing varies depending on the bit length of transmit data. In master mode, the RSPI data length depends on the SPCMDm.SPB[4:0] bits setting. The polarity of the SSL0n output pin depends on the SSLP register settings.
For details on the RSPI transfer format, refer to section 38.3.5, Transfer Format (Frame Format).

[Receive-only operation in master mode]
The RSPIA terminates the serial transfer when any of the following conditions is satisfied.

- After the RSPIA detects the RSPCK0 edge corresponding to the final sampling timing irrespective of the SPCMDm.CPHA bit
- When writing 1 to the SPRMCR.TERM bit during the serial transfer period

If the number of data stored in the receive FIFO is less the number of FIFO stages, after termination of serial transfer, the RSPIA copies data from the shift register to the receive buffer of the SPDR register.
It should be noted that the final sampling timing varies depending on the bit length of transmit data. In master mode, the RSPI data length depends on the SPCMDm.SPB[4:0] bits setting. The polarity of the SSL0n output pin depends on the SSLP register settings.
For details on the RSPI transfer format, refer to section 38.3.5, Transfer Format (Frame Format).

(3) Sequence Control

The transfer format that is employed in master mode is determined by the following registers.

- RSPI sequence control register (SPSCR)
- RSPI command register m (SPCMDm) (m = 0 to 7)
- RSPI bit rate register (SPBR)
- RSPI clock delay register (SPCKD)
- RSPI slave select negation delay register (SSLND)
- RSPI next-access delay register (SPND)

SPSCR is a register used to determine the sequence configuration for serial transfers that are executed by the RSPIA in master mode. The following items are set in the SPCMDm register: SSL0n pin 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. The SPBR register holds some of the bit rate settings; the SPCKD register, an RSPI clock delay value; the SSLND register, an SSL negation delay; and the SPND register, a next-access delay value.

According to the sequence length that is assigned to the SPSCR register, the RSPIA makes up a sequence comprised of a part or all of the SPCMDm register. The RSPIA contains a pointer to the SPCMDm register that makes up the sequence. The value of this pointer can be checked by reading the SPSSR.SPCP[2:0] bits. When the SPCR.SPE bit is set to 1 and the RSPI function is enabled, the RSPIA loads the pointer to the commands in SPCMD0, and incorporates the SPCMD0 settings into the transfer format at the beginning of serial transfer. The RSPIA 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 RSPIA sets the pointer in SPCMD0, and in this manner the sequence is executed repeatedly.

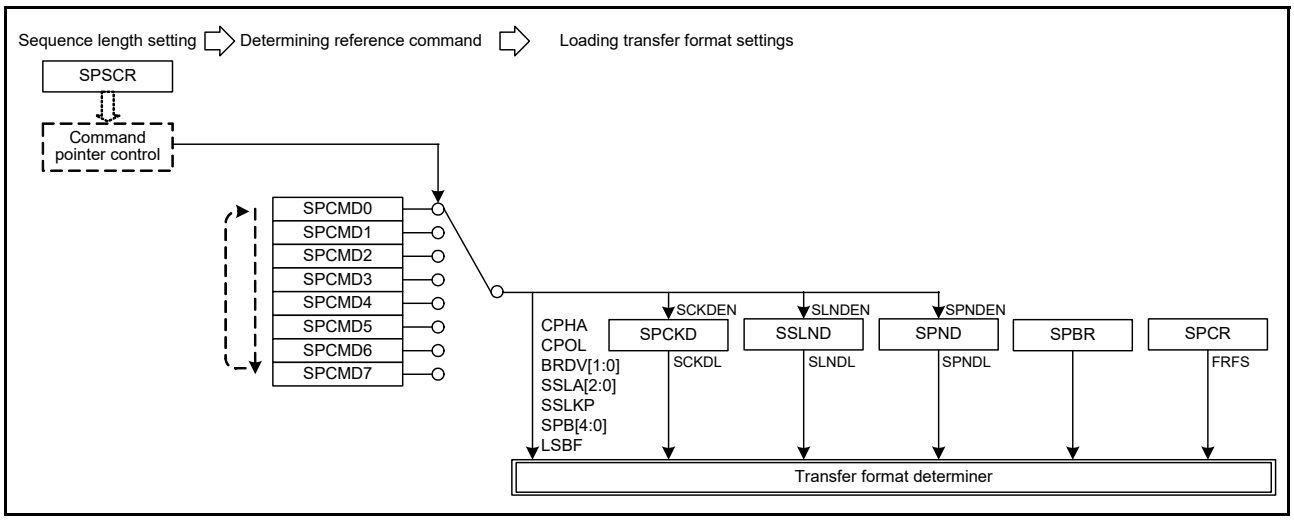


Figure 38.54 Procedure for Determining the Form of Serial Transfer in Master Mode

In this section, a frame is the combination of the data (SPDR) and the settings (SPCMDm).

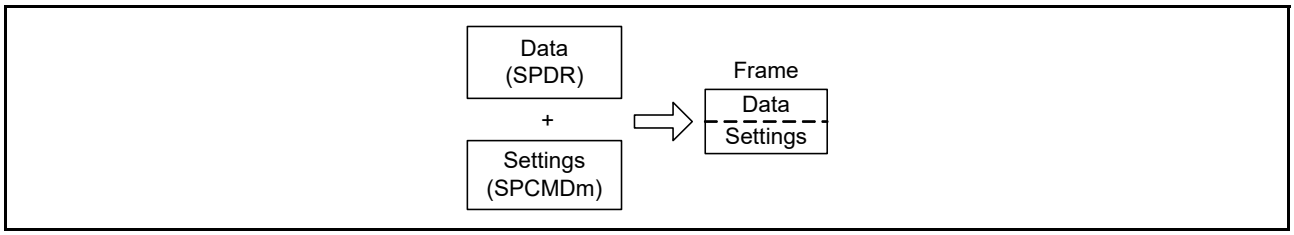


Figure 38.55 Concept of a Frame

Figure 38.56 shows the relationship between the command and the transmit buffer (SPTXn, n = 0 to 3) and receive buffer (SPRXn, n = 0 to 3) in sequence operations.

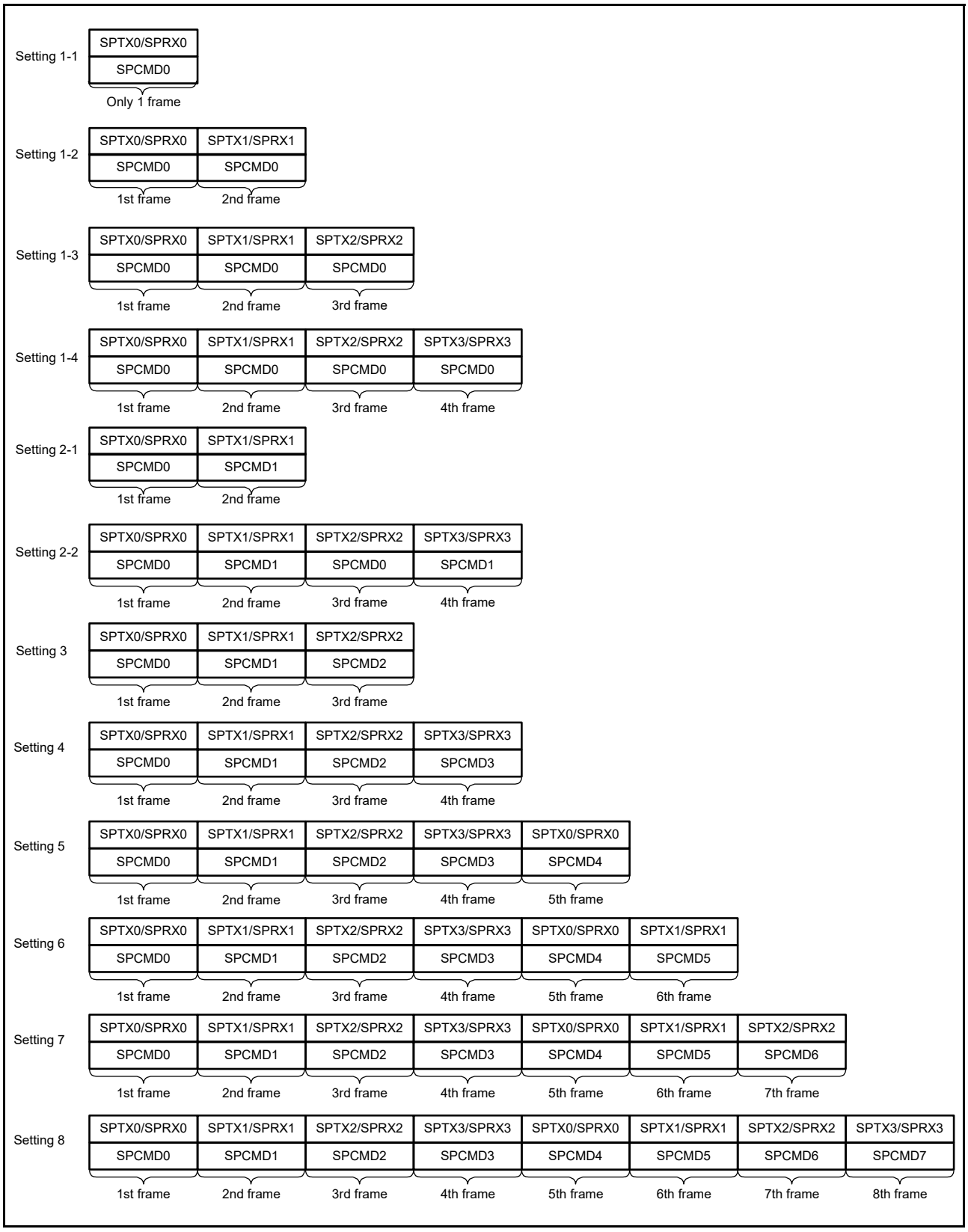


Figure 38.56 Correspondence between the RSPI Command Register m and Transmit/Receive Buffers in Sequence Operations

(4) Burst Transfer

[Motorola SPI]

If the SPCMDm.SSLKP bit that the RSPIA references during the current serial transfer is 1, the RSPIA keeps the SSL0n signal level during the serial transfer until the beginning of the SSL0n signal assertion for the next serial transfer. If the SSL0n signal level for the next serial transfer is the same as the SSL0n signal level for the current serial transfer, the RSPIA can execute continuous serial transfers while keeping the SSL0n signal assertion status (burst transfer).

- When the SPCR.SCKDDIS bit (RSPCK delay between data bytes disable bit) is 0

Figure 38.57 shows an example of an SSL0n signal operation for the case where a burst transfer is implemented using SPCMD0 and SPCMD1 register settings. The text below explains the RSPI operations (1) to (8) as shown in Figure 38.57. It should be noted that the polarity of the SSL0n output signal depends on the SSLP register settings.

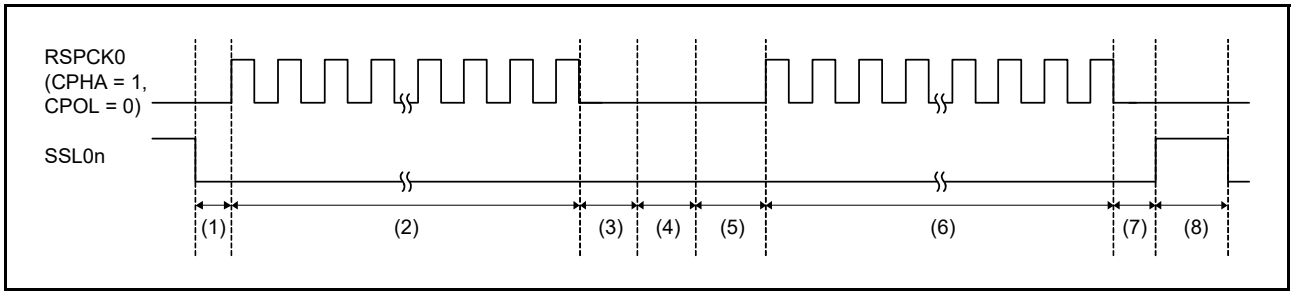


Figure 38.57 Example of Burst Transfer Operation Using SSLKP Bit (SCKDDIS = 0, FRFS = 0)

(1) Based on SPCMD0, the RSPIA asserts the SSL0n signal and inserts RSPCK delays.
(2) The RSPIA executes serial transfers according to SPCMD0.
(3) The RSPIA inserts SSL negation delays.
(4) In transmit-receive mode or transmit-only mode, since the SPCMD0.SSLKP bit is 1, the RSPIA keeps the SSL0n signal value on SPCMD0. This period is sustained, at the shortest, for a period equal to the next-access delay of SPCMD0 and additional 5 PCLKA cycles. If the shift register is empty after the passage of a minimum period, this period is sustained until the transmit data is stored in the shift register for the next transfer.
In receive-only mode, since the SPCMD0.SSLKP bit is 1, the RSPIA keeps the SSL0n signal value on SPCMD0. This period is sustained for a period equal to the next-access delay of SPCMD0 and additional 5 PCLKA cycles.
(5) Based on SPCMD1, the RSPIA asserts the SSL0n signal and inserts RSPCK delays.
(6) The RSPIA executes serial transfers according to SPCMD1.
(7) The RSPIA inserts SSL negation delays.
(8) Since the SPCMD1.SSLKP bit is 0, the RSPIA negates the SSL0n signal. In addition, a next-access delay is inserted according to SPCMD1.

Note: If the SSL0n signal output settings in the SPCMDm register in which 1 is assigned to the SSLKP bit (to use for burst transfer) are different from the SSL0n signal output settings in the SPCMDm register to be used in the next transfer, the RSPIA switches the SSL0n signal status to SSL0n signal assertion ((5) in Figure 38.57) corresponding to the command for the next transfer. Note that if such an SSL0n signal switching occurs, the slaves that drive the MISO0 signal compete, and collision of signal levels may occur.

The RSPIA in master mode references the SSL0n signal operation within the module for the case where the SSLKP bit is not used. Even when the SPCMDm.CPHA bit is 0, the RSPIA can accurately start serial transfers by using the SSL0n signal assertion for the next transfer that is detected internally. For this reason, burst transfer in master mode is enabled regardless of the CPHA bit setting. (Refer to section 38.3.13, SPI Operation.)

- When the SPCR.SCKDDIS bit (RSPCK delay between data bytes disable bit) is 1

Figure 38.58 shows an example of an SSL0n signal operation for the case where a burst transfer is implemented using SPCMD0 and SPCMD1 register settings. The text below explains the RSPI operations (1) to (6) as shown in Figure 38.58. It should be noted that the polarity of the SSL0n output signal depends on the SSLP register settings.

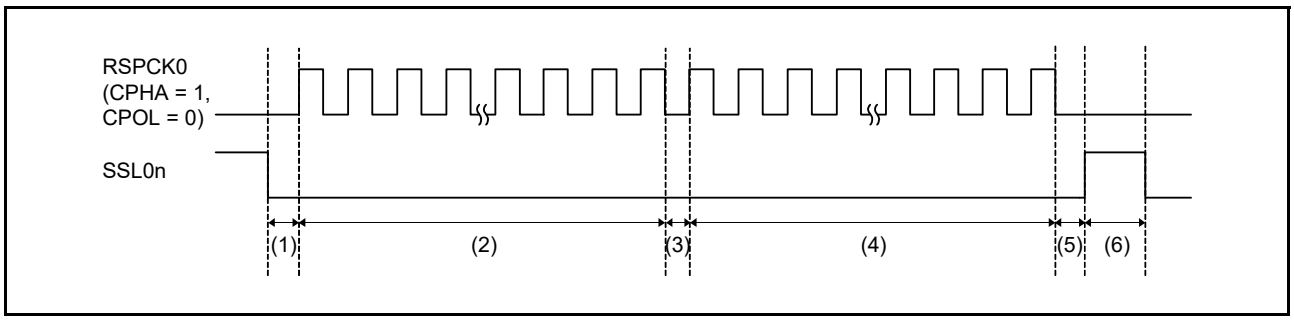


Figure 38.58 Example of Burst Transfer Operation Using SSLKP Bit (SCKDDIS = 1, FRFS = 0)

(1) Based on the SPCMD0 register, the RSPIA asserts the SSL0n signal and inserts RSPCK delays.
The RSPCK delay is inserted only the first frame of burst transfer.

(2) In transmit-receive mode or transmit-only mode, the RSPIA executes serial transfers according to the SPCMD0 register. The RSPIA waits for the last clock output until the next transmit data is stored in the shift register, if the shift register is empty during RSPCK negate period between frames.
In receive-only mode, the RSPIA executes serial transfers according to the SPCMD0 register.

(3) In transmit-receive mode or transmit-only mode, since the SPCMD0.SSLKP bit is 1, the RSPIA keeps the SSL0n signal value on the SPCMD0 register. If the shift register is not empty, RSPCK negate period between frames is 0.5 RSPCK.
In receive-only mode, since it is not the last frame*1, the RSPIA keeps the SSL0n signal value on the SPCMD0 register. RSPCK negate period between frames is 0.5 RSPCK.

(4) The RSPIA executes serial transfers according to the SPCMD1 register.

(5) The RSPIA inserts SSL negation delays.

(6) In transmit-receive mode or transmit-only mode, since the SPCMD1.SSLKP bit is 0, the RSPIA negates the SSL0n signal. In addition, a next-access delay is inserted according to the SPCMD1 register.
In receive-only mode, the RSPIA negates the SSL0n signal. In addition, a next-access delay is inserted according to the SPCMD1 register.

Note 1. In receive-only mode, the last frame is a frame set by the SPRMCR.RFC[4:0] bits when SPRMCR.RFC[4:0] ≠ 00000b or a frame in which SPRMCR.TERM = 1 has been accepted.

[TI SSP]
The RSPIA asserts the SSL0n signal for one cycle at the start of serial transfer.
The RSPIA can execute continuous serial transfers by asserting the SSL0n signal for one cycle at the start of serial transfer (burst transfer).

- When the SPCMDm.SSLKP bit (SSL signal level keeping bit) is 1, and the SPCR.SCKDDIS bit (RSPCK delay between data bytes disable bit) is 1

Figure 38.59 shows an example of an SSL0n signal operation and the serial data MISO0/MOSI0 for the case where a burst transfer is implemented using the SPCMD0 and SPCMD1 register settings. The text below explains the RSPI operations (1) to (6) as shown in Figure 38.59. It should be noted that the polarity of the SSL0n output signal depends on the SSLP register settings.

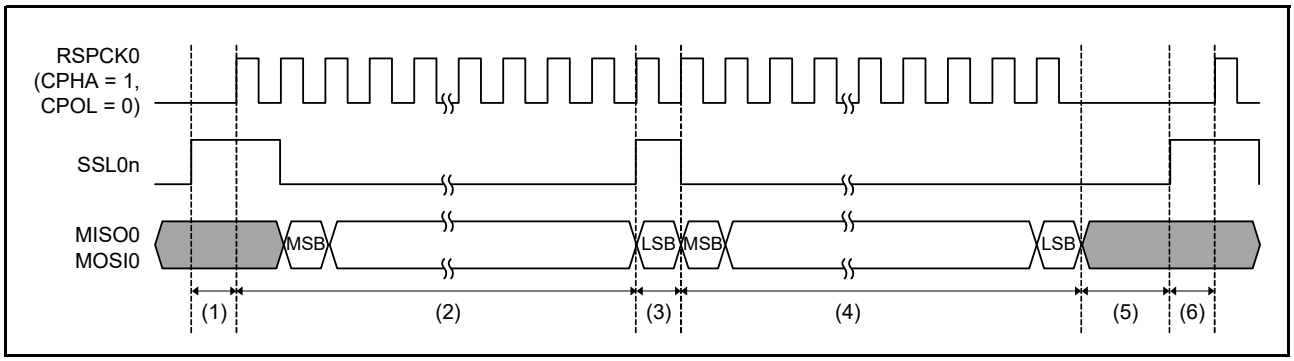


Figure 38.59 Example of Burst Transfer Operation (FRFS = 1)

(1) Based on the SPCMD0 register, the RSPIA asserts the SSL0n signal and inserts RSPCK delays.
The RSPCK delay is inserted only the first frame of burst transfer.
(2) The RSPIA executes serial transfers according to the SPCMD0 register.
(3) Final data transfer and the SSL0n assertion are performed simultaneously.
During transmit-receive/transmit-only operation, the RSPIA waits for the last clock output until the next transmit data is stored in the shift register, if the shift register is empty during RSPCK negate period between frames.
(4) The RSPIA executes serial transfers according to the SPCMD1 register.
(5) The RSPIA inserts SSL0n negation delays at the last frame*1.
(6) A next-access delay is inserted according to the SPCMD1 register.

Note 1. In receive-only mode, the last frame is a frame set by the SPRMCR.RFC[4:0] bits when SPRMCR.RFC[4:0] ≠ 00000b or a frame in which SPRMCR.TERM = 1 has been accepted.
Note: If the SSL0n signal output settings in the SPCMDm register in which 1 is assigned to the SSLKP bit are different from the SSL0n signal output settings in the SPCMDm register to be used in the next transfer, the RSPIA switches the SSL0n signal status to SSL0n signal assertion ((5) in Figure 38.59) corresponding to the command for the next transfer. Note that if such an SSL0n signal switching occurs, the slaves that drive the MISO0 signal compete, and collision of signal levels may occur.

(5) RSPCK Delay (t1)

The RSPCK delay value of the RSPIA in master mode depends on the SPCMDm.SCKDEN bit setting and the SPCKD register setting. The RSPIA determines the SPCMDm register to be referenced during serial transfer by pointer control, and determines an RSPCK delay value during serial transfer by using the SPCMDm.SCKDEN bit and the SPCKD.SCKDL[2:0] bits, as listed in Table 38.9. For a definition of RSPCK delay, refer to section 38.3.5, Transfer Format (Frame Format).

The RSPCK delay is inserted only the first frame of burst transfer when data transfer is performed without delay between frames in burst transfer (SPCMDm.SSLKP = 1, SPCR.SCKDDIS = 1).

Table 38.9 Relationship among SCKDEN Bit, SCKDL[2:0] Bits, and RSPCK Delay Value

SPCMDm.SCKDEN Bit	SPCKD.SCKDL[2:0] Bits	RSPCK Delay Value	
		Motorola SPI	TI SSP
0	000b to 111b	1 RSPCK	0 RSPCK
1	000b	1 RSPCK	1 RSPCK
	001b	2 RSPCK	2 RSPCK
	010b	3 RSPCK	3 RSPCK
	011b	4 RSPCK	4 RSPCK
	100b	5 RSPCK	5 RSPCK
	101b	6 RSPCK	6 RSPCK
	110b	7 RSPCK	7 RSPCK
	111b	8 RSPCK	8 RSPCK

(6) SSL Negation Delay (t2)

The SSL negation delay value of the RSPIA in master mode depends on the SPCMDm.SLNDEN bit setting and the SSLND register setting. The RSPIA determines the SPCMDm register to be referenced during serial transfer by pointer control, and determines an SSL negation delay value during serial transfer by using the SPCMDm.SLNDEN bit and SSLND.SLNDL[2:0] bits, as listed in Table 38.10. For a definition of SSL negation delay, refer to section 38.3.5, Transfer Format (Frame Format).

The SSL negation delay is inserted only the last frame of burst transfer when data transfer is performed without delay between frames in burst transfer (SPCMDm.SSLKP = 1, SPCR.SCKDDIS = 1).

Table 38.10 Relationship among SLNDEN Bit, SLNDL[2:0] Bits, and SSL Negation Delay Value

SPCMDm.SLNDEN Bit	SSLND.SLNDL[2:0] Bits	SSL Negation Delay Value
0	000b to 111b	1 RSPCK
1	000b	1 RSPCK
	001b	2 RSPCK
	010b	3 RSPCK
	011b	4 RSPCK
	100b	5 RSPCK
	101b	6 RSPCK
	110b	7 RSPCK
	111b	8 RSPCK

(7) Next-Access Delay (t3)

The next-access delay value of the RSPIA in master mode depends on the SPCMDm.SPNDEN bit setting and the SPND setting. The RSPIA determines the SPCMDm register to be referenced during serial transfer by pointer control, and determines a next-access delay value during serial transfer by using the SPCMDm.SPNDEN bit and the SPND.SPNDL[2:0] bits, as listed in Table 38.11. For a definition of next-access delay, refer to section 38.3.5, Transfer Format (Frame Format).

The next-access delay is inserted only the last frame of burst transfer when data transfer is performed without delay between frames in burst transfer (SPCMDm.SSLKP = 1, SPCR.SCKDDIS = 1).

Table 38.11 Relationship among SPNDEN Bit, SPNDL[2:0], and Next-Access Delay Value

SPCMDm.SPNDEN Bit	SPND.SPNDL[2:0] Bits	Next-Access Delay Value
0	000b to 111b	1 RSPCK + 5 PCLKA
1	000b	1 RSPCK + 5 PCLKA
	001b	2 RSPCK + 5 PCLKA
	010b	3 RSPCK + 5 PCLKA
	011b	4 RSPCK + 5 PCLKA
	100b	5 RSPCK + 5 PCLKA
	101b	6 RSPCK + 5 PCLKA
	110b	7 RSPCK + 5 PCLKA
	111b	8 RSPCK + 5 PCLKA

(8) Initialization Flowchart

Figure 38.60 is a flowchart illustrating an example of initialization in SPI operation when the RSPIA is used in master mode. For a description of how to set up the interrupt controller, DMAC, and I/O ports, refer to the descriptions given in the individual blocks.

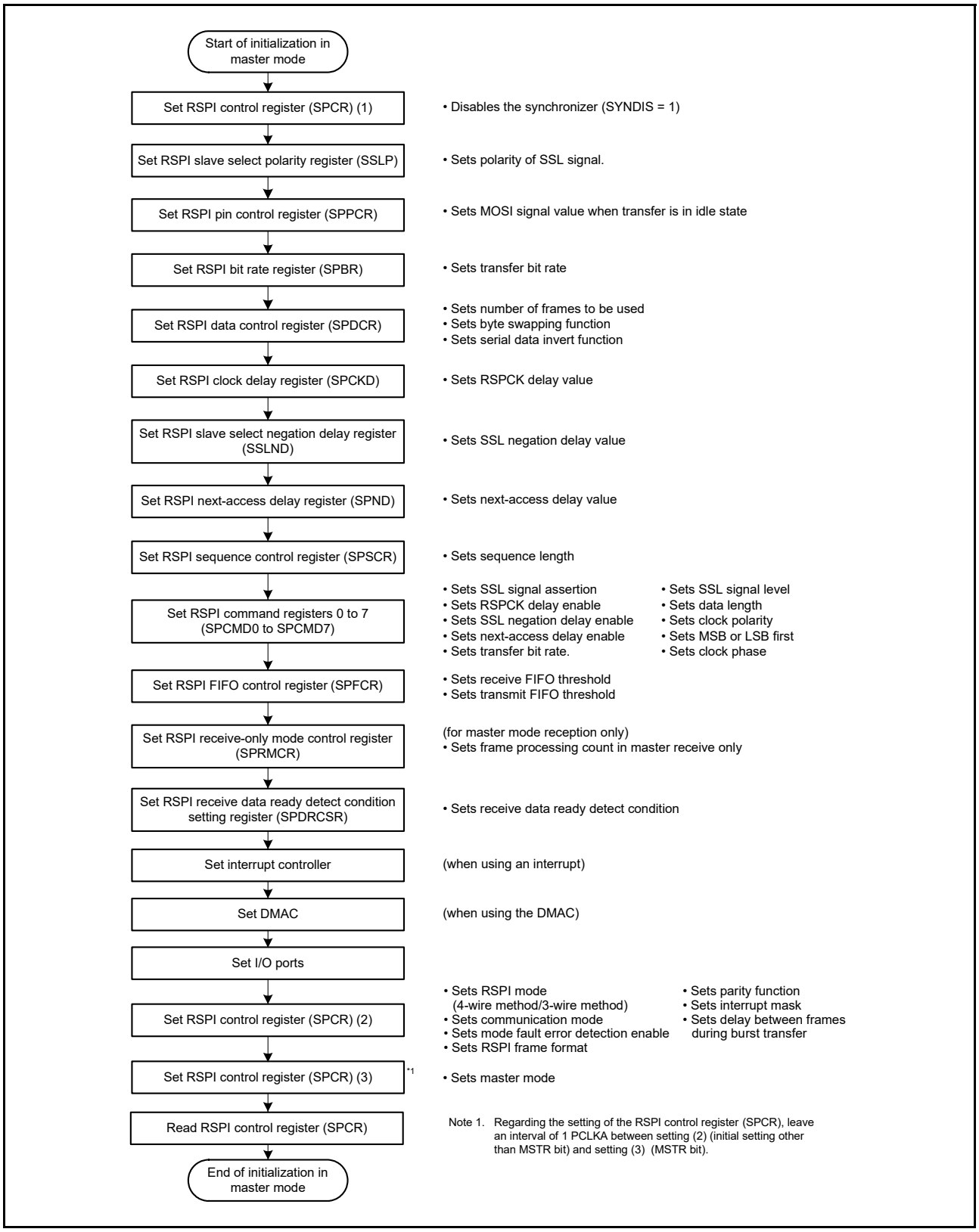


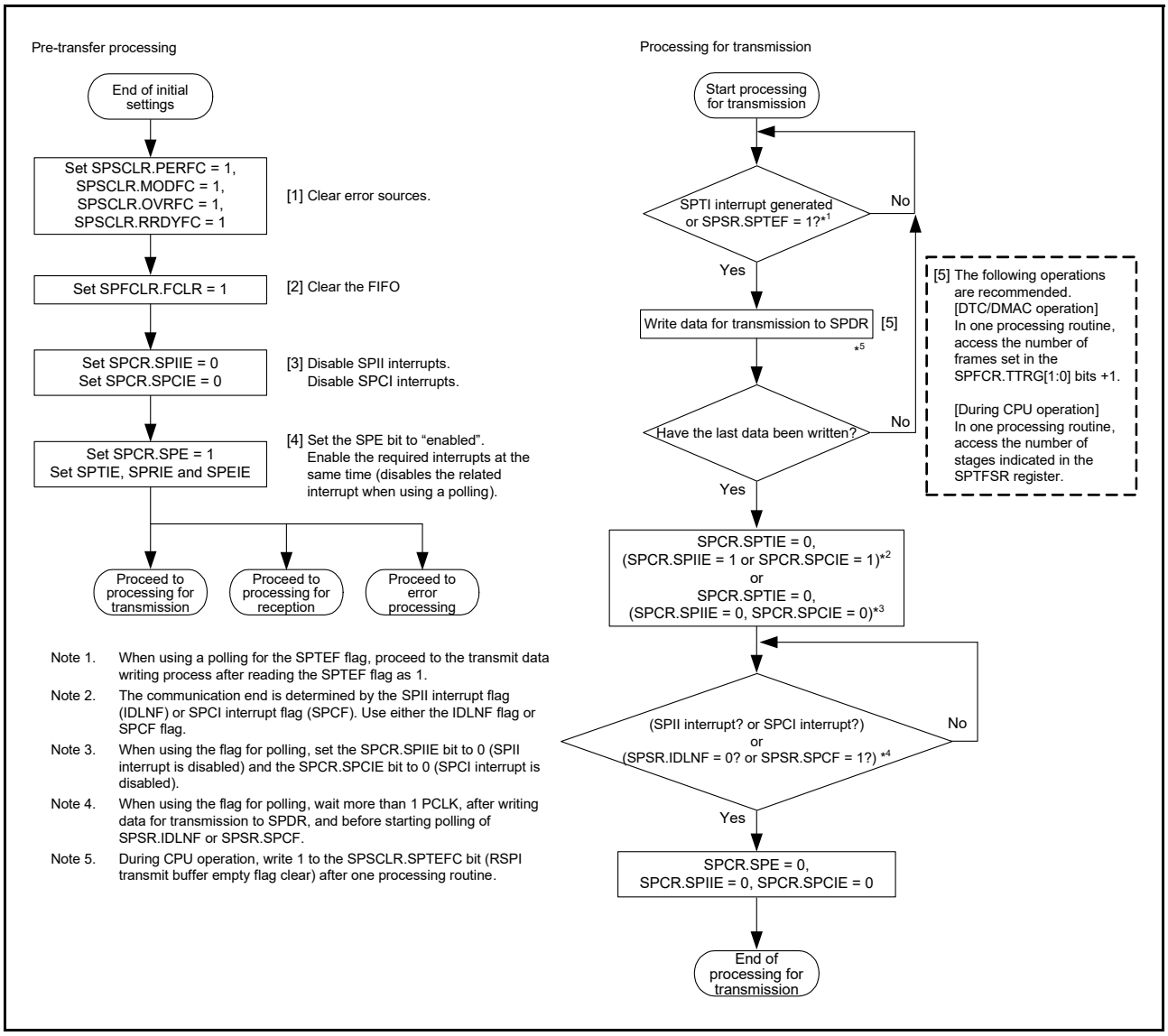
Figure 38.60 Example of Initialization Flowchart in Master Mode (SPI Operation)

(9) Software Processing Flow

Figure 38.61 to Figure 38.63 show examples of the flow of software processing.

(a) Transmit Processing Flow

Enabling SPII or SPCI interrupt after final transmit data is written makes it possible to notify the CPU that transmission of all data has been completed.



Note 1. When using a polling for the SPTEF flag, proceed to the transmit data writing process after reading the SPTEF flag as 1.

Note 2. The communication end is determined by the SPII interrupt flag (IDLNF) or SPCI interrupt flag (SPCF). Use either the IDLNF flag or SPCF flag.

Note 3. When using the flag for polling, set the SPCR.SPIIE bit to 0 (SPII interrupt is disabled) and the SPCR.SPCIE bit to 0 (SPCI interrupt is disabled).

Note 4. When using the flag for polling, wait more than 1 PCLK, after writing data for transmission to SPDR, and before starting polling of SPSR.IDLNF or SPSR.SPCF.

Note 5. During CPU operation, write 1 to the SPSCLR.SPTEFC bit (RSPI transmit buffer empty flag clear) after one processing routine.

Figure 38.61 Flowchart in Master Mode (Transmission)

(b) Receive Processing Flow

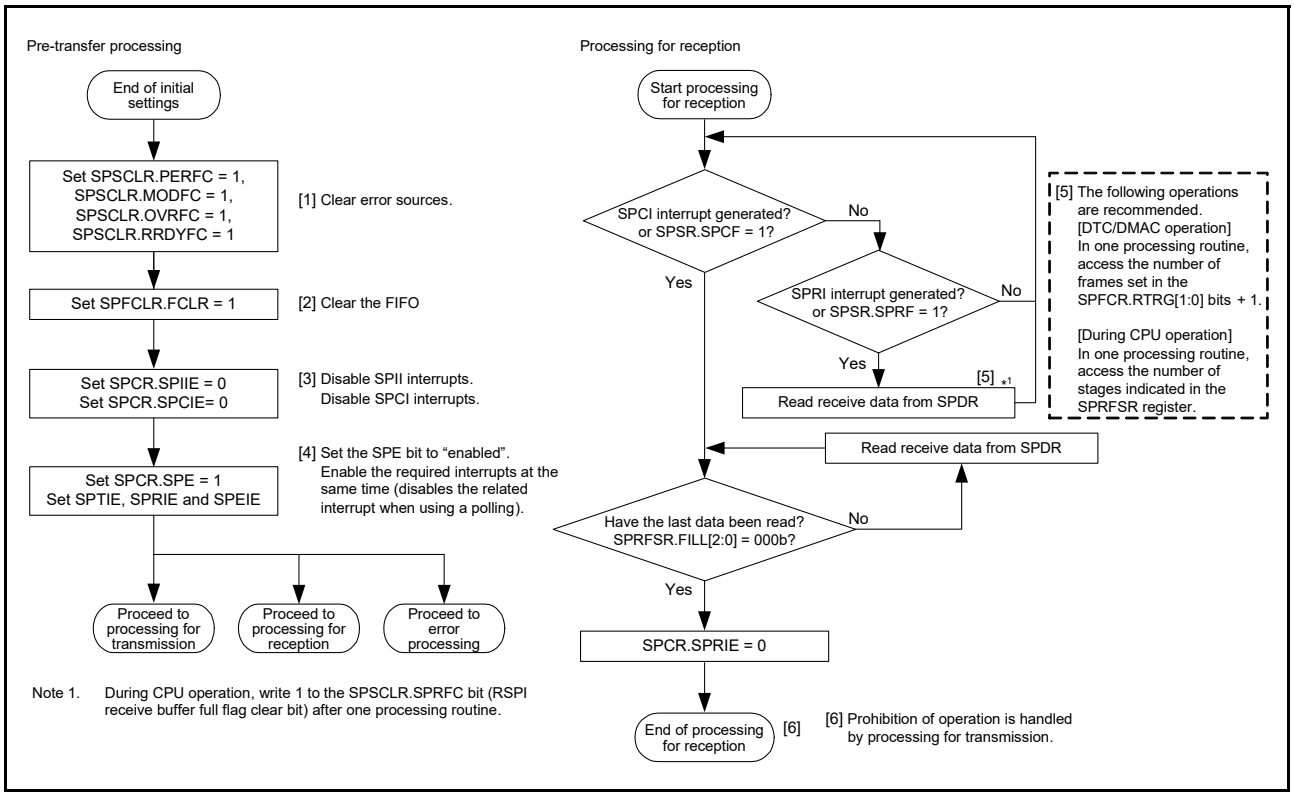


Figure 38.62 Flowchart in Master Mode (Reception)

(c) Receive-Only Processing Flow

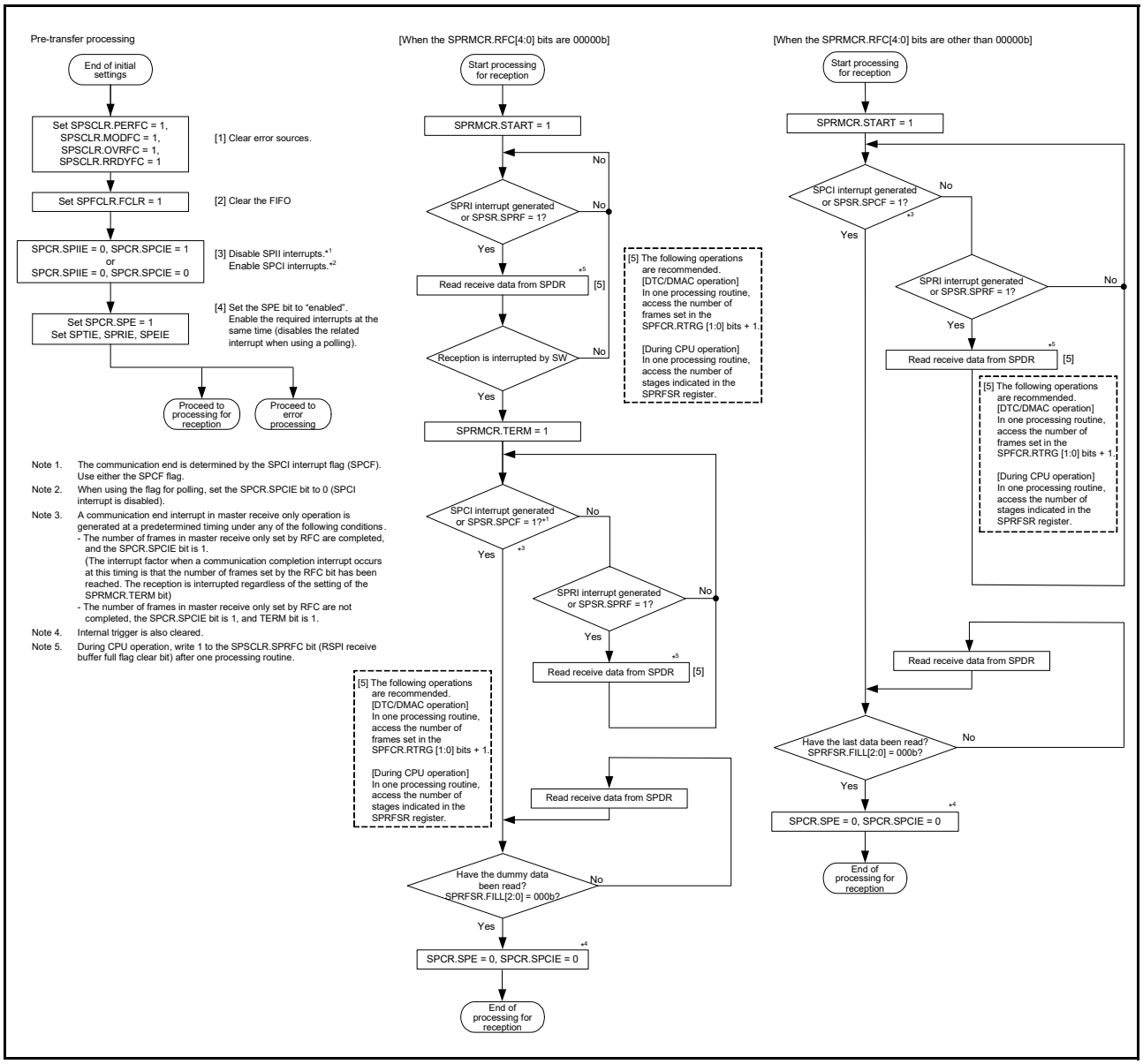


Figure 38.63 Flowchart in Master Mode (Receive-Only)

(d) Flow of Error Processing

When a mode fault error is generated, the SPCR.SPE bit is automatically cleared, stopping operations for transmission and reception. For errors from other sources, however, the SPCR.SPE bit is not cleared and operations for transmission and reception continue; accordingly, we recommend clearing of the SPCR.SPE bit to stop operations in the case of errors other than mode fault errors. Not doing so will lead to updating of the SPSSR.SPECM[2:0] bits.

When interrupts are used and an error occurs, if the ICU.IRn.IR flag for the SPTI or SPRI interrupt request is set to 1, clear the ICU.IRn.IR flag in the error processing routine. If the SPRI interrupt request is indicated, read the receive buffer and initialize the sequencer in the RSPIA.

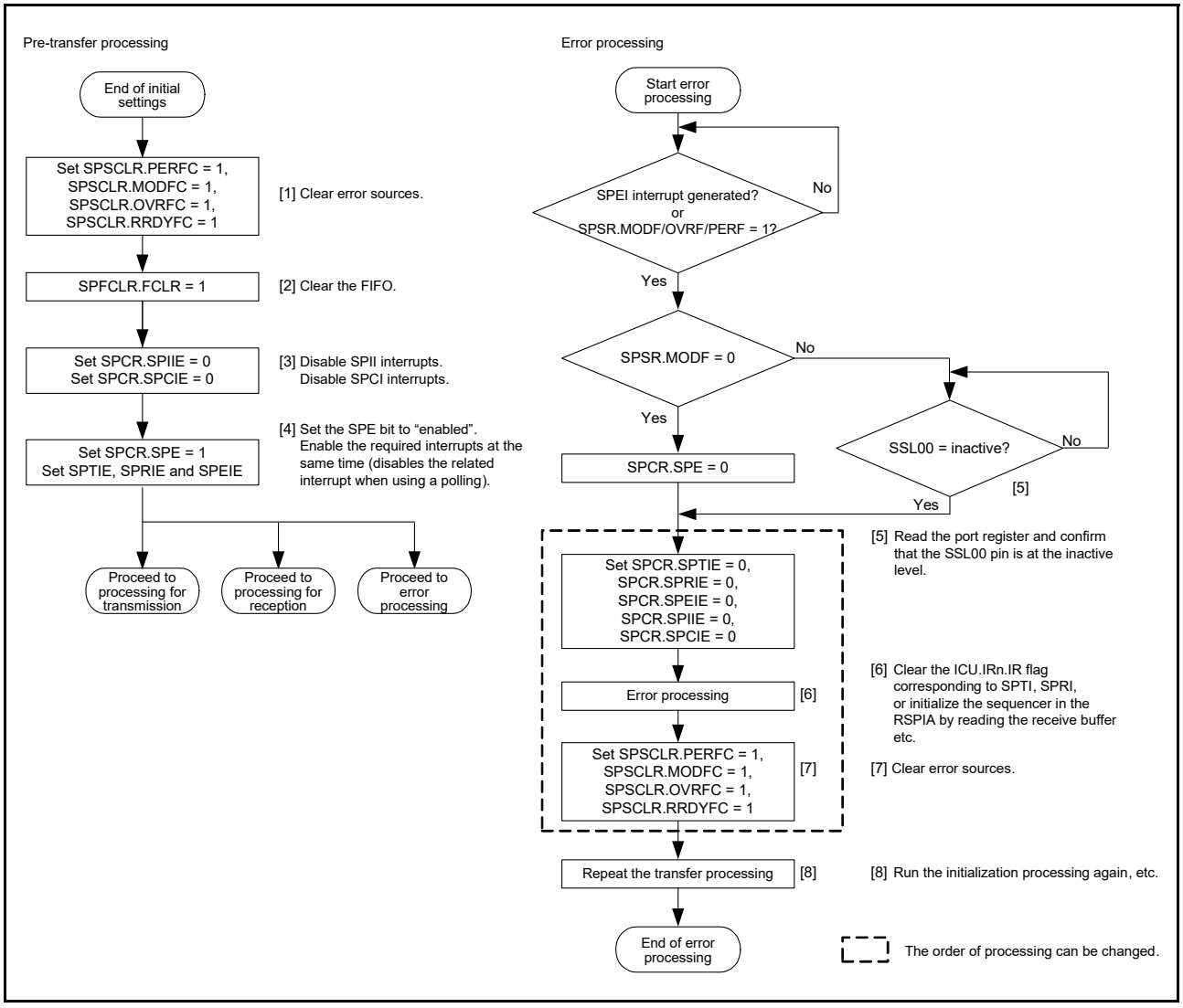


Figure 38.64 Flowchart for Master Mode (Error Processing)

38.3.13.2 Slave Mode Operation

(1) Starting a Serial Transfer

If the SPCMD0.CPHA bit is 0, when detecting an SSL00 input signal assertion, the RSPIA needs to start driving valid data to the MISO0 output signal. For this reason, when the CPHA bit is 0, the assertion of the SSL00 input signal triggers the start of a serial transfer.
If the CPHA bit is 1, when detecting the first RSPCK0 edge in an SSL00 signal asserted condition, the RSPIA needs to start driving valid data to the MISO0 output signal. For this reason, when the CPHA bit is 1, the first RSPCK0 edge in an SSL00 signal asserted condition triggers the start of a serial transfer.
Irrespective of the CPHA bit setting, the timing at which the RSPIA starts driving of the MISO0 output signal is the SSL00 signal assertion timing. The data which is output by the RSPIA is either valid or invalid, depending on the CPHA bit setting.
For details on the RSPI transfer format, refer to section 38.3.5, Transfer Format (Frame Format). The polarity of the SSL00 input signal depends on the setting of the SSLP.SSL0P bit.

(2) Terminating a Serial Transfer

Irrespective of the SPCMD0.CPHA bit, the RSPIA terminates the serial transfer after detecting an RSPCK0 edge corresponding to the final sampling timing. When the number of data stored in the receive FIFO is less than the number of FIFO stages, upon termination of serial transfer the RSPIA copies received data from the shift register to the receive buffer of the SPDR register. Upon termination of a serial transfer the RSPIA changes the status of the shift register to “empty”, regardless of the receive buffer state. A mode fault error occurs if the RSPIA detects an SSL00 input signal negation from the beginning of serial transfer to the end of serial transfer (refer to section 38.3.10, Error Detection).
The final sampling timing changes depending on the bit length of transmit data. In slave mode, the RSPI data length depends on the SPCMD0.SPB[4:0] bit setting. The polarity of the SSL00 input signal depends on the SSLP.SSL0P bit setting. For details on the RSPI transfer format, refer to section 38.3.5, Transfer Format (Frame Format).

(3) Notes on Single-Slave Operation

[Motorola SPI]
If the SPCMD0.CPHA bit is 0, when detecting an SSL00 input signal assertion edge, the RSPIA starts a serial transfer. Upon the SSL00 input signal is fixed to the active level when using the RSPIA in single slave mode in a configuration as shown in Figure 38.7, the RSPIA cannot start a serial transfer correctly while the CPHA bit is 0. To correctly perform transmission and reception of the RSPIA in slave mode in a configuration where the SSL00 input signal is fixed to the active level, set the CPHA bit to 1. If the CPHA bit must be set to 0, do not fix the SSL00 input signal to active level.

[TI SSP]
In the TI SSP mode, upon the SSL00 input signal is fixed to the inactive level when using the RSPIA in single slave mode in a configuration as shown in Figure 38.7, the RSPIA cannot start a serial transfer correctly.
When using the RSPIA in single slave mode, operate in a configuration as shown in Figure 38.6.

(4) Burst Transfer

[Motorola SPI]
When the SPCMD0.CPHA bit is 1, continuous serial transfer (burst transfer) is available while maintaining the SSL00 input signal assertion state. When the CPHA bit is 1, the period (from the first RSPCK0 edge in the SSL00 input signal active state until the sampling timing for receiving the final bit) corresponds to the serial transfer period. Even if the SSL00 input signal is always at the active level, start of an access can be detected, which allows burst transfer.
When the CPHA bit is 0, the second and subsequent serial burst transfers cannot be performed correctly for the same reason as (3).

[TI SSP]
In serial transfer, data transfer starts after the SSL0n input signal is asserted for RSPCK 1 cycle. Since frame transfer starts from the SSL0n input signal, SSL must be asserted between frames.

(5) Initialization Flowchart

Figure 38.65 shows an example of initialization flow when using the RSPIA in slave-mode SPI operation. For how to set the interrupt controller, DMAC, and input/output ports, see descriptions of each block.

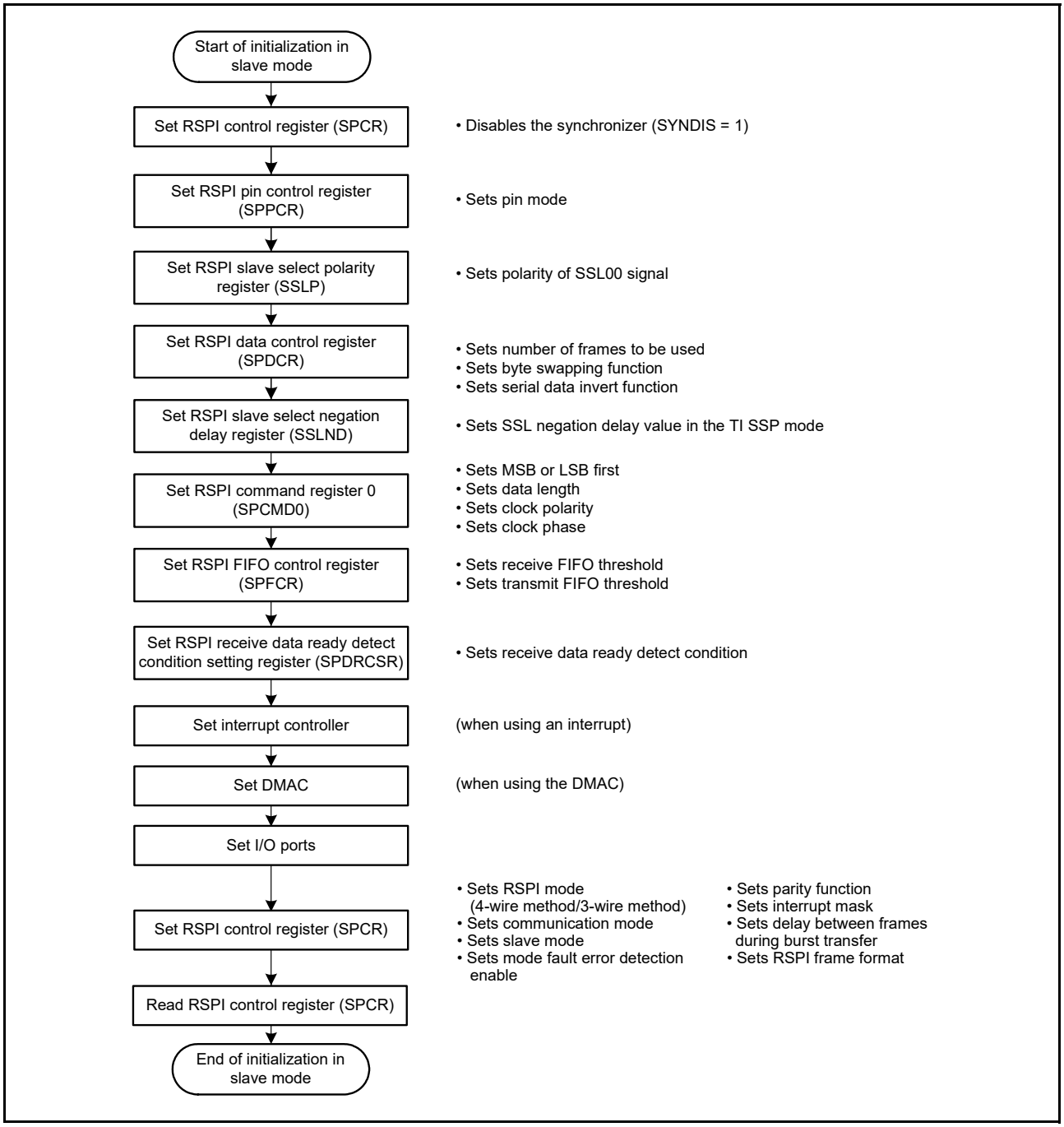
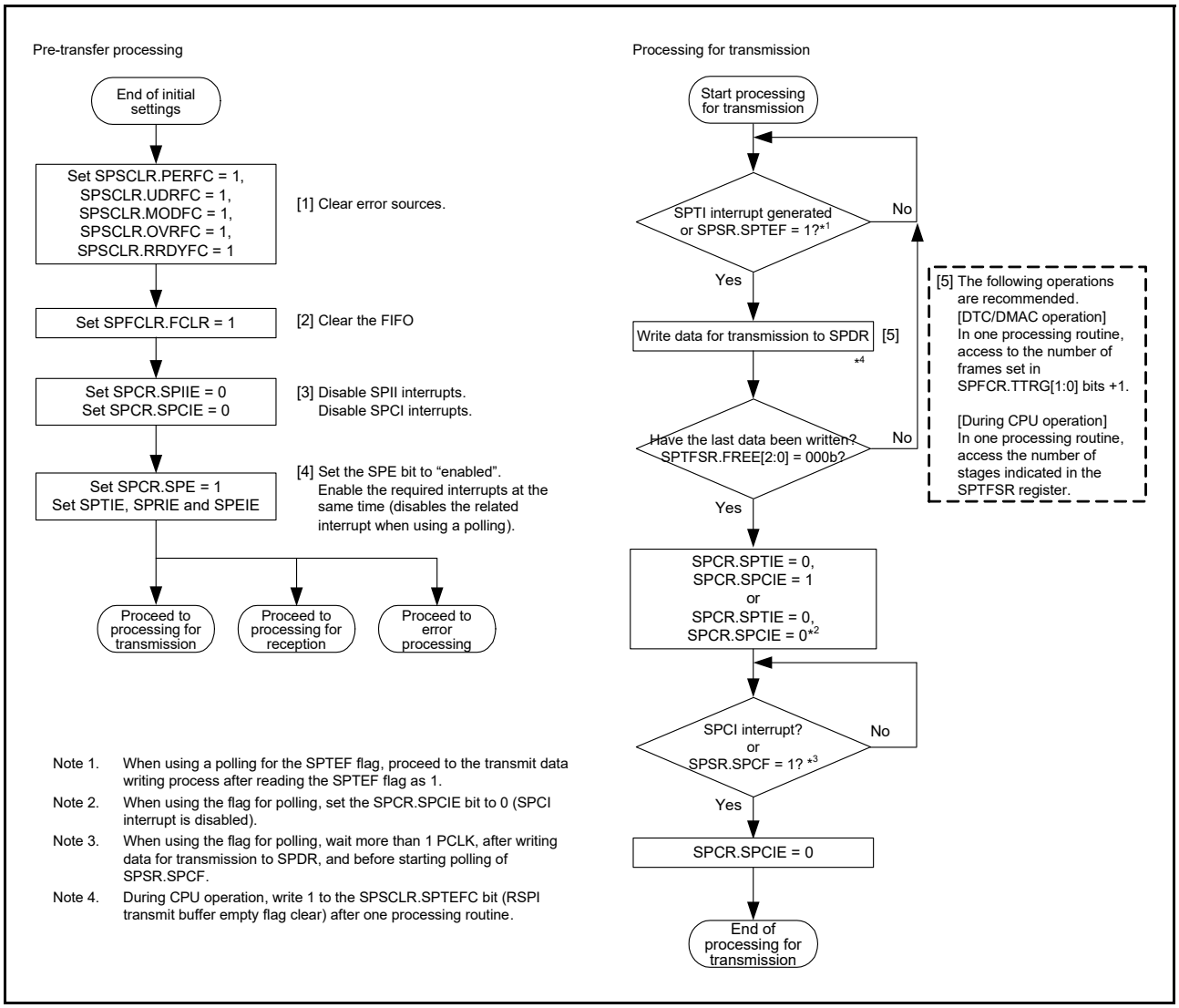


Figure 38.65 Example of Initialization Flow in Slave Mode

(6) Software Processing Flow

Figure 38.66 to Figure 38.69 show examples of the flow of software processing.

(a) Transmit Processing Flow



Note 1. When using a polling for the SPTEF flag, proceed to the transmit data writing process after reading the SPTEF flag as 1.

Note 2. When using the flag for polling, set the SPCR.SPCIE bit to 0 (SPCI interrupt is disabled).

Note 3. When using the flag for polling, wait more than 1 PCLK, after writing data for transmission to SPDR, and before starting polling of SPSR.SPCF.

Note 4. During CPU operation, write 1 to the SPSCLR.SPTEFC bit (RSPI transmit buffer empty flag clear) after one processing routine.

Figure 38.66 Flowchart in Slave Mode (Transmission)

(b) Receive Processing Flow

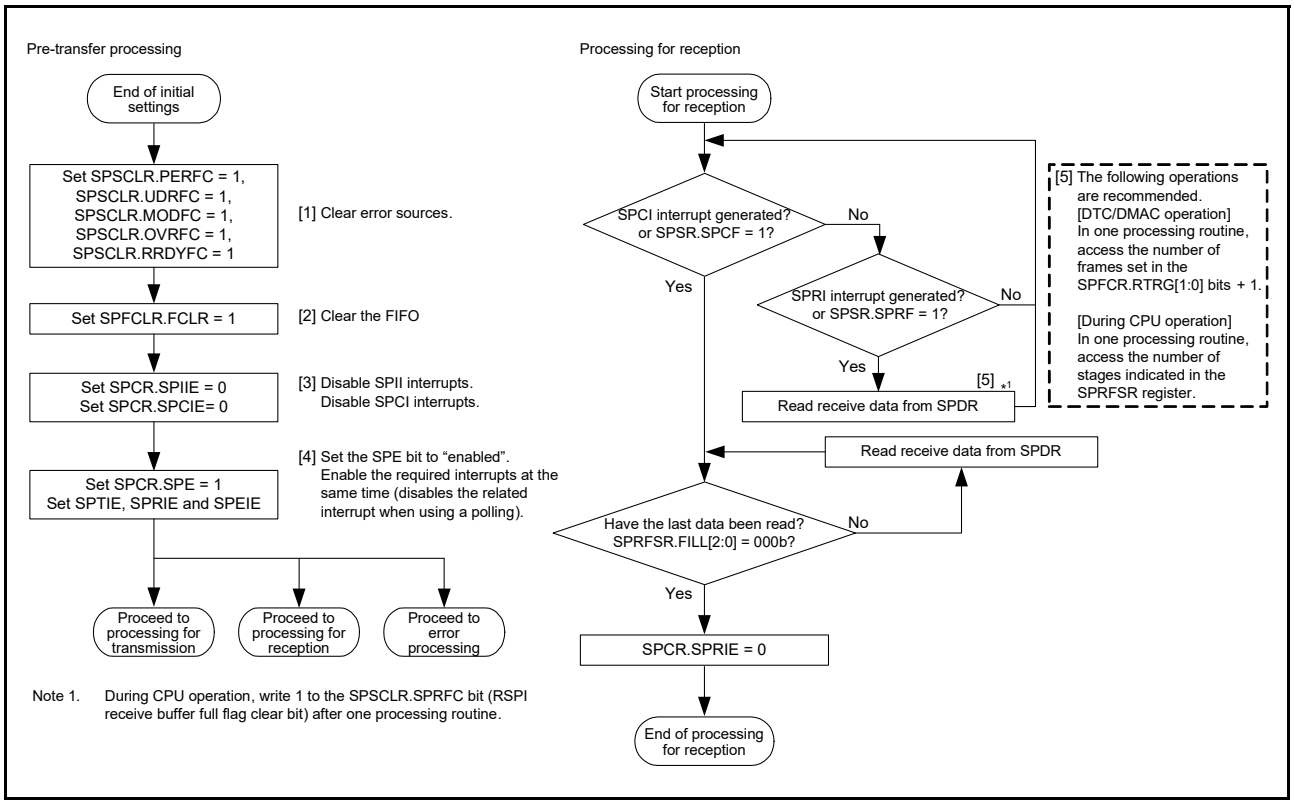


Figure 38.67 Flowchart in Slave Mode (Reception)

(c) Receive-Only Processing Flow

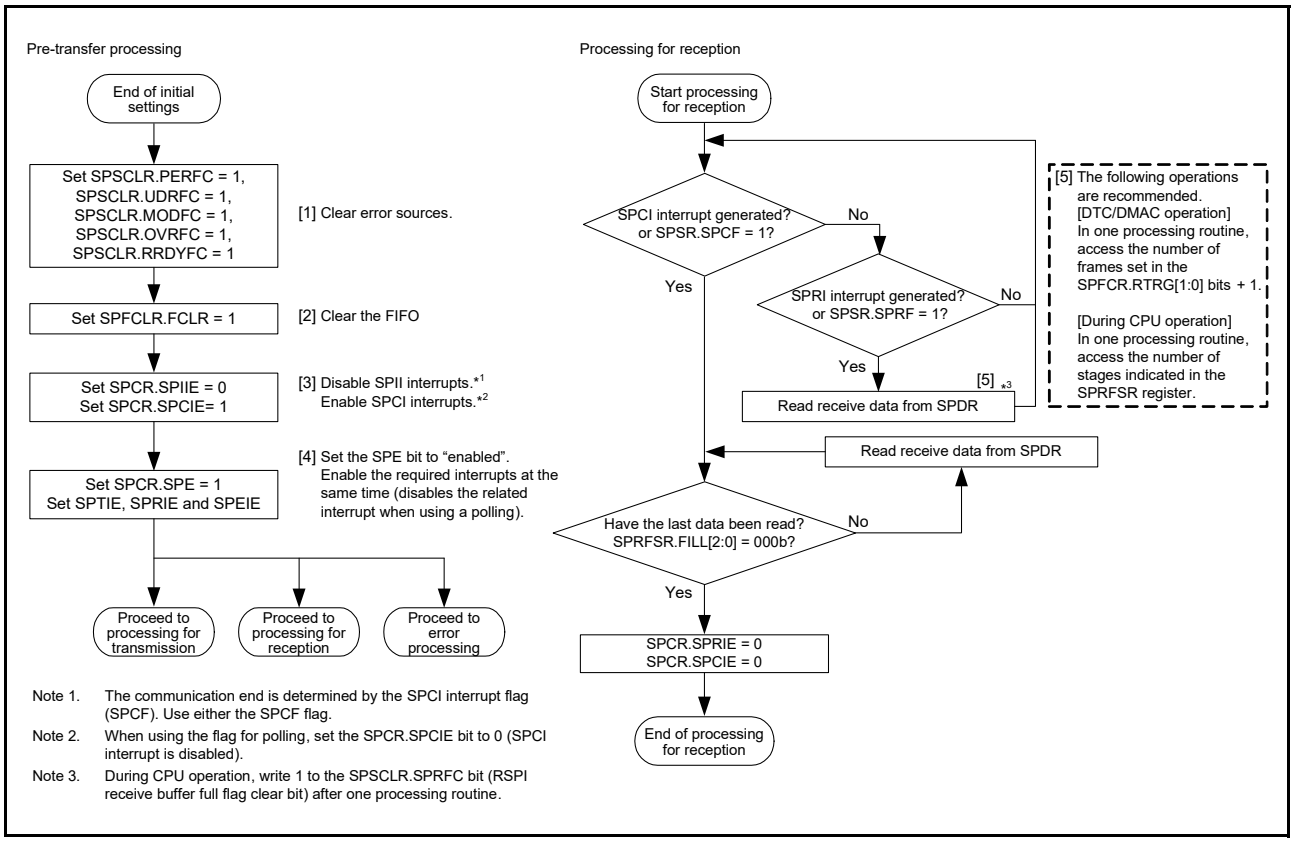


Figure 38.68 Flowchart in Slave Mode (Receive-Only)

(d) Flow of Error Processing

In slave mode operation, the MODF flag can be cleared regardless of the SSL00 pin status even when a mode fault error is generated.

When interrupts are used and an error occurs, if the ICU.IRn.IR flag for the SPTI or SPRI interrupt request is set to 1, clear the ICU.IRn.IR flag in the error processing routine. If the SPRI interrupt request is indicated, read the receive buffer and initialize the sequencer in the RSPIA.

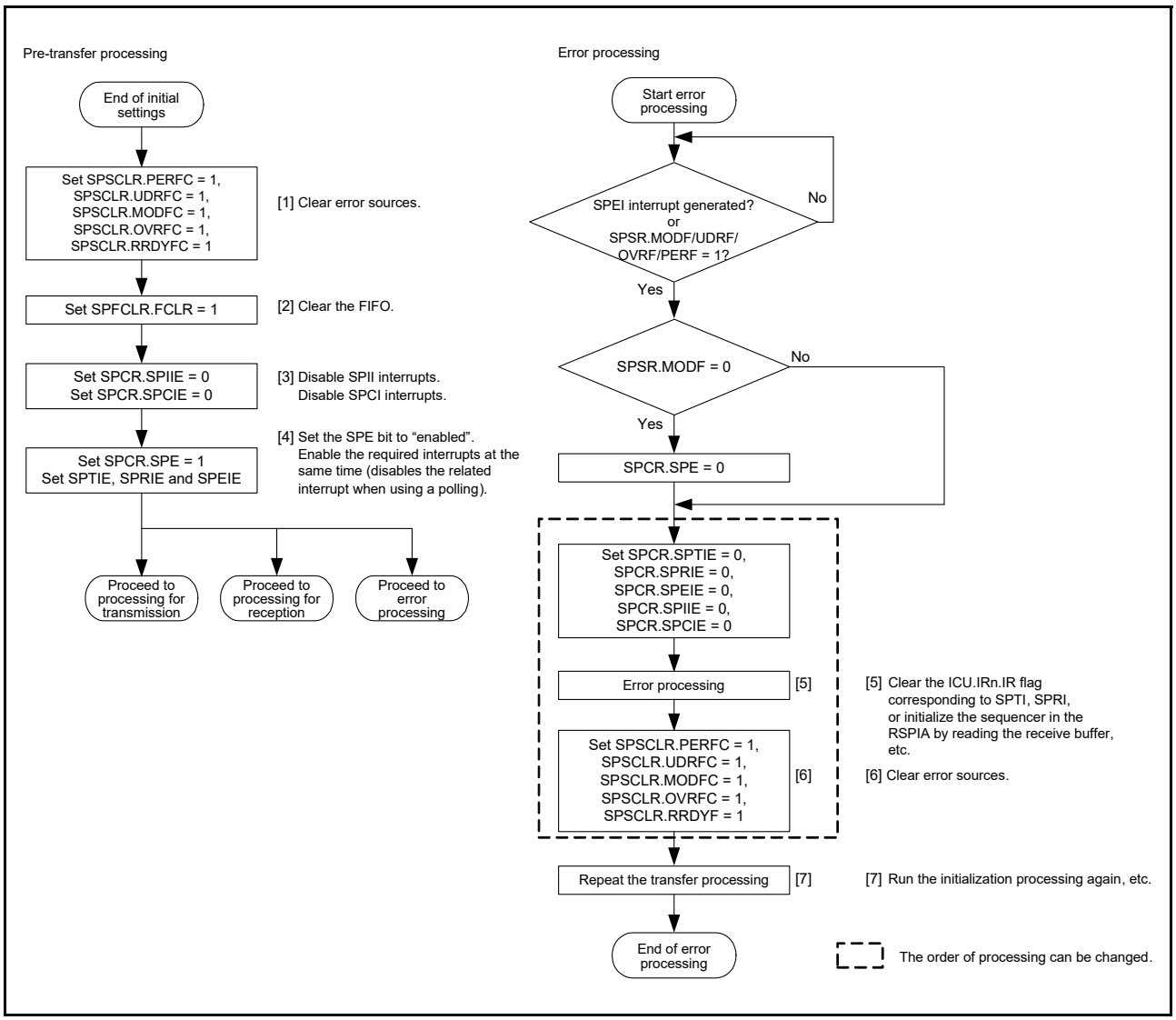


Figure 38.69 Flowchart in Slave Mode (Error Processing)

38.3.14 Clock Synchronous Operation

Setting the SPCR.SPMS bit to 1 selects clock synchronous operation of the RSPIA. In clock synchronous operation, the SSL0n pin is not used, and the three pins of RSPCK0, MOSI0, and MISO0 handle communications. The SSL0n pin is available as I/O port pins.
Although clock synchronous operation does not require use of the SSL0n pin, operation of the module is the same as in SPI operation. That is, in both master and slave mode operations, communications can be performed with the same flow as in SPI operation. However, mode fault errors are not detected because the SSL0n pin is not used.
Furthermore, do not set the SPCMDm.CPHA bit to 0 if clock synchronous operation is to proceed in slave mode (SPCR.MSTR = 0).

38.3.14.1 Master Mode Operation

(1) Starting a Serial Transfer

The RSPIA updates the data in the transmit buffer (SPTXn, n = 0 to 3) of the SPDR register when data is written to the SPDR register with the transmit buffer being empty (data for the next transfer is not set). When the shift register is empty, the RSPIA copies data from the transmit buffer to the shift register and starts serial transmission. Upon copying transmit data to the shift register, the RSPIA 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.
For details on the RSPI transfer format, refer to section 38.3.5, Transfer Format (Frame Format).
However, transfer in clock synchronous operation is conducted without the SSL0n output signal.

(2) Terminating a Serial Transfer

The RSPIA terminates the serial transfer after transmitting an RSPCK0 edge corresponding to the final sampling timing. If the number of data stored in the receive FIFO is less than the number of FIFO stages, upon termination of serial transfer, the RSPIA 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 transmit data. In master mode, the RSPI data length depends on the SPCMDm.SPB[4:0] bit setting. For details on the RSPI transfer format, refer to section 38.3.5, Transfer Format (Frame Format).
However, transfer in clock synchronous operation is conducted without the SSL0n output signal.

(3) Sequence Control

The transfer format employed in master mode is determined by the following registers:

- RSPI sequence control register (SPSCR)
- RSPI command register m (SPCMDm) (m = 0 to 7)
- RSPI bit rate register (SPBR)
- RSPI clock delay register (SPCKD)
- RSPI slave select negation delay register (SSLND)
- RSPI next-access delay register (SPND)

Although the SSL0n signals are not output in clock synchronous operation, these settings are valid.

The SPSCR register is a register used to determine the sequence configuration for serial transfers that are executed by the RSPIA in master mode. The following items are set in the SPCMDm register: MSB/LSB first, data length, some of the bit rate settings, RSPCK polarity/phase, whether the SPCKD register is to be referenced, whether the SSLND register is to be referenced, and whether the SPND register is to be referenced. The SPBR register holds some of the bit rate settings; the SPCKD register, an RSPI clock delay value; the SSLND register, an SSL negation delay; and the SPND register, a next-access delay value.
According to the sequence length that is assigned to the SPSCR register, the RSPIA makes up a sequence comprised of a part or all of the SPCMDm register. The RSPIA contains a pointer to the SPCMDm register that makes up the sequence.

The value of this pointer can be checked by reading the SPSSR.SPCP[2:0] bits. When the SPCR.SPE bit is set to 1 and the RSPI function is enabled, the RSPIA loads the pointer to the commands in the SPCMD0 register, and incorporates the SPCMD0 register setting into the transfer format at the beginning of serial transfer. The RSPIA 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 RSPIA sets the pointer in the SPCMD0 register, and in this manner the sequence is executed repeatedly.

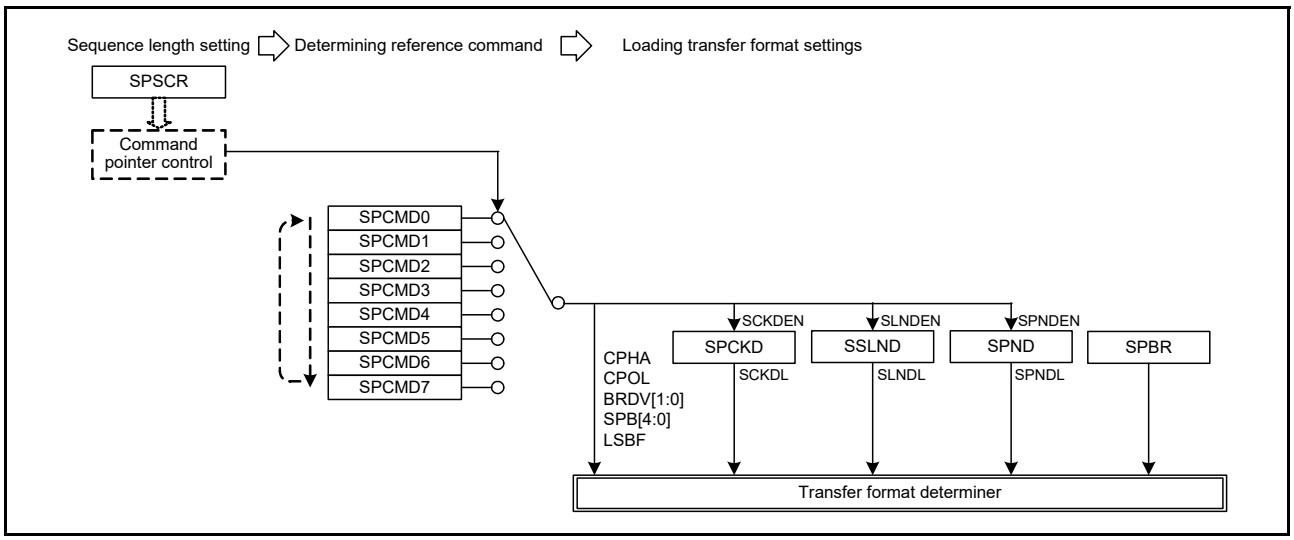


Figure 38.70 Procedure for Determining the Form of Serial Transfer in Master Mode

In this section, a frame is the combination of the data (SPDR) and the settings (SPCMDm).

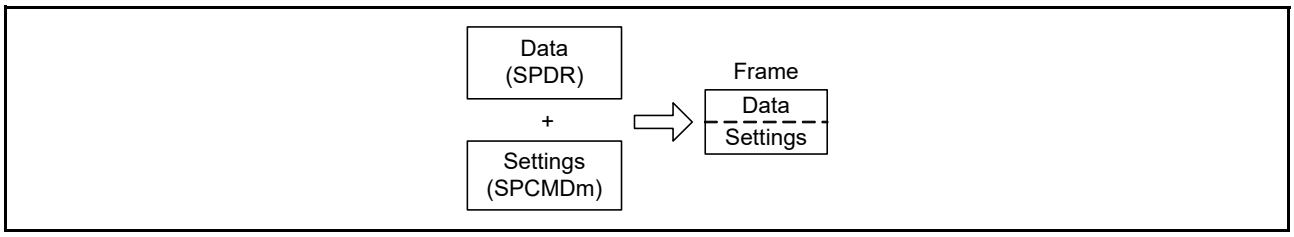


Figure 38.71 Concept of a Frame

Figure 38.72 shows the relationship between the command and the transmit buffer (SPTXn, n = 0 to 3) and receive buffer (SPRXn, n = 0 to 3) in the sequence of operations.

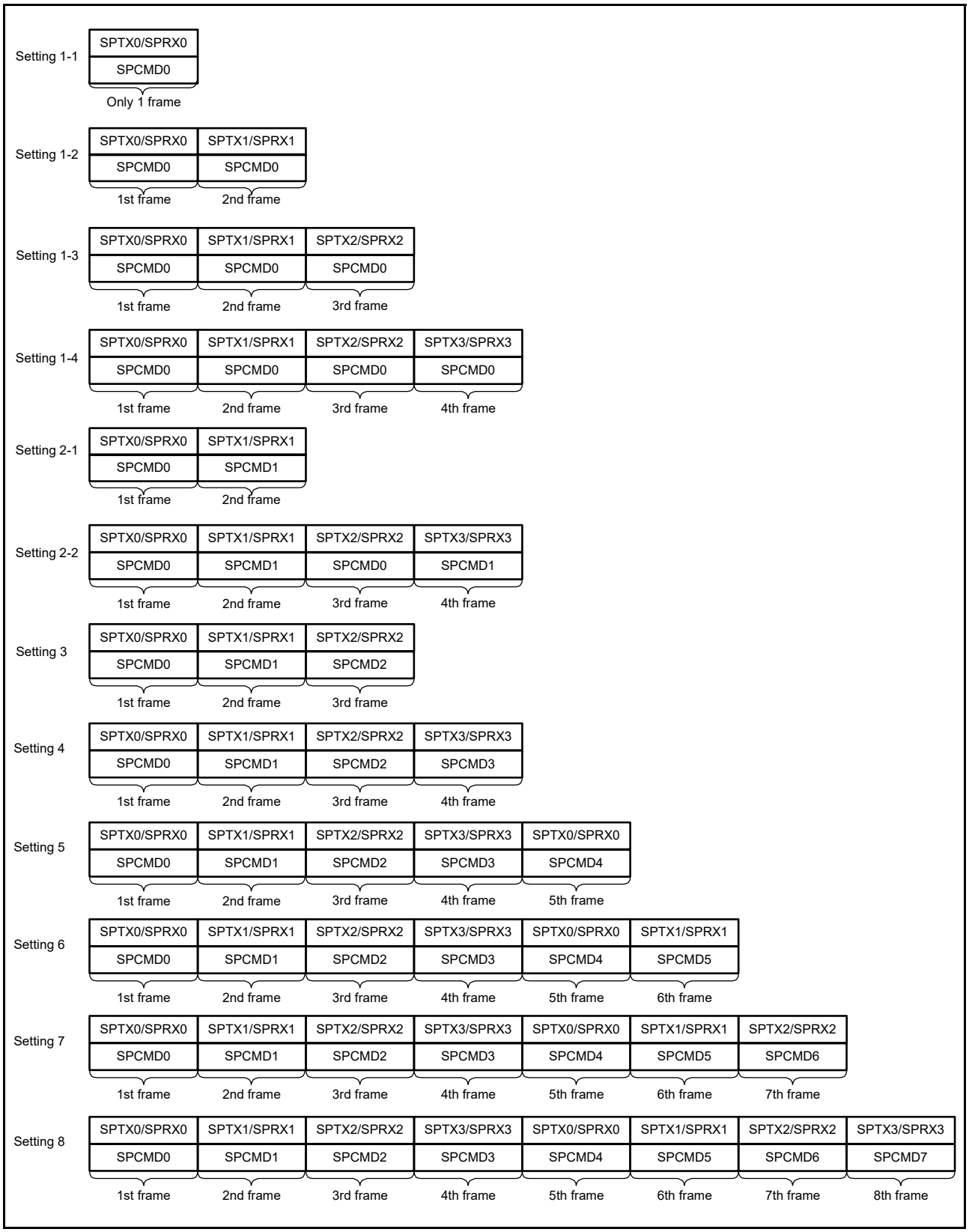


Figure 38.72 Correspondence between the RSPI Command Register m and Transmit/Receive Buffers in Sequence Operations

(4) Initialization Flowchart

Figure 38.73 is a flowchart illustrating an example of initialization in clock synchronous operation when the RSPIA is used in master mode. For a description of how to set up the interrupt controller, DMAC, and I/O ports, refer to the descriptions given in the individual blocks.

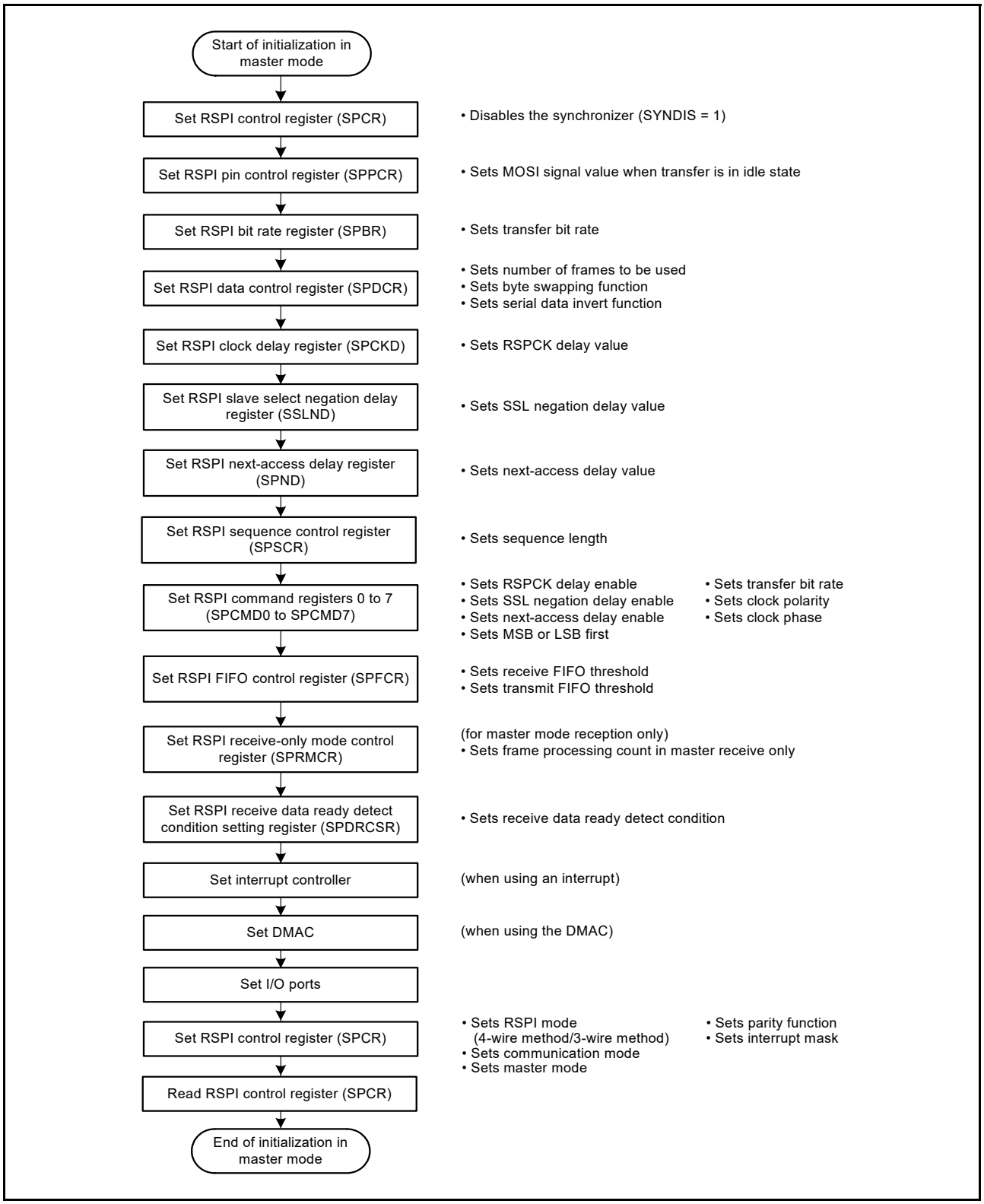


Figure 38.73 Example of Initialization Flowchart in Master Mode (Clock Synchronous Operation)

(5) Flow of Software Processing

Software processing during clock-synchronous operation when the RSPIA is used in master mode is the same as that for SPI master mode operation. For details, refer to section 38.3.13.1, (9) Software Processing Flow. Note that mode fault errors will not occur.

38.3.14.2 Slave Mode Operation

(1) Starting a Serial Transfer

When the SPCR.SPMS bit is 1, the first RSPCK0 edge triggers the start of a serial transfer in the RSPIA.
When the SPMS bit is 1, the RSPIA drives the MISO0 output signal.
For details on the RSPI transfer format, refer to section 38.3.5, Transfer Format (Frame Format).
It should be noted that the SSL00 input signal is not used in clock synchronous operation.

(2) Terminating a Serial Transfer

The RSPIA terminates the serial transfer after detecting an RSPCK0 edge corresponding to the final sampling timing. When the number of data stored in the receive FIFO is less than the number of FIFO stages, upon termination of serial transfer the RSPIA copies received data from the shift register to the receive buffer of the SPDR register. Upon termination of a serial transfer the RSPIA changes the status of the shift register to “empty” regardless of the receive buffer status. The final sampling timing changes depending on the bit length of transmit data. In slave mode, the RSPI data length depends on the SPCMD0.SPB[4:0] bit setting.
For details on the RSPI transfer format, refer to section 38.3.5, Transfer Format (Frame Format).

(3) Initialization Flowchart

Figure 38.74 is a flowchart illustrating an example of initialization in clock synchronous operation when the RSPIA is used in slave mode. For a description of how to set up the interrupt controller, DMAC, and I/O ports, refer to the descriptions given in the individual blocks.

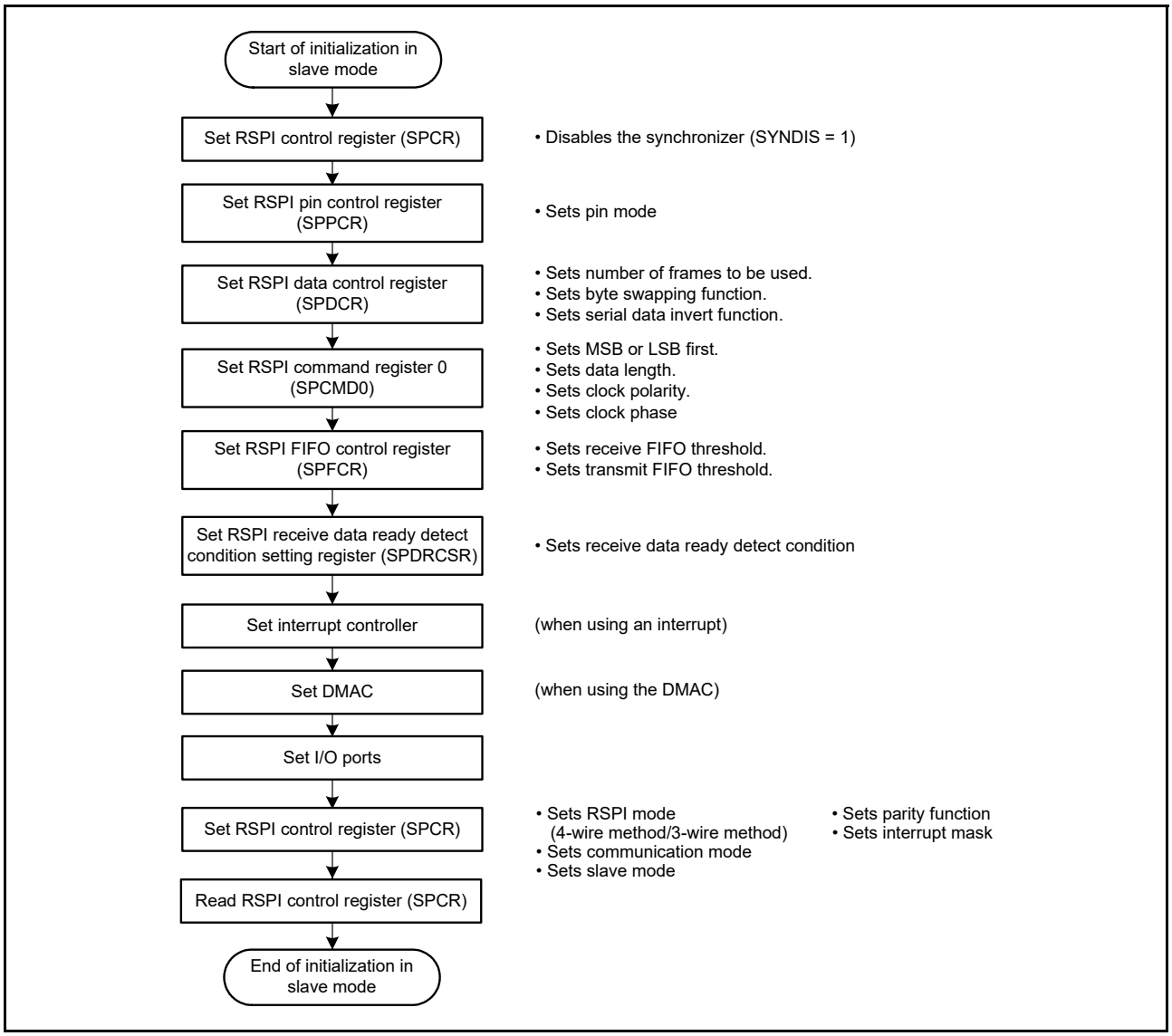


Figure 38.74 Example of Initialization Flowchart in Slave Mode (Clock Synchronous Operation)

(4) Flow of Software Processing

Software processing during clock-synchronous operation when the RSPIA is used in slave mode is the same as that for SPI slave mode operation. For details, refer to section 38.3.13.2, (6) Software Processing Flow. Note that mode fault errors will not occur.

38.3.15 Loopback Mode

When 1 is written to the SPPCR.SPLP2 bit or SPPCR.SPLP bit, the RSPIA shuts off the path between the MISO0 pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSI0 pin and the shift register if the SPCR.MSTR bit is 0, and connects the input path and output path of the shift register. The RSPIA does not shut off the path between the MOSI0 pin and the shift register if the SPCR.MSTR bit is 1, and between the MISO0 pin and the shift register if the SPCR.MSTR bit is 0. This is called loopback mode.
When a serial transfer is executed in loopback mode, the transmit data for the RSPIA or the reversed transmit data becomes the received data for the RSPIA.
Table 38.12 lists the relationship among the SPLP2 and SPLP bits and the received data.

Table 38.12 SPLP2 and SPLP Bit Settings and Received Data

SPPCR.SPLP2 Bit	SPPCR.SPLP Bit	Received Data
0	0	Input data from the MOSI0 pin or MISO0 pin
0	1	Inverted transmit data
1	x	Transmit data

x: Don’t care.

Figure 38.75 shows the configuration of the shift register I/O paths in loopback mode.

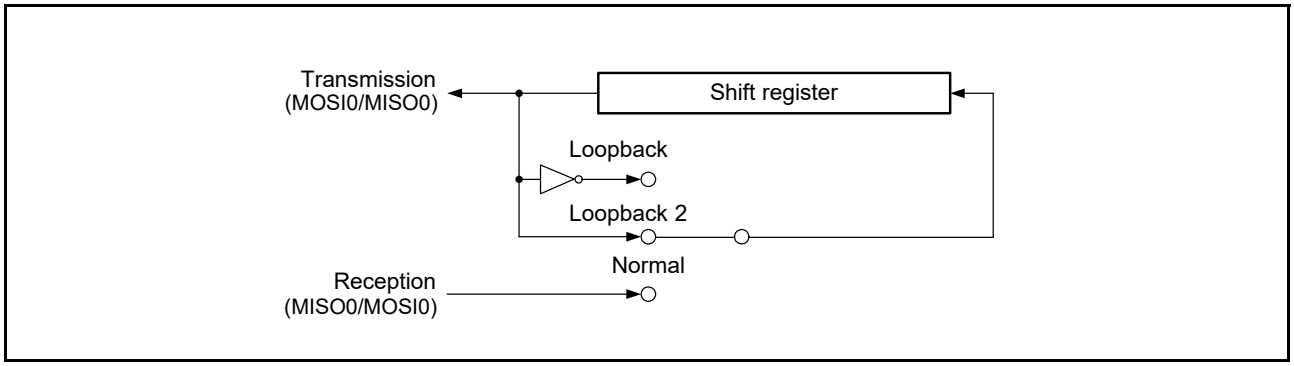


Figure 38.75 Configuration of Shift Register I/O Paths in Loopback Mode

38.3.16 Self-Diagnosis of Parity Bit Function

The parity circuit consists of a parity bit adding unit used for transmit data and an error detecting unit used for received data. In order to detect defects in the parity bit adding unit and error detecting unit of the parity circuit, self-diagnosis is executed for the parity circuit following the flowchart shown in Figure 38.76.

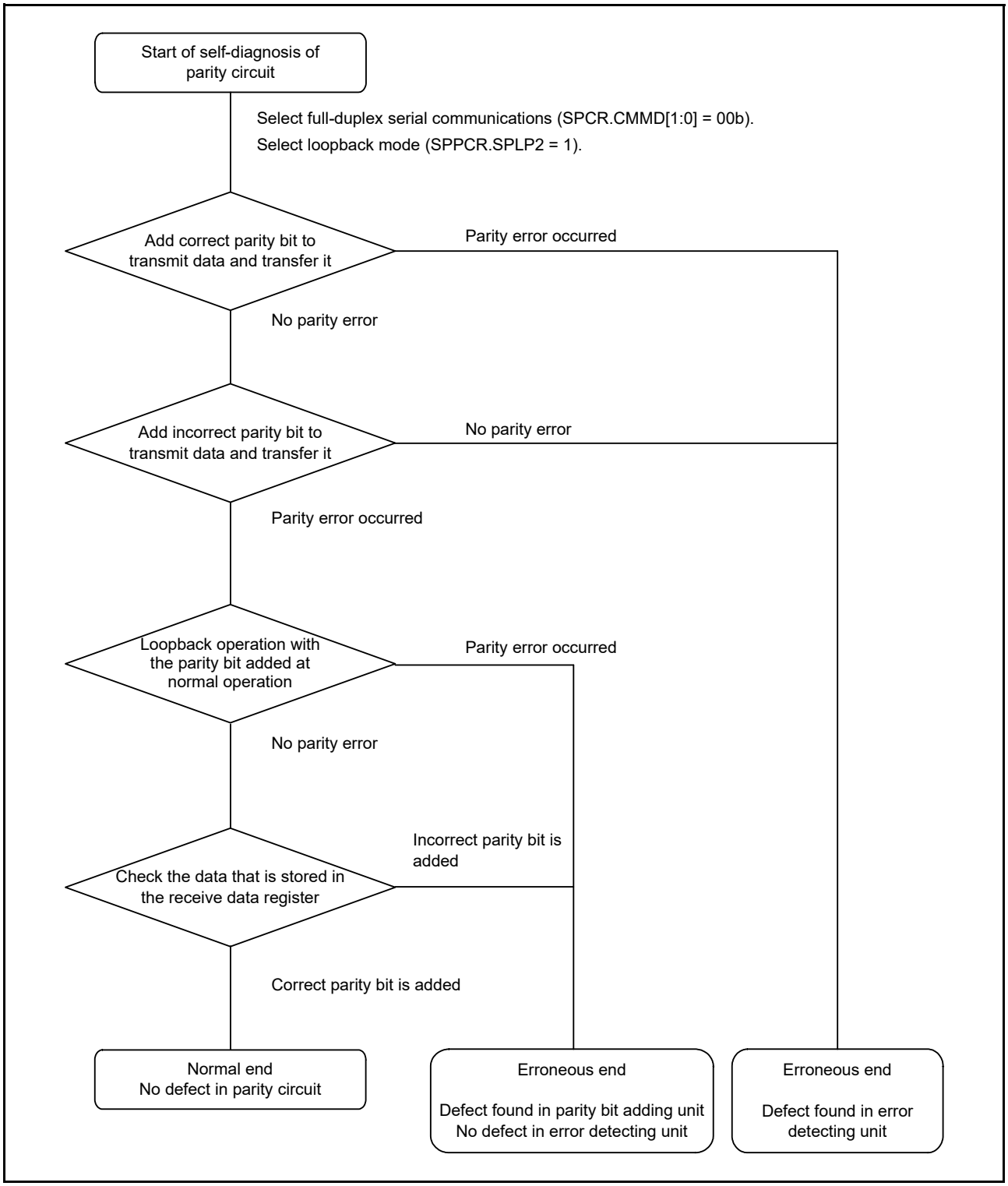


Figure 38.76 Flowchart for Self-Diagnosis of Parity Circuit

38.3.17 Interrupt Sources

The RSPIA has interrupt sources of receive buffer full, transmit buffer empty, error (mode fault, underrun, overrun, and parity error), idle, communication end, and receive data ready. In addition, the DTC or DMAC can be activated by the receive buffer full or transmit buffer empty interrupt to perform data transfer.

Since the vector address for SPEI is allocated to interrupt requests due to mode fault, underrun, overrun, parity errors, and receive data ready (only when the RDRIS bit is 1), the actual interrupt source must be determined from the flags.

Interrupt sources for the RSPIA are listed in Table 38.13. An interrupt is generated on satisfaction of an interrupt condition in Table 38.13. Clear the receive buffer full and transmit buffer empty sources through data transfer.

When using the DTC or DMAC to perform data transmission/reception, the DTC or DMAC must be set up first to be in a status in which transfer is enabled before making the RSPIA settings. For the method for setting the DTC or DMAC, refer to section 17, DMA Controller (DMACAa), or section 18, Data Transfer Controller (DTCb).

If the conditions for generating a transmit buffer empty or receive buffer full interrupt are generated while the ICU.IRn.IR flag is 1, the interrupt is not output as a request for ICU but is retained internally (the capacity for retention is one request per source). A retained interrupt request is output when the ICU.IRn.IR flag becomes 0. A retained interrupt request is automatically discarded once it is output as an actual interrupt request. The interrupt enable bit (the SPCR.SPTIE or SPCR.SPRIE bit) for an internally retained interrupt request can also be cleared to 0.

Table 38.13 Interrupt Sources of RSPIA

Interrupt Source	Symbol	Interrupt Condition	DMAC/DTC Activation
Receive buffer full Receive data ready	SPRI	The SPSR.SPRF flag is set to 1 while the SPCR.SPRIE bit is 1, or the RRDYF flag is set to 1 while the SPEIE is 1 and RDRIS bit is 0.	Possible
Transmit buffer empty	SPTI	The SPSR.SPTEF flag is set to 1 while the SPCR.SPTIE bit is 1.	Possible
Mode fault error Underrun error Overrun error Parity error Receive data ready	SPEI	The SPSR.MODF, OVRF, or PERF flag is set to 1 while the SPCR.SPEIE bit is 1, or the RRDYF flag is set to 1 while the SPEIE and RDRIS bits are 1.	Impossible
Idle	SPII	The SPSR.IDLNF flag is set to 0 while the SPCR.SPIIE bit is 1.	Impossible
Communication end	SPCI	The SPSR.SPCF flag is set to 1 while the SPCR.SPCIE bit is 1.	Impossible

38.3.18 Link Operation by Event Linking

The RSPIA supports the following event output for the event link controller (ELC). The event link output signal is output regardless of the interrupt enable bit setting.

38.3.18.1 Receive Buffer Full Event Output

This event signal is output when the number of data stored in the receive FIFO is greater than the threshold value, or when the SPCR.RDRIS bit = 0 and after writing to the receive FIFO, and the number of data stored in the receive FIFO is equal to or less than the threshold value, the value set in the SPDRCSR register has elapsed.
A receive buffer full event is output at the timing shown in Figure 38.77.

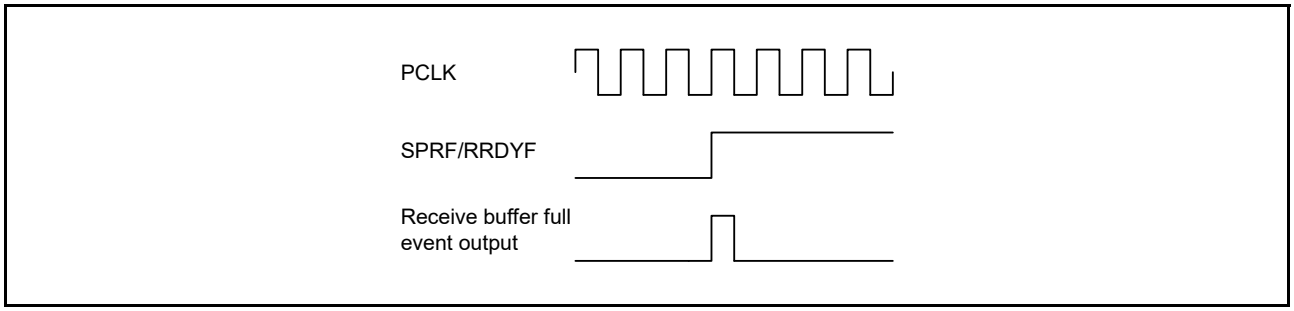


Figure 38.77 Event Output Timing of the Receive Buffer Full

38.3.18.2 Transmit Buffer Empty Event Output

This event signal is output when the number of empty stages in the transmit FIFO is greater than the threshold value or when the value of the SPCR.SPE bit has changed from 0 to 1.
A transmit buffer empty event is output at the timing shown in Figure 38.78.

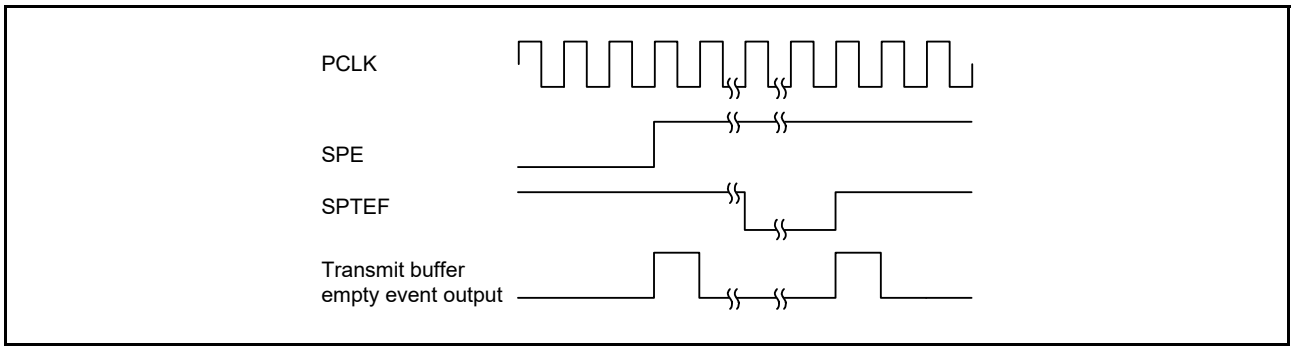


Figure 38.78 Event Output Timing of the Transmit Buffer Empty

38.3.18.3 Error Event Output

The error event signal is output due to five sources (mode fault, underrun, overrun, parity error, and receive data ready). For notes on error event output, see section 38.4.2, Note on Error Events Output.

(1) Mode Fault

A mode fault event occurs when all the following conditions are satisfied:

- SPI operation (SPCR.SPMS = 0)
- Slave mode (SPCR.MSTR = 0)
- The detection of mode fault error is enabled (SPCR.MODFEN = 1)
- Any of the following conditions are satisfied
  1. In the Motorola SPI mode, the SSL00 pin is deactivated during data transfer.
  2. In the TI SSP mode, the SSL00 pin is activated during data transfer.

(2) Underrun

The condition for this event signal being output in response to an underrun is the start of serial transfer while the value of the SPCR.MSTR bit is 0 (slave mode), the value of the SPCR.SPE bit is 1 (enables the RSPI function), and the transmit data are not ready for output.

(3) Overrun

The condition for this event signal being output in response to an overrun is completion of the serial transfer while the value of the SPCR.CMMD[1:0] bits are 00b or 10b, and the data is stored in the receive FIFO for the number of FIFO stages, in which case the SPSR.OVRF flag is set to 1.

(4) Parity Error

The condition for this event signal being output in response to a parity error is detection of a parity error on completion of the serial transfer while the value of the SPCR.CMMD[1:0] bits are 00b or 10b, and the value of the SPCR.SPE bit is 1.

(5) Receive Data Ready

The condition for this event signal being output in response to a receive data ready is elapsing of the value set by the SPDRSCR register when the number of data stored in the receive FIFO is less than or equal to the receive FIFO threshold value after the data has stored in the receive FIFO, while the value of the SPCR.CMMD[1:0] bits are 00b or 10b, the value of the SPCR.RDRIS bit is 1.

(6) Output Timing

Each error event signal is output at the timing shown in Figure 38.79.
Error event is output each time an error occurs.

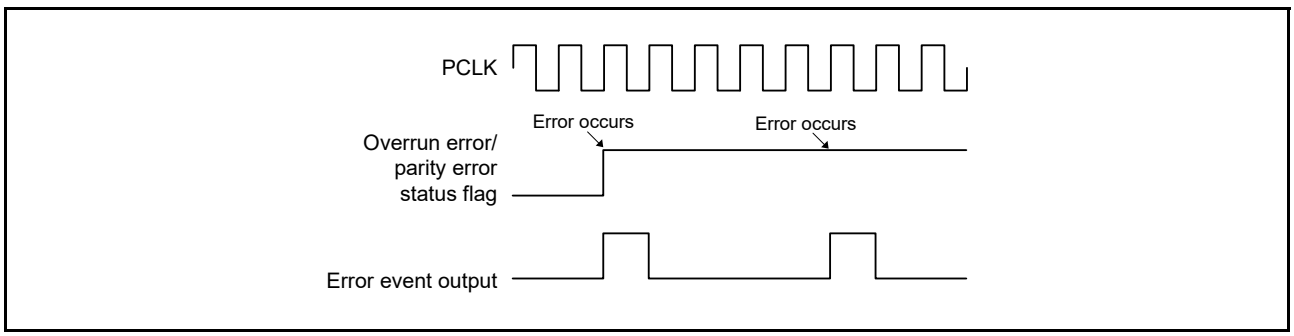


Figure 38.79 Error Event Output Timing

38.3.18.4 Idle Event Output

(1) In Master Mode

[Transmit-receive mode/transmit-only mode]

In transmit-receive master mode or transmit-only master mode, an event is output when the SPSR.IDLNF flag (idle flag) changes from 1 to 0.

The SPSR.IDLNF flag changes from 1 to 0 only when any of the following conditions are satisfied.

(a) The SPCR.SPE bit becomes 0 (RSPI is initialized) during transmission

(b) All the following conditions are satisfied:

- The next transmit data is not set in the transmission buffer (SPTXn, n = 0 to 3)
- The SPSSR.SPCP[2:0] bits are 000b (beginning of sequence control)
- The operations up to the next-access delay have finished (the master main state machine has entered the idle state)

[Receive-only mode]

In receive-only master mode, an event is output when any of the following conditions are satisfied.

(a) The SPCR.SPE bit is set to 0 (RSPI is initialized).

(b) Any the following conditions are satisfied:

- When the SPRMCR.RFC[4:0] bits are 00000b, after 1 is written to the SPRMCR.TERM bit, the operations up to the next-access delay have finished (the master main state machine has entered the idle state)
- When the SPRMCR.RFC[4:0] bits are other than 00000b, after 1 is written to the SPRMCR.TERM bit, the operations up to the next-access delay have finished (the master main state machine has entered the idle state)
- When the SPRMCR.RFC[4:0] bits are other than 00000b, after the operation is completed for the number of received frames set by the SPRMCR.RFC[4:0] bits, the operations up to the next-access delay have finished (the master main state machine has entered the idle state)

(2) In Slave Mode

In slave mode, an event is output when the SPCR.SPE bit is set to 0 (RSPI is initialized).

(3) Output Timing

An idle event is output at the timing shown in Figure 38.80.

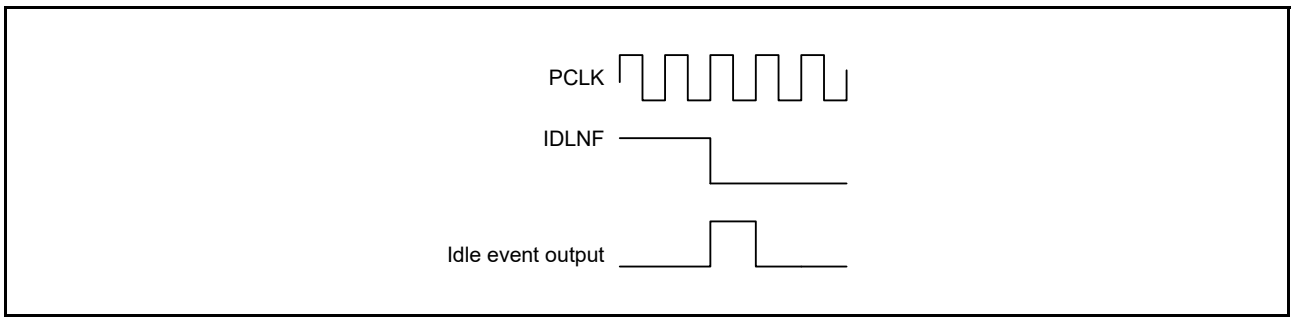


Figure 38.80 Idle Event Output Timing

38.3.18.5 Communication End Event Output

In master mode, an event is output under the condition for setting the SPSR.IDLNF flag (idle flag) from 1 to 0. In slave mode, an event is output under the conditions listed in Table 38.14 and Table 38.15.

Table 38.14 Generating Conditions of Communication End Event (Transmit-Receive Slave Mode/Transmit-Only Slave Mode)

RSPI Mode	Transmit Buffer State	Shift Register State	Others
SPI operation (SPMS = 0, FRFS = 0)	Empty	Empty	Negation of the SSL00 input
SPI operation (SPMS = 0, FRFS = 1)	Empty	Empty	Completion of the SSL negation delay
Clock synchronous operation (SPMS = 1)	Empty	Empty	Even edge detection of the last RSPCK0 for the last data

Table 38.15 Generating Conditions of Communication End Event (Receive-Only Slave Mode)

RSPI Mode	Condition
SPI operation (SPMS = 0, FRFS = 0)	Negation of the SSL00 input after storing the number of frames set by the SPDCR.SPFC[1:0] bits in the receive buffer
SPI operation (SPMS = 0, FRFS = 1)	Completion of the SSL negation delay after storing the number of frames set by the SPDCR.SPFC[1:0] bits in the receive buffer
Clock synchronous operation (SPMS = 1)	Even edge detection of the last RSPCK0 for the last frame set by the SPDCR.SPFC[1:0] bits

Whether the operation is in master mode or slave mode, an event is not output if 0 is written to the SPCR.SPE bit during communication or the SPCR.SPE bit is cleared by the mode fault error or underrun error.
A communication end event is output at timings shown in Figure 38.81 to Figure 38.86. The communication end event output timing in master operation is omitted because it is output at the same timing as an idle event.

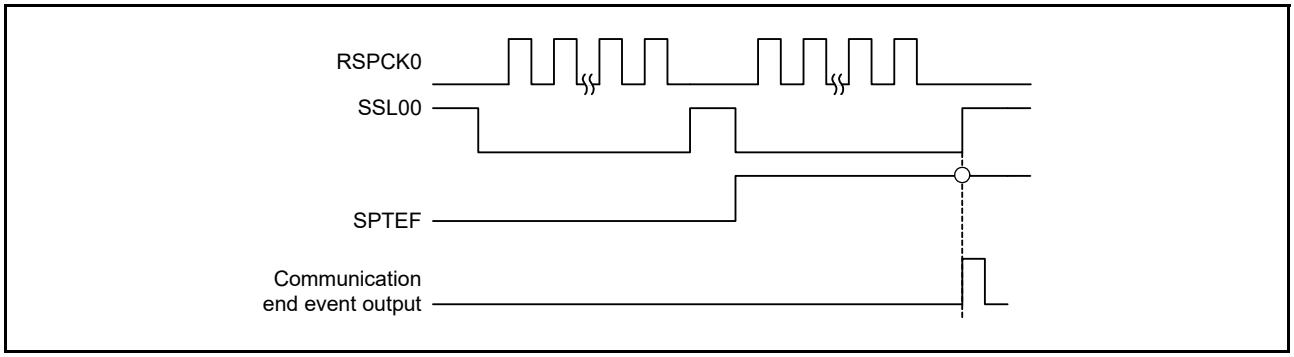


Figure 38.81 Communication End Event Output Timing (Transmit-Receive Slave Mode/Transmit-Only Slave Mode, Motorola SPI Operation)

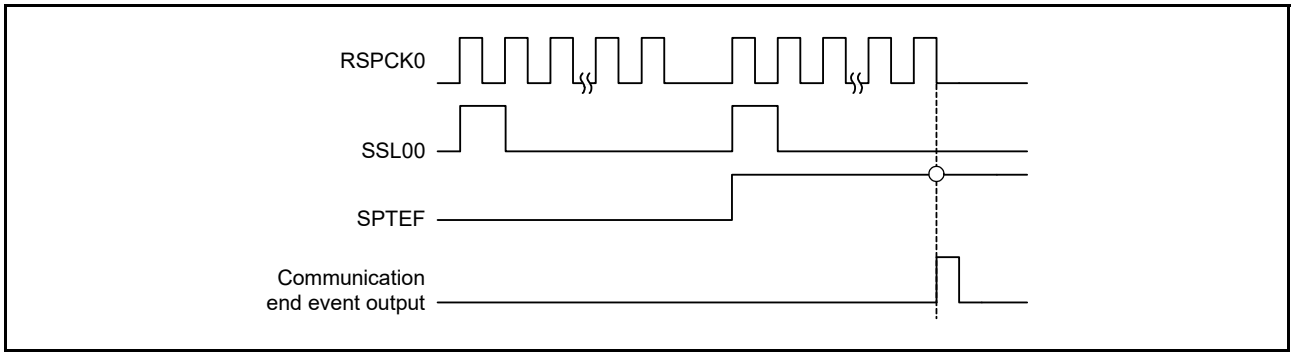


Figure 38.82 Communication End Event Output Timing (Transmit-Receive Slave Mode/Transmit-Only Slave Mode, TI SSP Operation)

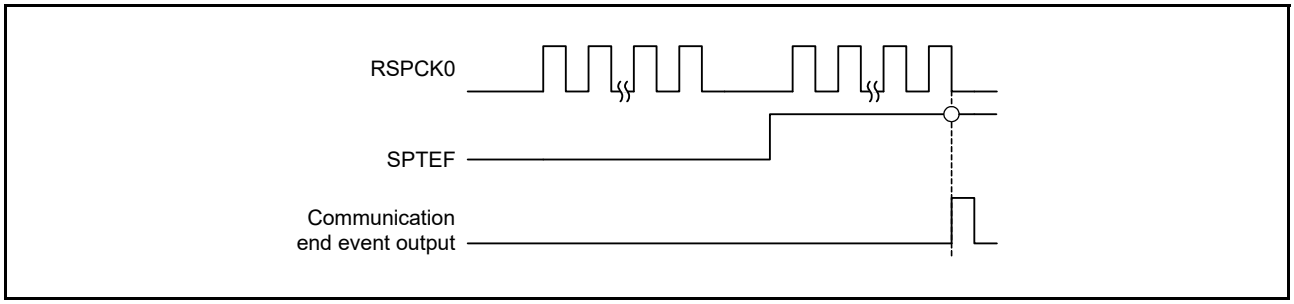


Figure 38.83 Communication End Event Output Timing (Transmit-Receive Slave Mode/Transmit-Only Slave Mode, Clock Synchronous Operation)

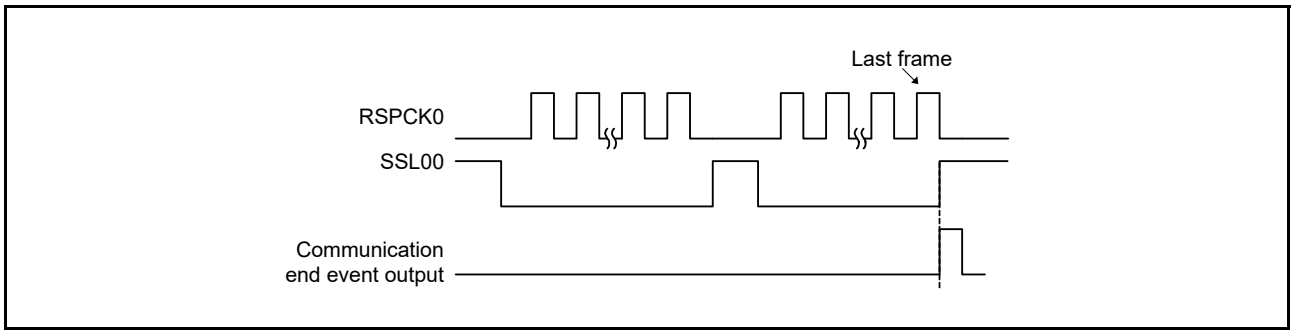


Figure 38.84 Communication End Event Output Timing (Receive-Only Slave Mode, Motorola SPI Operation)

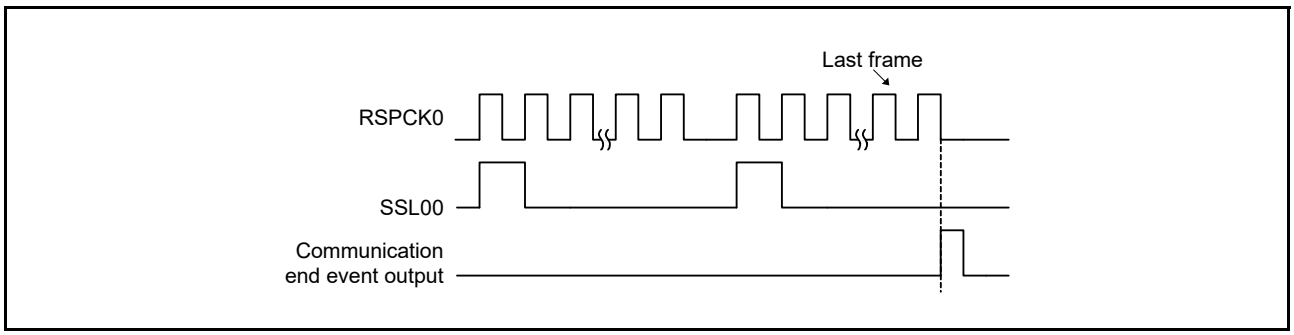


Figure 38.85 Communication End Event Output Timing (Receive-Only Slave Mode, TI SSP Operation)

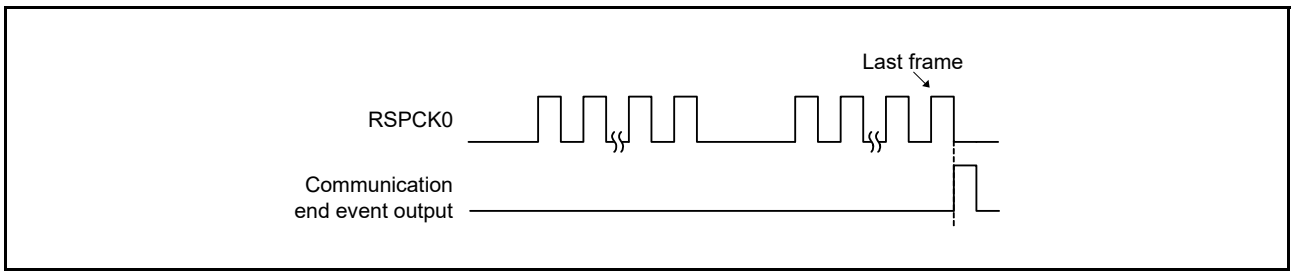


Figure 38.86 Communication End Event Output Timing (Receive-only slave mode, Clock Synchronous Operation)

38.4 Usage Notes

38.4.1 Notes on Starting Transfer

When the ICU.IRn.IR flag is 1 at the time transfer is to start, follow the procedure below to clear interrupt requests before enabling operations (by setting the SPCR.SPE bit to 1).

1. Confirm that transfer has stopped (i.e. that the SPCR.SPE bit is 0).
2. Set the relevant interrupt enable bit (the SPCR.SPTIE or SPCR.SPRIE bit) to 0.
3. Read the relevant interrupt enable bit (the SPCR.SPTIE or SPCR.SPRIE bit) and confirm that its value is 0.
4. Set the ICU.IRn.IR flag to 0.

38.4.2 Note on Error Events Output

To enable SPI operation of the RSPIA in master mode in the multi-master environment (SPCR.SPMS bit = 0, SPCR.MSTR bit = 1, and SPCR.MODFEN bit = 1), do not use any output of error events.

38.4.3 Note on Low Power Consumption Functions

To reduce power consumption of this module by using the low power consumption functions, set the SPCR.SPE bit to 0 to terminate communication, and then use the low power consumption functions.

38.4.4 Note on the SPRF and SPTEF flags

When polling the SPSR.SPRF flag and/or SPSR.SPTEF flag, set the SPCR.SPRIE bit and/or SPCR.SPTIE bit to 0.

38.4.5 Notes on Burst Transfer in Master Mode

During burst transfer, changing the following settings between the SPCMDm register with the SSLKP bit set to 1 and the SPCMDm register used in the next transfer is prohibited.

- SSL Signal Assertion setting (SSLA[2:0] bits)
- RSPCK output setting (CPHA, CPOL, BRDV[1:0] bits)

38.4.6 Notes on Operating in Slave, TI SSP Mode

In the slave TI SSP mode, the delay between frames must observe the following intervals. Secure the following intervals on the master side.

[Master side: interval from the RSPCK edge of last bit to the RSPCK edge of next SSL] > [Slave side: OE delay time = PCLKA * (1 to 2) + SLNDL[2:0] setting value]

38.4.7 Notes on the Data Length

Table 38.16 shows the relationship between the data length and operation of the RSPIA.

Table 38.16 Relationship between the Data Length and Operation of the RSPIA

Data Length										
4 bit	**5 bit**	**6 bit**	**7 bit**	**8 bit**	**9 bit**	**10 bit**	**11 bit**	**12 bit**	**13 bit**	**14 bit**
Conditionally possible	Conditionally possible	Conditionally possible	Possible	Possible	Possible	Possible	Possible	Possible	Possible	Possible

Possible: Operating possible
Conditionally possible: Operating possible when the SPCR.SCKASE or SCKDDIS bit is 0.

38.4.8 Notes when SPCR.SPE = 1

If the value of the registers listed below are changed while the SPCR.SPE bit is 1, the operation after the change is not guaranteed.

Table 38.17 List of Registers that Cannot Be Rewritten When SPCR.SPE = 1 (1/2)

Register	Bit
SPCKD	SCKDL[2:0]
SSLND	SLNDL[2:0]
SPND	SPNDL[2:0]
SPCR	SYNDIS
	MSTR
	CMMD[1:0]
	FRFS
	SPMS
	MODFEN
	SCKDDIS
	SCKASE
	PTE
	SPOE
	SPPE
SPRMCR	RFC[4:0]
SPDRCSR	—
SPPCR	MOIFE
	MOIFV
	SPLP2
	SPLP
SSLP	SSL3P
	SSL2P
	SSL1P
	SSL0P
SPBR	—
SPSCR	SPSLN[2:0]

Table 38.17 List of Registers that Cannot Be Rewritten When SPCR.SPE = 1 (2/2)

Register	Bit
SPCMD0*1	SSLA[2:0]
	SPB[4:0]
	SCKDEN
	SLNDEN
	SPNDEN
	LSBF
	SSLKP
	BRDV[1:0]
	CPOL
	CPHA
SPDCR	SPFC[1:0]
	DINV
	SPRDTD
	BYSW
SPFCR	TTRG[1:0]
	RTRG[1:0]
SPFCLR	FCLR

Note 1. Rewriting prohibited in slave mode. In master mode, rewriting is available only when there is no next transmit data in the transmit FIFO.

39. CRC Calculator (CRCA)

The CRC (Cyclic Redundancy Check) calculator generates CRC codes.

39.1 Overview

Table 39.1 lists the specifications of the CRC calculator, and Figure 39.1 shows a block diagram of the CRC calculator.

Table 39.1 CRC Specifications

Item	Description	
Data size	8 bits	32 bits
Data for CRC calculation*1	CRC codes are generated for any desired data in 8n-bit units (where n is a whole number)	CRC codes are generated for any desired data in 32n-bit units (where n is a whole number)
CRC processor unit	8-bit parallel processing	32-bit parallel processing
CRC generating polynomial	One of three generating polynomials is selectable • 8-bit CRC $X^8 + X^2 + X + 1$ • 16-bit CRC $X^{16} + X^{15} + X^2 + 1$ $X^{16} + X^{12} + X^5 + 1$	One of two generating polynomials is selectable • 32-bit CRC $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$ $X^{32} + X^{28} + X^{27} + X^{26} + X^{25} + X^{23} + X^{22} + X^{20} + X^{19} + X^{18} + X^{14} + X^{13} + X^{11} + X^{10} + X^9 + X^8 + X^6 + 1$
CRC calculation switching	The order of the bits produced by CRC calculation can be switched for LSB first or MSB first communication	
Low power consumption	Module stop state can be set.	

Note 1. The circuit does not have a function to divide data for calculation into CRC calculation units. Write data in 8-bit or 32-bit units.

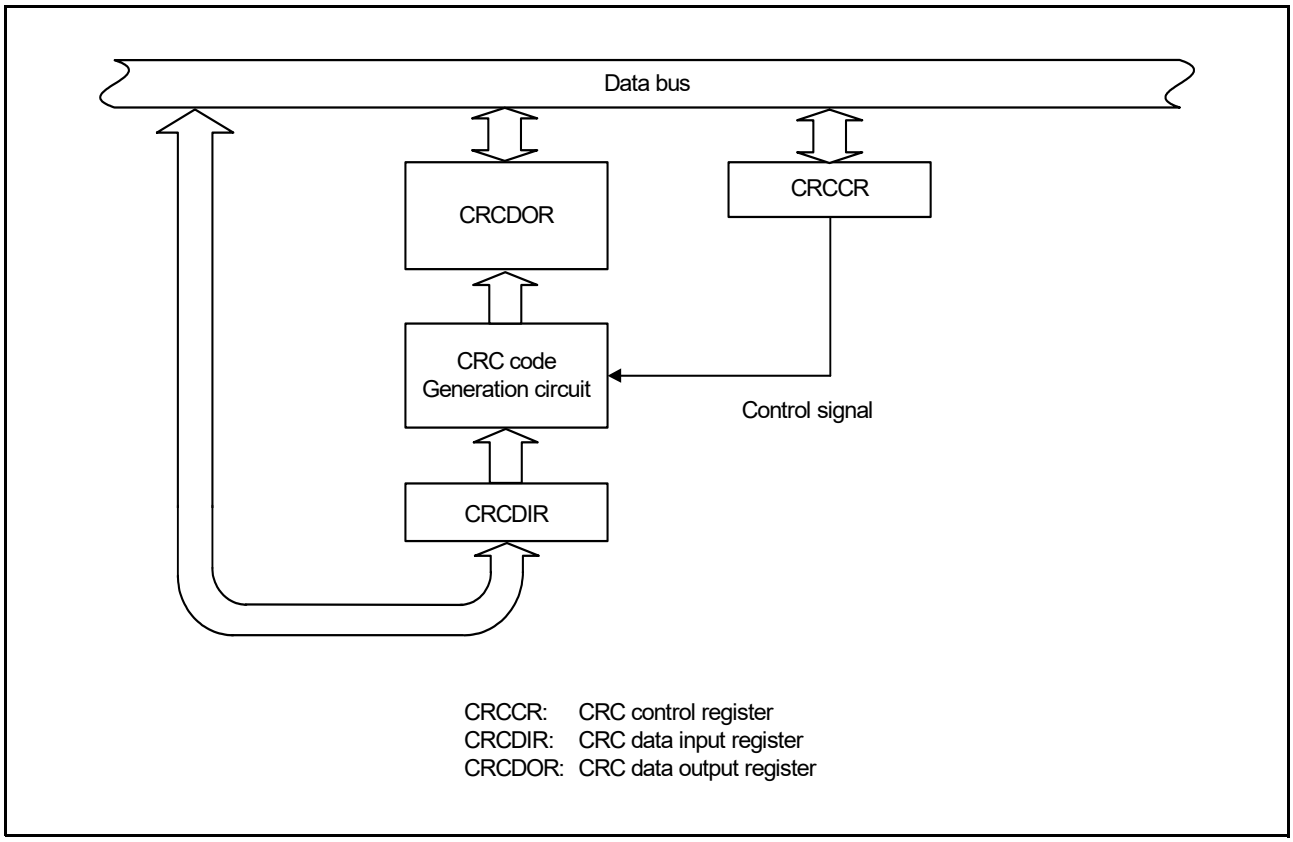


Figure 39.1 CRC Block Diagram

39.2 Register Descriptions

39.2.1 CRC Control Register (CRCCR)

Address(es): CRC.CRCCR 0008 8280h

	b7	b6	b5	b4	b3	b2	b1	b0
	DORCLR	LMS	—	—	—	GPS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	GPS[2:0]	CRC Generating Polynomial Switching	b2 b1 b0 0 0 0: No calculation is executed. 0 0 1: 8-bit CRC ($X^8 + X^2 + X + 1$) 0 1 0: 16-bit CRC ($X^{16} + X^{15} + X^2 + 1$) 0 1 1: 16-bit CRC ($X^{16} + X^{12} + X^5 + 1$) 1 0 0: 32-bit CRC ($X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$) 1 0 1: 32-bit CRC ($X^{32} + X^{28} + X^{27} + X^{26} + X^{25} + X^{23} + X^{22} + X^{20} + X^{19} + X^{18} + X^{14} + X^{13} + X^{11} + X^{10} + X^9 + X^8 + X^6 + 1$) 1 1 0: No calculation is executed. 1 1 1: No calculation is executed.	R/W
b5 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	LMS	CRC Calculation Switching	0: Generates CRC for LSB first communication. 1: Generates CRC for MSB first communication.	R/W
b7	DORCLR	CRCDOR Register Clear	1: Clears the CRCDOR register. This bit is read as 0.	W*1

Note 1. Only 1 can be written.

LMS Bit (CRC Calculation Switching)

The setting this bit selects the order of the bits of generated CRC codes. The bit selects transmission of the lower-order byte of the CRC code first for LSB first communication, or the higher-order byte first for MSB first communication. For details on the transmission and reception of CRC codes, refer to section 39.3, Operation.

DORCLR Bit (CRCDOR Register Clear)

Write 1 to this bit so that the CRCDOR register is set to 0000 0000h.
This bit is read as 0. Writing 0 to this bit has no effect.

39.2.2 CRC Data Input Register (CRCDIR)

- When 32-bit CRC is selected

Address(es): CRC.CRCDIR 0008 8284h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- When 16-bit or 8-bit CRC is selected

Address(es): CRC.CRCDIR 0008 8284h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

The CRCDIR register is a readable and writable register. Write data for CRC calculation to this register.
When generating a 32-bit CRC, the CRCDIR register should be accessed in longword units. When generating a 8-bit or 16-bit CRC, the CRCDIR register should be accessed in byte units.

39.2.3 CRC Data Output Register (CRCDOR)

- When 32-bit CRC is selected

Address(es): CRC.CRCDOR 0008 8288h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- When 16-bit CRC is selected

Address(es): CRC.CRCDOR 0008 8288h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- When 8-bit CRC is selected

Address(es): CRC.CRCDOR 0008 8288h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

The CRCDOR register is a readable and writable register.
The value after a reset is 0000 0000h. When calculating with the initial value set to a value other than 0000 0000h, rewrite the CRCDOR register.
Writing data to the CRCDIR register stores result of calculation in the CRCDOR register. In addition, following communication data, when a CRC code is written to the CRCDIR register and if the calculation result is 0000 0000h, there is no error in the communication data.
When a 32-bit CRC is selected, the CRCDOR register should be accessed in longword units. When a 16-bit CRC is selected, the CRCDOR register should be accessed in word units. When an 8-bit CRC is selected, the CRCDOR register should be accessed in byte units.

39.3 Operation

The CRC calculator generates CRC codes for use in LSB first or MSB first communication.
The following shows examples of generating the CRC code for input data (F0h) using the 16-bit CRC generating polynomial ($X^{16} + X^{12} + X^5 + 1$). In these examples, the value of the CRC data output register (CRCDOR) is cleared by setting the DORCLR bit to 1 to set the initial value for CRC calculation to 0000 0000h.

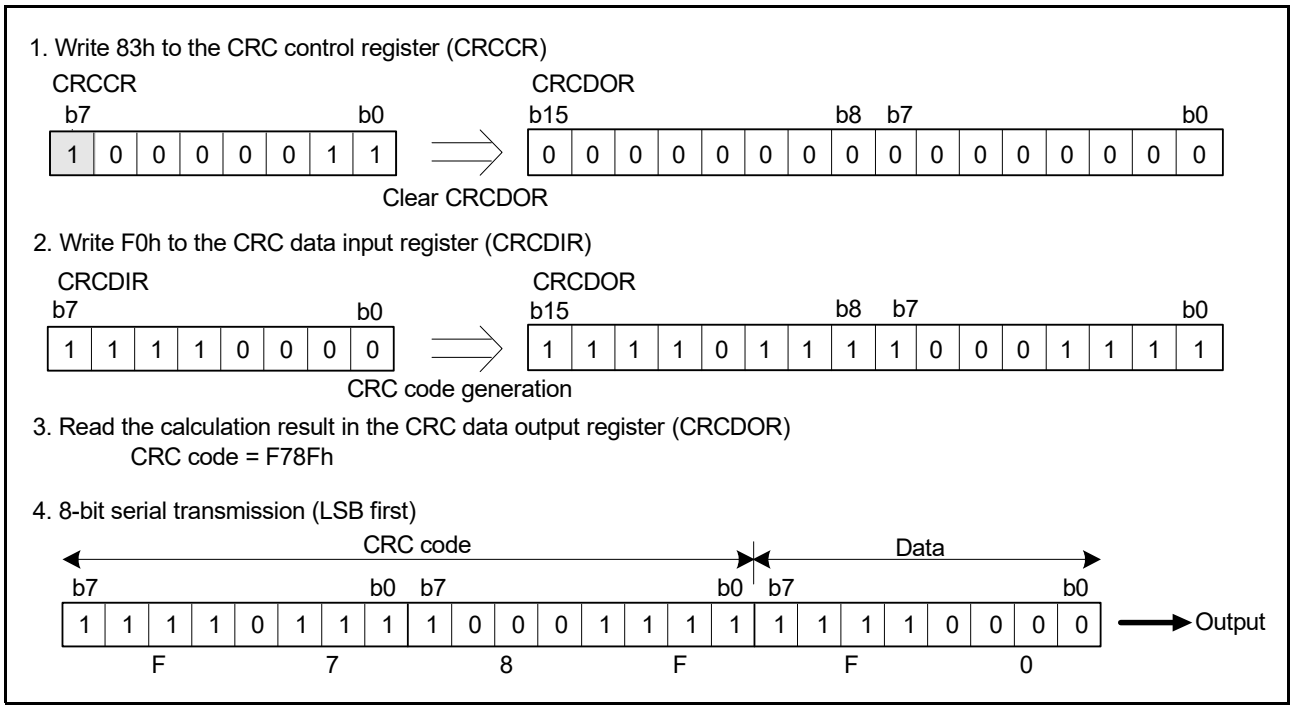


Figure 39.2 LSB First Data Transmission

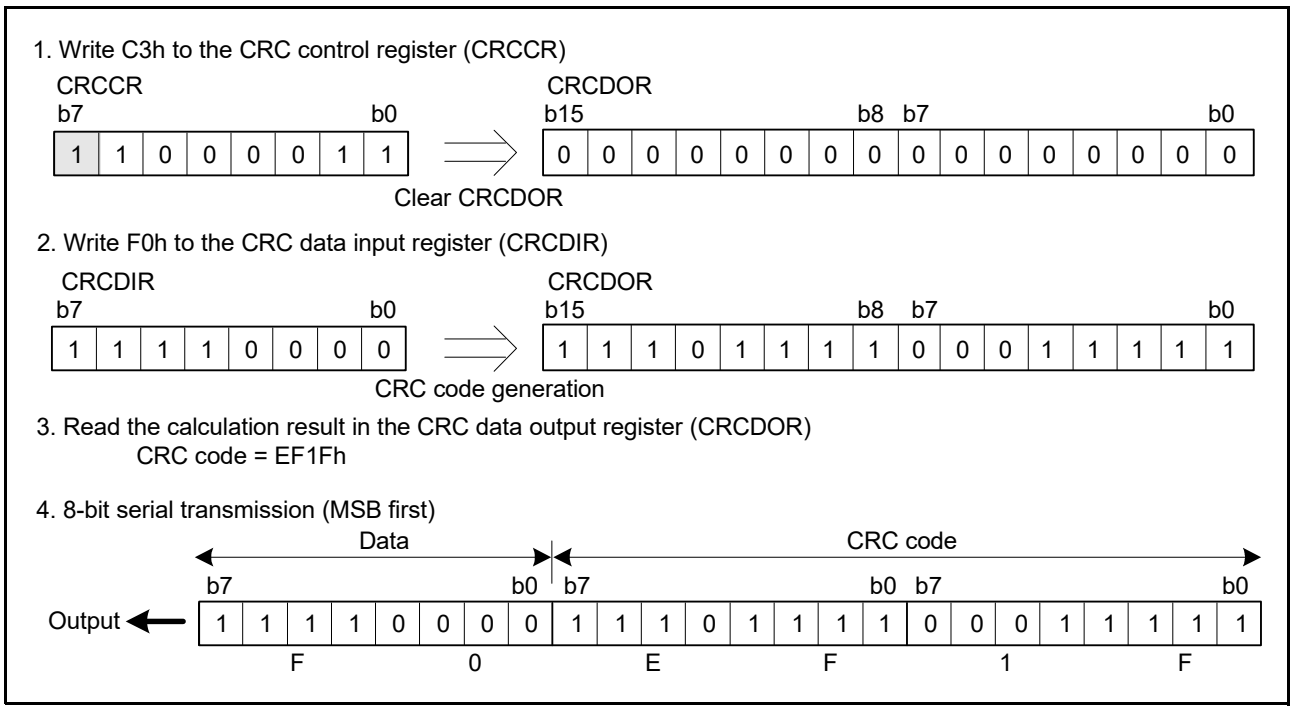


Figure 39.3 MSB First Data Transmission

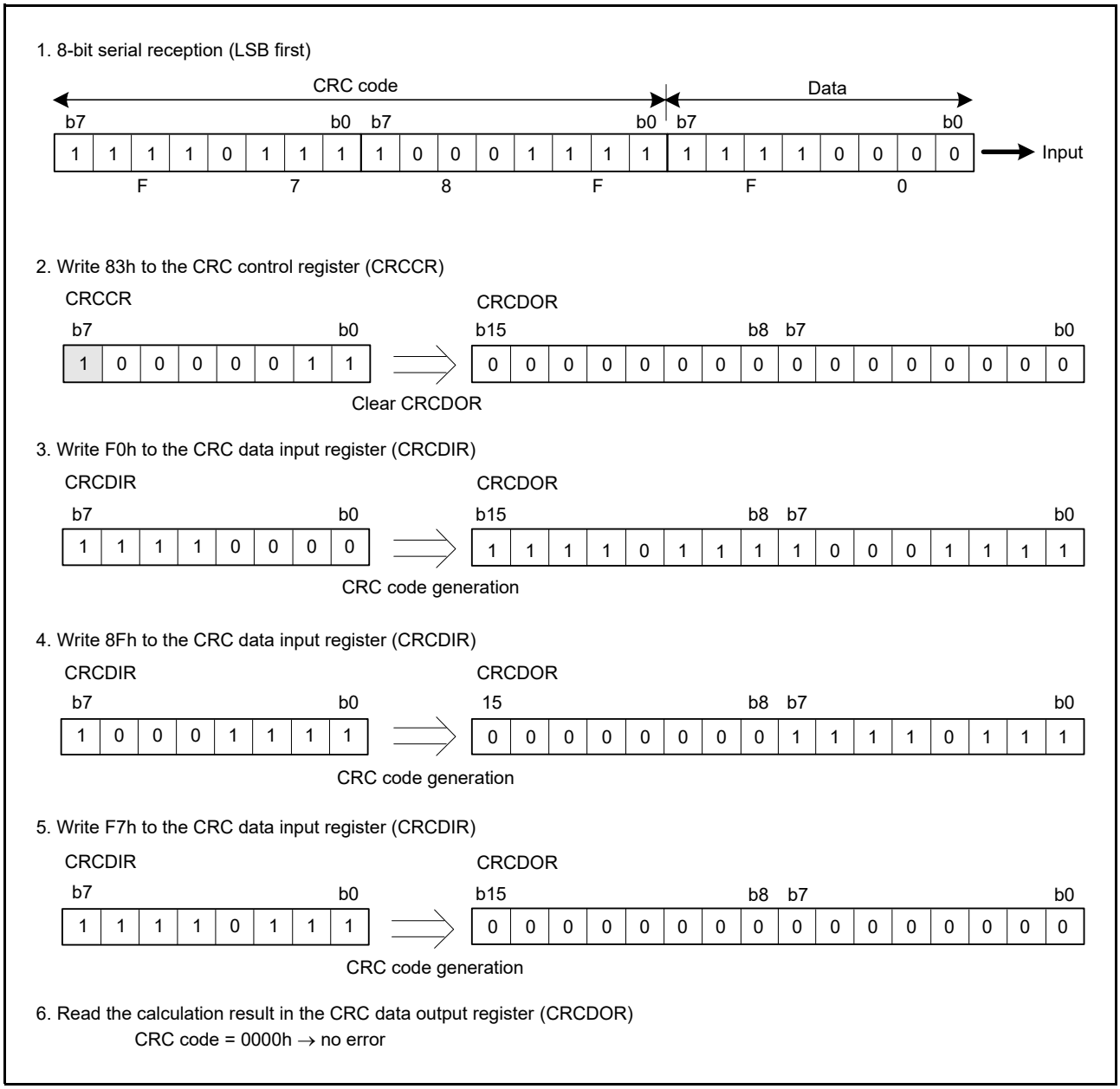


Figure 39.4 LSB First Data Reception

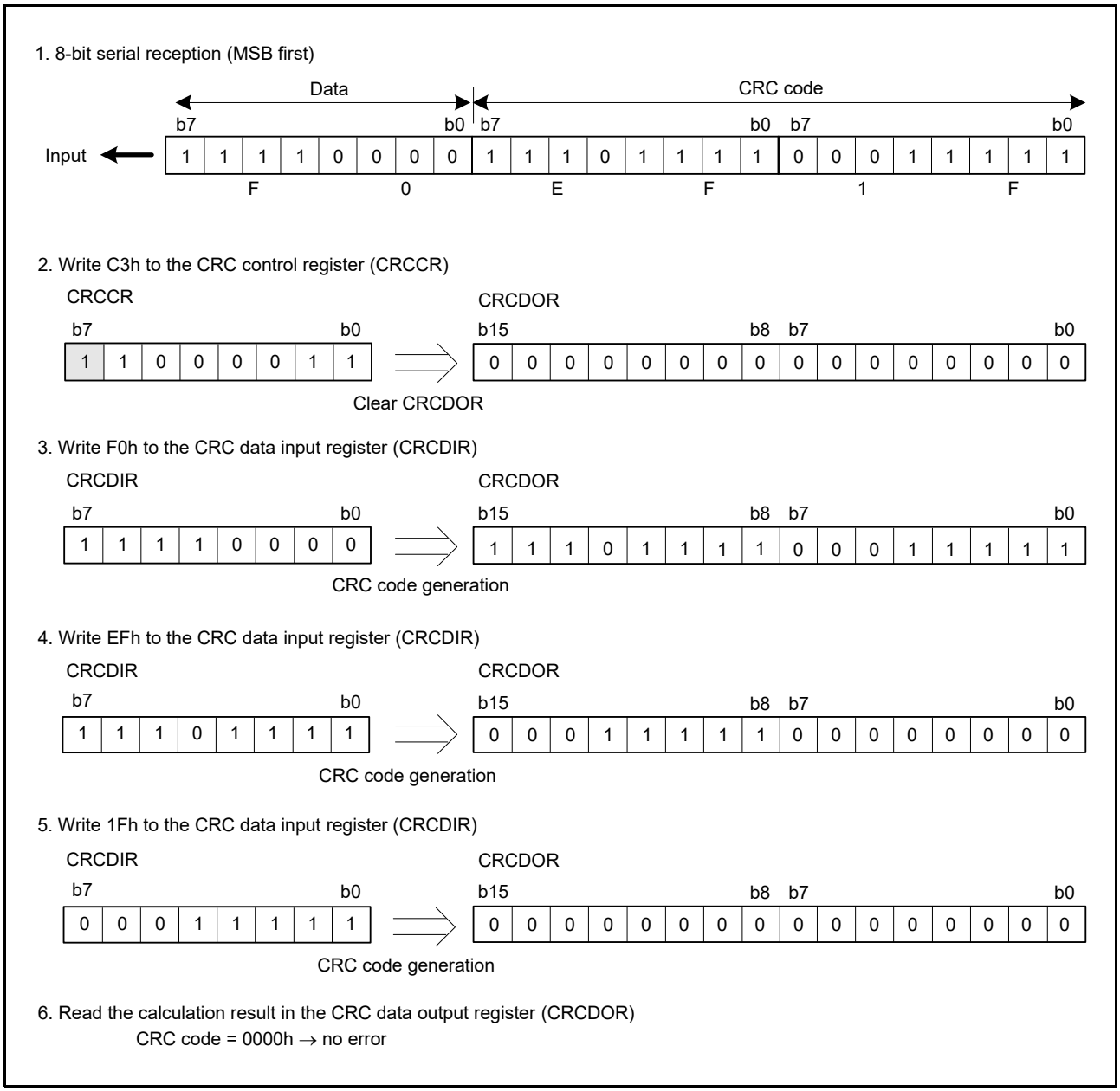


Figure 39.5 MSB First Data Reception

39.4 Usage Notes

39.4.1 Module Stop Function Setting

Operation of the CRC calculator can be disabled or enabled using the module stop control register B (MSTPCRB). After a reset, the CRC is in the module stop state. Register access is enabled by releasing the module stop state.
For details, refer to section 11, Low Power Consumption.

39.4.2 Note on Transmission

Note that the sequence of transmission for the CRC code differs according to whether transmission is LSB first or MSB first.

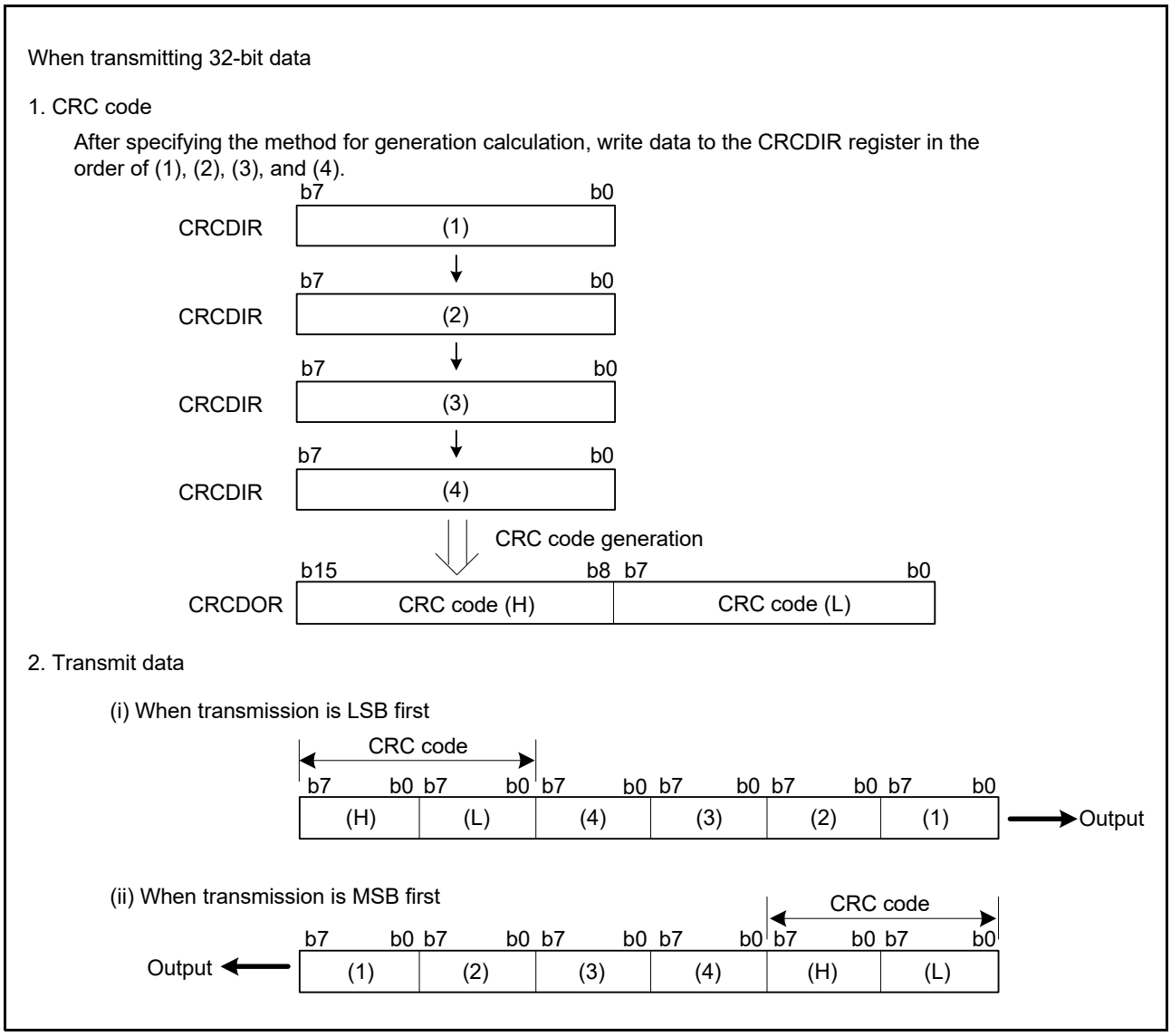


Figure 39.6 LSB First and MSB First Data Transmission

40. Arithmetic Unit for Trigonometric Functions (TFUv2)

This MCU has an arithmetic unit for trigonometric functions (TFUv2). TFUv2 has an upward compatible architecture with TFUv1 and supports fixed-point numbers as well as floating-point numbers.

The TFU has hardware processing functionality for sin, cos, atan, and hypot_k calculations that provide sinθ, cosθ, atan(y/x), and $\sqrt{x^2+y^2}/k$, respectively. For the value of k, see section 40.3.1, Arithmetic Processing.

40.1 Overview

Table 40.1 lists the specifications of the TFU.

Table 40.1 TFU specifications

Item	Description				
Arithmetic Processing	Calculation of sine, cosine, arctangent, and hypotenuse • A sine and cosine can be simultaneously calculated. • An arctangent and hypotenuse can be simultaneously calculated.				
Range and Unit of Values	With the input and output of floating-point numbers	I/O		Range	Unit
	Calculating sine	Input	Angle θ	–float_max ≤ θ ≤ float_max*1	radians
		Output	sinθ	–1.0 ≤ sinθ ≤ 1.0	—
	Calculating cosine	Input	Angle θ	–float_max ≤ θ ≤ float_max*1	radians
		Output	cosθ	–1.0 ≤ cosθ ≤ 1.0	—
	Calculating arctangent	Input	x and y coordinates	–float_max ≤ x ≤ float_max*1 –float_max ≤ y ≤ float_max*1	—
		Output	atan(y/x)	–π ≤ atan(y/x) ≤ π	radians
	Calculating hypotenuse	Input	x and y coordinates	–float_max ≤ x ≤ float_max*1 –float_max ≤ y ≤ float_max*1	—
		Output	$\sqrt{x^2+y^2}/k$	0 ≤ $\sqrt{x^2+y^2}/k$ ≤ ∞	—
	With the input and output of fixed-point numbers	I/O		Range	Unit
	Calculating sine	Input	Angle θ	–1.0 ≤ θ < 1.0	turns
				–4.0 ≤ θ < 4.0	radians
		Output	sinθ	–1.0 ≤ sinθ ≤ 1.0	—
	Calculating cosine	Input	Angle θ	–1.0 ≤ θ < 1.0	turns
				–4.0 ≤ θ < 4.0	radians
		Output	cosθ	–1.0 ≤ cosθ ≤ 1.0	—
	Calculating arctangent	Input	x and y coordinates	–1.0 ≤ x < 1.0	—
				–1.0 ≤ y < 1.0	
		Output	atan(y/x)	–0.5 ≤ atan(y/x) ≤ 0.5	turns
				–π ≤ atan(y/x) ≤ π	radians
	Calculating hypotenuse	Input	x and y coordinates	–1.0 ≤ x < 1.0	—
				–1.0 ≤ y < 1.0	
		Output	$\sqrt{x^2+y^2}/k$	0 ≤ $\sqrt{x^2+y^2}/k$ ≤ $\sqrt{2}/k$	—
Data Type for Processing	Single-precision floating-point, fixed-point two's-complement				
Number of cycles for calculation	Sine: 5 Cosine: 5 Arctangent: 14 Hypotenuse: 14				

Note 1. float_max is the maximum value that can be expressed as single-precision floating-point: $(2 - 2^{-23}) \times 2^{127}$.

40.2 Register Descriptions

40.2.1 Fixed-Point Sincos Input and Output Setting Register (FXSCIOC)

Address(es): TFU.FXSCIOC 0008 1404h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	OF[1:0]		—	—	—	IUF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IUF	Input Unit and Format Setting	0: Unit of turns, Q1.31 format 1: Unit of radians, Q3.29 format	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5, b4	OF[1:0]	Output Format Setting	b5 b4 0 0: Q1.31 0 1: Q2.30 1 0: Q3.29 1 1: Setting prohibited	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Set this register before starting a calculation. Operation cannot be guaranteed if writing to this register is attempted during a calculation.

IUF Bit (Input Unit and Format Setting)

The IUF bit is used to set the unit and format of the input values in fixed-point sincos operations.

OF[1:0] Bits (Output Format Setting)

The OF[1:0] bits are used to set the format of the output values in fixed-point sincos operations. Exactly expressing the value +1.0 is not possible in Q1.31 format. When the exact value of $\sin\theta$ or $\cos\theta$ should be +1.0 in Q1.31 format, the value read from the FXSCDT0 or FXSCDT1 register is $+1.0 - 2^{-31}$ instead.

40.2.2 Fixed-Point Atanhypot_k Input and Output Setting Register (FXATIOC)

Address(es): TFU.FXATIOC 0008 1405h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	OUF	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	OUF	Output Unit and Format Setting	0: Unit of turns, Q1.31 format 1: Unit of radians, Q3.29 format	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Set this register before starting a calculation. Operation cannot be guaranteed if writing to this register is attempted during a calculation.

OUF Bit (Output Unit and Format Setting)

The OUF bit is used to set the unit and format of the output values of atan calculations in fixed-point atanhypot_k operations. Note that the format of the output value of every hypot_k calculation simultaneous with an atan calculation is Q3.29 regardless of the value of this bit.

40.2.3 Trigonometric Status Register (TRGSTS)

Address(es): TFU.TRGSTS 0008 1408h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	ERRF	BSYF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BSYF	Calculation in Progress Flag	0: No calculating 1: Calculating	R/W *1
b1	ERRF	Input Error Flag	0: No input error occurred 1: Input error occurred	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing to this flag is invalid.

BSYF Flag (Calculation in Progress Flag)

The BSYF flag indicates whether a calculation is in progress. Writing to this flag is ignored.

[Setting condition]

- When calculation started

[Clearing condition]

- When calculation completed

ERRF Flag (Input Error Flag)

The ERRF flag indicates whether an input error occurred during a calculation. Follow the procedure for restoring the error flag described in section 40.4.2, How to Use Interrupts, to write to this register.

[Setting condition]

- When input error occurred*1 or this flag is set to 1

[Clearing condition]

- When the next calculation started or this flag is set to 0

Note 1. For details on input error, see section 40.3.5, Relationship Between Input and Output Values for Atan Operation.

40.2.4 Floating-Point Sincos Data Register 0 (FPSCDT0)

Address(es): TFU.FPSCDT0 0008 1410h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The FPSCDT0 register is used to read the output value of cosθ in a floating-point sincos operation. For details, see Table 40.2. For how to use this register, see section 40.4.1, Procedures for Using the TFU.
Writing to this register during operation is prohibited. If read access is made to this register during the calculation, the result is read after the calculation is completed. At this time, the bus access is forced to wait until the operation is completed.
Only write to or read from this register in a floating-point sincos operation. Writing to any other data register 0, that is, to the FPATDT0, FXSCDT0, or FXATDT0 register, makes the value of this register undefined. Writing to data saving and restoring register 0 (DTSR0) makes the value of this register the same as that written to the DTSR0 register.

40.2.5 Floating-Point Sincos Data Register 1 (FPSCDT1)

Address(es): TFU.FPSCDT1 0008 1414h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The FPSCDT1 register is used to input an angle value θ and to read the output value of sinθ in a floating-point sincos operation. For details, see Table 40.2. Writing to this register starts a floating-point sincos operation. For how to use this register, see section 40.4.1, Procedures for Using the TFU.
Writing to this register during operation is prohibited. If read access is made to this register during the calculation, the result is read after the calculation is completed. At this time, the bus access is forced to wait until the operation is completed.
Only write to or read from this register in a floating-point sincos operation. Writing to any other data register 1, that is, to the FPATDT1, FXSCDT1, or FXATDT1 register, makes the value of this register undefined. Writing to data saving and restoring register 1 (DTSR1) makes the value of this register the same as that written to the DTSR1 register.

Table 40.2 Input/Output value of FPSCDT0 and FPSCDT1

Register	Input value	Output value
FPSCDT0	Not used	cosθ
FPSCDT1	Angleθ	sinθ

40.2.6 Floating-Point Atanhypot_k Data Register 0 (FPATDT0)

Address(es): TFU.FPATDT0 0008 1418h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The FPATDT0 register is used to input a coordinate value x and to read the output value of $\sqrt{x^2+y^2}/k$ in a floating-point atanhypot_k operation. For details, see Table 40.3. For how to use this register, see section 40.4.1, Procedures for Using the TFU.

Writing to this register during operation is prohibited. If read access is made to this register during the calculation, the result is read after the calculation is completed. At this time, the bus access is forced to wait until the operation is completed.

Only write to or read from this register in a floating-point atanhypot_k operation. Writing to any other data register 0, that is, to the FPSCDT0, FXSCDT0, or FXATDT0 register, makes the value of this register undefined. Writing to data saving and restoring register 0 (DTSR0) makes the value of this register the same as that written to the DTSR0 register.

40.2.7 Floating-Point Atanhypot_k Data Register 1 (FPATDT1)

Address(es): TFU.FPATDT1 0008 141Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The FPATDT1 register is used to input a coordinate value y and to read the output value of atan(y/x) in a floating-point atanhypot_k operation. For details, see Table 40.3. Writing to this register starts a floating-point atanhypot_k operation. For how to use this register, see section 40.4.1, Procedures for Using the TFU.

Writing to this register during operation is prohibited. If read access is made to this register during the calculation, the result is read after the calculation is completed. At this time, the bus access is forced to wait until the operation is completed.

Only write to or read from this register in a floating-point atanhypot_k operation. Writing to any other data register 1, that is, to the FPSCDT1, FXSCDT1, or FXATDT1 register, makes the value of this register undefined. Writing to data saving and restoring register 1 (DTSR1) makes the value of this register the same as that written to the DTSR1 register.

Table 40.3 Input/Output value of FPATDT0 and FPATDT1

Register	Input value	Output value
FPATDT0	Input coordinates x	$\sqrt{x^2+y^2}/k$
FPATDT1	Input coordinates y	atan(y/x)

40.2.8 Fixed-Point Sincos Data Register 0 (FXSCDT0)

Address(es): TFU.FXSCDT0 0008 1420h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The FXSCDT0 register is used to read the output value of cosθ in a fixed-point sincos operation. For details, see Table 40.4. For how to use this register, see section 40.4.1, Procedures for Using the TFU.
Writing to this register during operation is prohibited. If read access is made to this register during the calculation, the result is read after the calculation is completed. At this time, the bus access is forced to wait until the operation is completed.
Only write to or read from this register in a fixed-point sincos operation. Writing to any other data register 0, that is, to the FPSCDT0, FPATDT0, or FXATDT0 register, makes the value of this register undefined. Writing to data saving and restoring register 0 (DTSR0) makes the value of this register the same as that written to the DTSR0 register.

40.2.9 Fixed-Point Sincos Data Register 1 (FXSCDT1)

Address(es): TFU.FXSCDT1 0008 1424h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The FXSCDT1 register is used to input an angle value θ and to read the output value of sinθ in a fixed-point sincos operation. For details, see Table 40.4. Writing to this register starts a fixed-point sincos operation. For how to use this register, see section 40.4.1, Procedures for Using the TFU.
Writing to this register during operation is prohibited. If read access is made to this register during the calculation, the result is read after the calculation is completed. At this time, the bus access is forced to wait until the operation is completed.
Only write to or read from this register in a fixed-point sincos operation. Writing to any other data register 1, that is, to the FPSCDT1, FPATDT1, or FXATDT1 register, makes the value of this register undefined. Writing to data saving and restoring register 1 (DTSR1) makes the value of this register the same as that written to the DTSR1 register.

Table 40.4 Input/Output value of FXSCDT0 and FXSCDT1

Register	Input value	Unit and format of the input values	Output value	Output value formats
FXSCDT0	Not used	—	cosθ	Q1.31, Q2.30, or Q3.29 in accordance with the value of FXSCIOC.OF[1:0] bits
FXSCDT1	Angleθ	FXSCIOC.IUF = 0: turns, Q1.31 FXSCIOC.IUF = 1: radians, Q3.29	sinθ	

Exactly expressing the value +1.0 is not possible when the output value format is Q1.31. When the exact value of sinθ or cosθ should be +1.0 in Q1.31 format, the value read from the FXSCDT0 or FXSCDT1 register is $+1.0 - 2^{-31}$ instead.

40.2.10 Fixed-Point Atanhypot_k Data Register 0 (FXATDT0)

Address(es): TFU.FXATDT0 0008 1428h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The FXATDT0 register is used to input a coordinate value x and to read the output value of $\sqrt{x^2+y^2}/k$ in a fixed-point atanhypot_k operation. For details, see Table 40.5. For how to use this register, see section 40.4.1, Procedures for Using the TFU.

Writing to this register during operation is prohibited. If read access is made to this register during the calculation, the result is read after the calculation is completed. At this time, the bus access is forced to wait until the operation is completed.

Only write to or read from this register in a fixed-point atanhypot_k operation. Writing to any other data register 0, that is, to the FPSCDT0, FPATDT0, or FXSCDT0 register, makes the value of this register undefined. Writing to data saving and restoring register 0 (DTSR0) makes the value of this register the same as that written to the DTSR0 register.

40.2.11 Fixed-Point Atanhypot_k Data Register 1 (FXATDT1)

Address(es): TFU.FXATDT1 0008 142Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The FXATDT1 register is used to input a coordinate value y and to read the output value of atan(y/x) in a fixed-point atanhypot_k operation. For details, see Table 40.5. Writing to this register starts a fixed-point atanhypot_k operation. For how to use this register, see section 40.4.1, Procedures for Using the TFU.

Writing to this register during operation is prohibited. If read access is made to this register during the calculation, the result is read after the calculation is completed. At this time, the bus access is forced to wait until the operation is completed.

Only write to or read from this register in a fixed-point atanhypot_k operation. Writing to any other data register 1, that is, to the FPSCDT1, FPATDT1, or FXSCDT1 register, makes the value of this register undefined. Writing to data saving and restoring register 1 (DTSR1) makes the value of this register the same as that written to the DTSR1 register.

Table 40.5 Input/Output value of FXATDT0 and FXATDT1

Register	Input value	Input value formats	Output value	Unit and format of the output values
FXATDT0	Input coordinates x	Q1.31	$\sqrt{x^2 + y^2}/k$	—, Q3.29
FXATDT1	Input coordinates y	Q1.31	atan(y/x)	FXATIOC.OUF = 0: turns, Q1.31 FXATIOC.OUF = 1: radians, Q3.29

40.2.12 Data Saving and Restoring Register 0 (DTSR0)

Address(es): TFU.DTSR0 0008 1430h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The DTSR0 register is used to save and restore the data in data registers 0 that are FPSCDT0, FPATDT0, FXSCDT0, or FXATDT0 registers. The value read from this register is either that most recently written to any of data registers 0 or the latest result of arithmetic processing in any of data registers 0. Writing to this register makes the values of all data registers 0 the same as that written to this register.

Writing to this register during operation is prohibited. If read access is made to this register during the calculation, the result is read after the calculation is completed. At this time, the bus access is forced to wait until the operation is completed.

For how to use this register, see section 40.4.2, How to Use Interrupts.

40.2.13 Data Saving and Restoring Register 1 (DTSR1)

Address(es): TFU.DTSR1 0008 1434h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The DTSR1 register is used to save and restore the data in data registers 1 that are FPSCDT1, FPATDT1, FXSCDT1, or FXATDT1 registers. The value read from this register is either that most recently written to any of data registers 1 or the latest result of arithmetic processing in any of data registers 1. Writing to this register makes the values of all data registers 1 the same as that written to this register.

Writing to this register during operation is prohibited. If read access is made to this register during the calculation, the result is read after the calculation is completed. At this time, the bus access is forced to wait until the operation is completed.

For how to use this register, see section 40.4.2, How to Use Interrupts.

40.3 Operation

40.3.1 Arithmetic Processing

The trigonometric function unit has two arithmetic operations, the sincos operation and the atanhypot_k operation. Here, “sincos operation” means simultaneous sin and cos calculations and “atanhypot_k operation” means simultaneous atan and hypot_k calculations. For details, see Table 40.6.

Table 40.6 Arithmetic Processing

Operation	Input value	Output value
sincos	Angle value θ	cosθ and sinθ
atanhypot_k	Coordinates x and y	atan(y/x) and $\sqrt{x^2+y^2}/k$

Note: The value of scaling factor *k* is:

$$k = \prod_{i=0}^{\infty} \frac{1}{\sqrt{1+2^{-2i}}} \fallingdotseq 0.607252935008881256169 4$$

Multiplying the output value of the TFU $\sqrt{x^2+y^2}/k$ by k provides $\sqrt{x^2+y^2}$.

40.3.2 Input and Output Value Formats

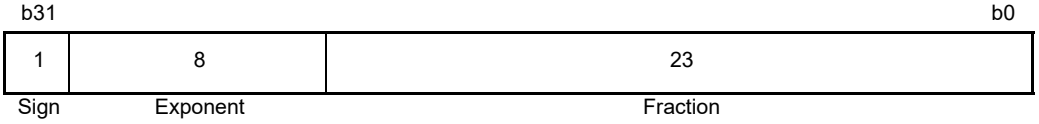
The TFU supports single-precision floating-point numbers and fixed-point numbers as its input and output values. For details on single-precision floating-point numbers and fixed-point numbers, see section 40.3.2.1, Single-Precision Floating-Point Numbers, and section 40.3.2.2, Fixed-Point Numbers, respectively. Note that the precision of fixed-point calculations is lower than 32 bits, although the input and output values are fixed-point numbers in 32-bit format.

40.3.2.1 Single-Precision Floating-Point Numbers

Floating-Point Number

The TFU supports the format for single-precision floating-point numbers specified in the IEEE754 standard.

- Single-Precision Floating-Point Number



Sign: sign field (1-bit)
Exponent: exponent field (8-bit)
Fraction: significand field (23-bit)

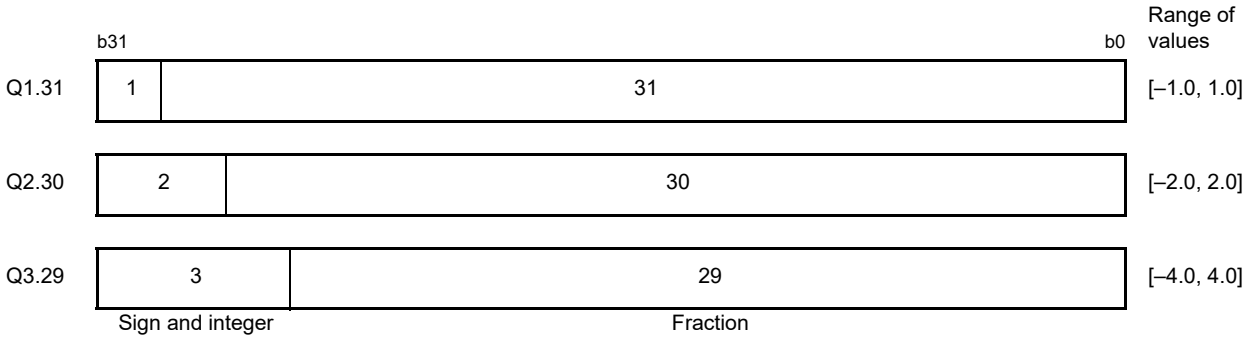
Numerical = $(-1)^S \times (1 + F \times 2^{-23}) \times 2^{(E-127)}$ (normalized value)
Numerical = $(-1)^S \times (0 + F \times 2^{-23}) \times 2^{-126}$ (denormalized value)

The TFU supports the numerical formats of single-precision floating-point numbers listed below.

S	E	F	Numerical
Any value	0 < E < 255	Any value	$(-1)^S \times 1.F \times 2^{(E-127)}$ (Normal Numbers)
Any value	E = 0	F > 0	$(-1)^S \times 0.F \times 2^{-126)}$ (Subnormal Numbers)
S = 0	E = 0	F = 0	$(-1)^0 \times 0.0$ (Positive zero: +0)
S = 1	E = 0	F = 0	$(-1)^{-1} \times 0.0$ (Negative zero: –0)
S = 0	E = 255	F = 0	Positive infinity: +∞
S = 1	E = 255	F = 0	Negative infinity: –∞
Any value	E = 255	400000h > F > 0	SNaN (Signaling Not a Number)
Any value	E = 255	F ≥ 400000h	QNaN (Quiet Not a Number)

40.3.2.2 Fixed-Point Numbers

The TFU supports the two’s-complement formats shown below.



40.3.3 Units of Angles

The TFU supports radians and turns as the units for angles. 0 to 2π radians and 0 to 1 turns respectively represent 0- to 360-degree rotations.

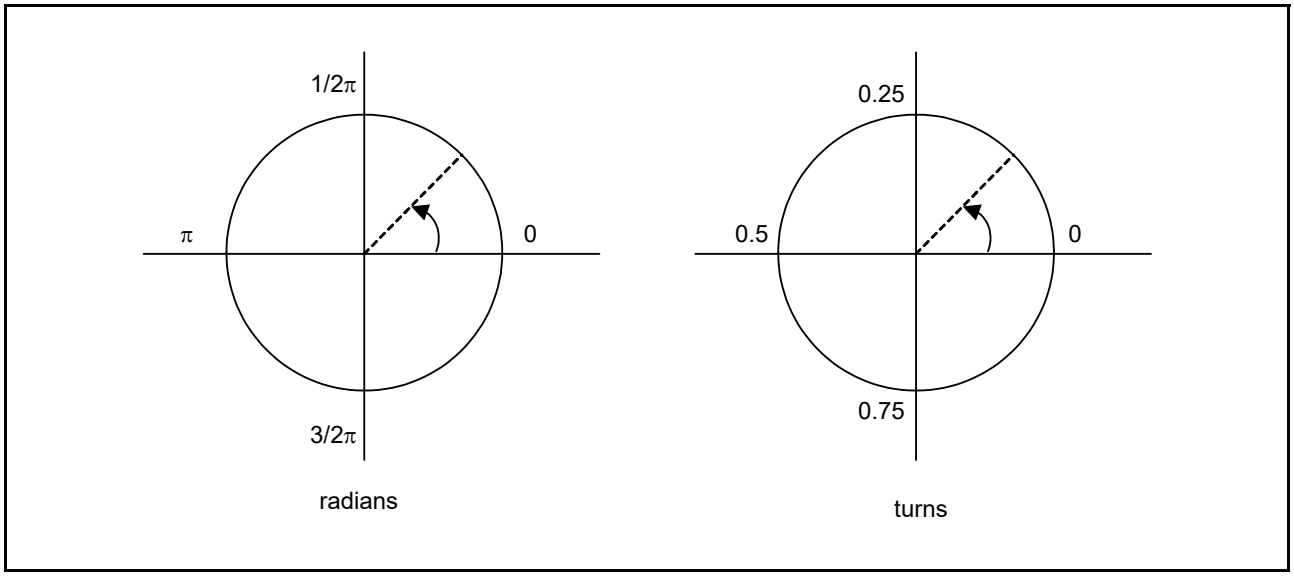


Figure 40.1 Units of Angles

40.3.4 Relationship Between Input and Output Values for Sincos Operation

The output value of a floating-point sin or cos calculation is specified as listed in Table 40.7 when the input value is ±0, ±∞, SNaN (Signaling Not a Number), or QNaN (Quiet Not a Number).

Table 40.7 Relationship Between Special Input Value and Its Output Value (for Sincos Operation)

Input (*θ*)	Output (cos)	Output (sin)
–∞	QNaN	QNaN
–0	+1	–0
+0	+1	+0
+∞	QNaN	QNaN
SNaN/QNaN	QNaN	QNaN

Note: The output value of QNaN is FFC0 0000h.

40.3.5 Relationship Between Input and Output Values for Atan Operation

The output value of a floating-point atan calculation is specified as listed in Table 40.8 when either of the input values x or y is ±0, ±∞, SNaN (Signaling Not a Number), or QNaN (Quiet Not a Number). An input error is detected when both of the input values x and y are ±0.

Table 40.8 Relationship Between Special Input Value and Its Output Value (for Floating-Point Atan Operation)

y \ x	−∞	Negative value	−0	+0	Positive value	+∞	SNaN/QNaN
−∞	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN
Negative value	QNaN	—	−π/2	−π/2	—	QNaN	QNaN
−0	QNaN	−π	QNaN*1	QNaN*1	−0	QNaN	QNaN
+0	QNaN	+π	QNaN*1	QNaN*1	+0	QNaN	QNaN
Positive value	QNaN	—	+π/2	+π/2	—	QNaN	QNaN
+∞	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN
SNaN/QNaN	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN

Note 1. An input error occurs for the special input value, and the input error flag (TRGSTS.ERRF) is set.

An input error is detected when both of the input values x and y of a fixed-point atan calculation are 0, even for the values other than 0 in 32-bit fixed-point format that are handled as 0 by the TFU. The output value of an atan calculation is 0 when an input error has occurred.

40.3.6 Relationship Between Input and Output Values for hypot_k Operation

The output value of a floating-point hypot_k calculation is specified as listed in Table 40.9 in either of the following cases: when either of the input values x or y is ±∞, SNaN (Signaling Not a Number), or QNaN (Quiet Not a Number), or when both of the input values x and y are ±0. An input error is detected when both of the input values x and y are ±0. The reason is that such values cause errors in atan calculations although hypot_k calculations are possible.

Table 40.9 Relationship Between Special Input Value and Its Output Value (for hypot_k Operation)

y \ x	−∞	Negative value	−0	+0	Positive value	+∞	SNaN/QNaN
−∞	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN
Negative value	QNaN	—	—	—	—	QNaN	QNaN
−0	QNaN	—	+0*1	+0*1	—	QNaN	QNaN
+0	QNaN	—	+0*1	+0*1	—	QNaN	QNaN
Positive value	QNaN	—	—	—	—	QNaN	QNaN
+∞	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN
SNaN/QNaN	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN

Note 1. An input error occurs for the special input value, and the input error flag (TRGSTS.ERRF) is set.

An input error is detected when both of the input values x and y of a fixed-point hypot_k calculation are 0, even for the values other than 0 in 32-bit fixed-point format that are handled as 0 by the TFU. The reason is that such values cause errors in atan calculations although hypot_k calculations are possible. The output value of a hypot_k calculation is 0 when an input error has occurred.

40.4 Usage Procedures

40.4.1 Procedures for Using the TFU

There are two procedures for reading the result of an operation: reading the data register after the operation is completed (Procedure 1) and reading the data register without waiting for the operation to be completed (Procedure 2). Table 40.10 shows the advantages and disadvantages of each procedure. Figure 40.2 shows the procedures for sincos operations and Figure 40.3 shows the procedures for atanhypot_k operations. Read SCDTn (n = 0, 1) in Figure 40.2 as FPSCDTn and FXSCDTn in the case of floating-point sincos operations and fixed-point sincos operations, respectively. Read ATDTn in Figure 40.3 as FPATDTn and FXATDTn in the case of floating-point atanhypot_k operations and fixed-point atanhypot_k operations, respectively.

Table 40.10 Advantages and Disadvantages of Procedures for Sincos and Atanhypot_k Operations

Method	Advantages	Disadvantages
Procedure 1: reading the data register after the operation is completed	The bus is not occupied.	Detecting completion of a calculation by checking the TRGSTS.BSYF flag is required.
Procedure 2: reading the data register without waiting for the operation to be completed	Detecting completion of a calculation by checking the TRGSTS.BSYF flag is not required, resulting in fewer cycles being required for execution.	The bus is occupied until completion of the calculation.

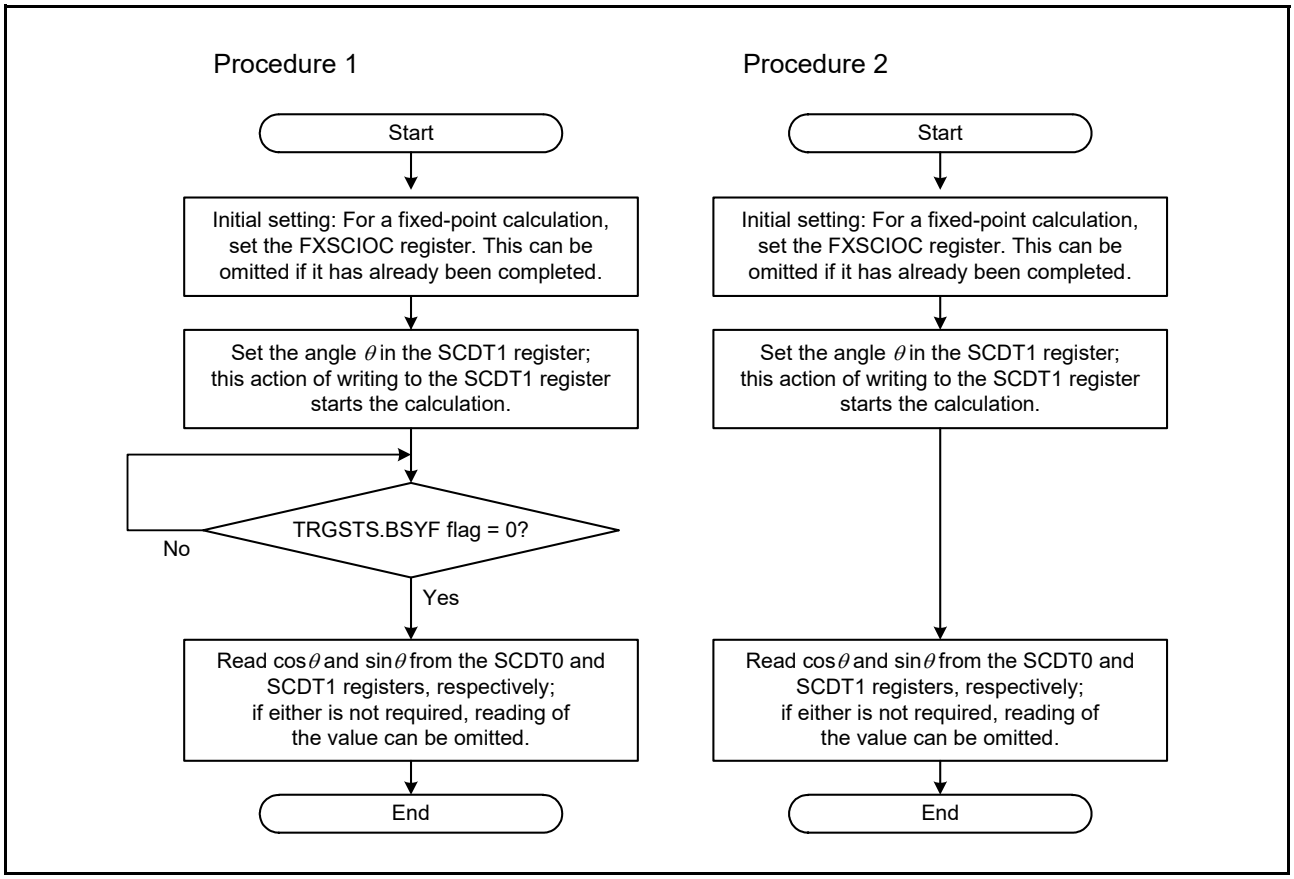


Figure 40.2 Procedure for Using TFU (Sincos Operation)

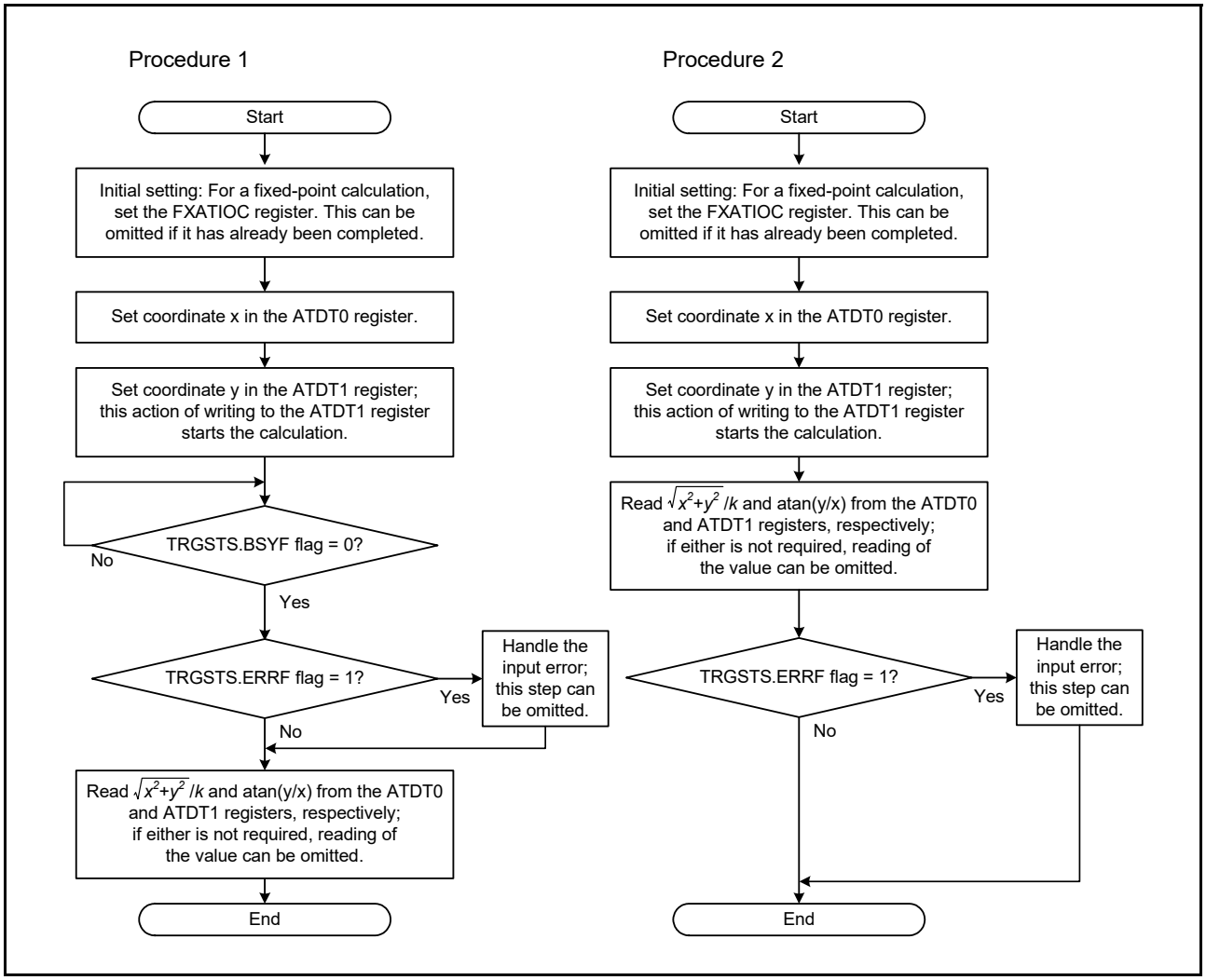


Figure 40.3 Procedure for Using TFU (Atanhypot_k Operation)

40.4.2 How to Use Interrupts

If an interrupt is accepted in the following order, the values stored in the data registers and the error flag prior to the acceptance of the interrupt may become incorrect in terms of the software after execution is returned from the interrupt handler.

1. The interrupt is accepted between writing to the data registers and reading of the data registers and the error flag during any of the procedures described in section 40.4.1, Procedures for Using the TFU.
2. The TFU is used to execute any calculation in the handler.

Accordingly, avoid such incorrect operation when, for example, using the TFU with multiple interrupts and their handlers by taking either of the following measures.

(1) Disabling interrupts that otherwise may disrupt processing between writing to the data registers and reading of the data registers and the error flag

(2) Saving and restoring data registers and the error flag in interrupt handlers that may disrupt processing*1, *2

Note 1. Regardless of the type of operation executed in the interrupt handler or the type of operation executed before accepting an interrupt, use the DTSR0, DTSR1, and TRGSTS registers to save and restore data registers and error flags. Figure 40.4 shows the save/restore procedure in an interrupt handler. By following this procedure, after returning from the interrupt handler, the processing that was being performed before accepting the interrupt can be continued without malfunctioning.

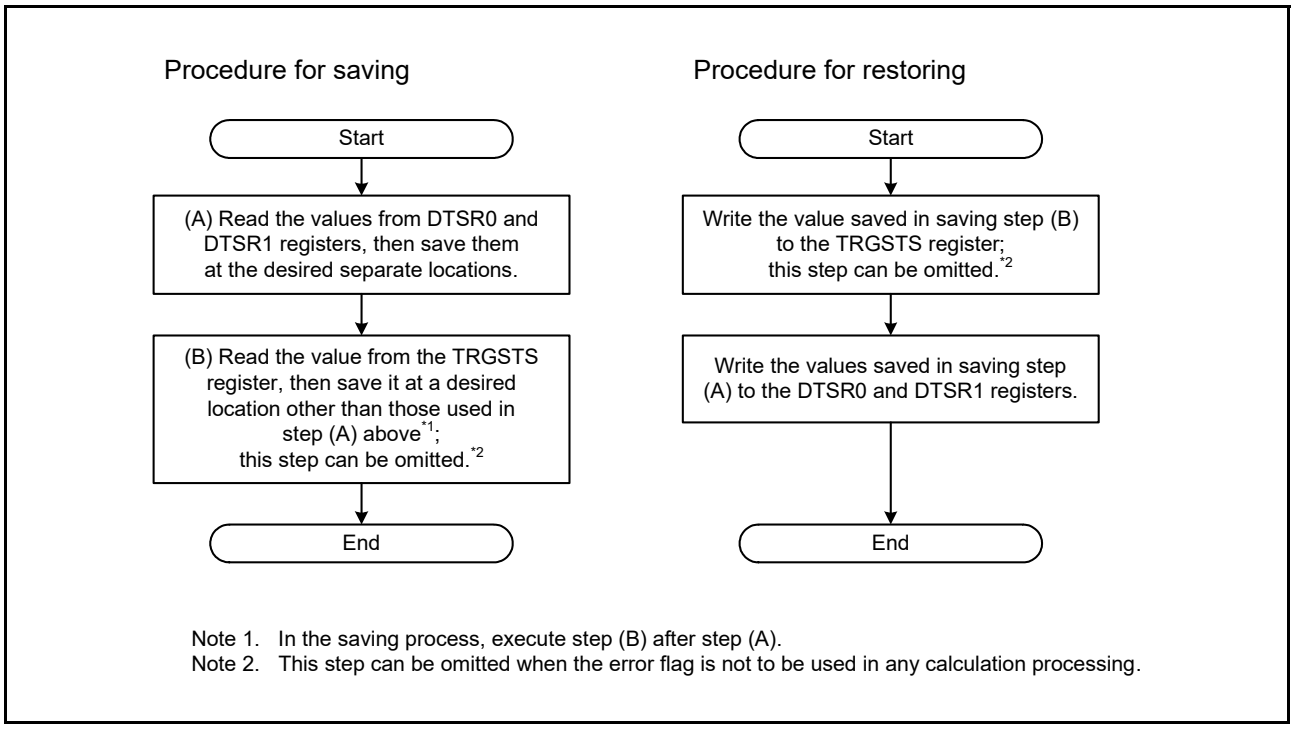


Figure 40.4 Procedures for Saving and Restoring in Interrupt Handlers

Note 2. As well as the data registers and the error flag, interrupt handlers that may change the values of the setting registers (FXSCIOC and FXATIOC) also require saving and restoring of the setting register values. Accordingly, save and restore the values of the setting registers as required. Saving of the setting registers must be executed after step (A) of the saving process in Figure 40.4.

41. Trusted Secure IP (TSIP-Lite)

This MCU incorporates a Trusted Secure IP Lite (TSIP-Lite) module to provide security functions. The module consists of an access management circuit, encryption engine, and random number generator. In combination with the TSIP-Lite library, the TSIP-Lite can prevent eavesdropping (confidentiality), falsification of information (integrity), and impersonation (authenticity).
Key information to be used in encrypting and decrypting data is only stored within the TSIP-Lite, and any external access can be shut out to obtain a system with strong security.

41.1 Overview

Table 41.1 summarizes the specifications of the TSIP-Lite. Figure 41.1 shows a block diagram of the TSIP-Lite.

Table 41.1 Specifications of TSIP-Lite

Item	Description
Access control	Access management circuit • In case of irregular access to the TSIP-Lite due to a falsified program or runaway execution of a program, this circuit blocks all subsequent access and stops the output of data from the TSIP-Lite.
Encryption engine	AES: Compliant with NIST FIPS PUB 197 algorithm • Key sizes: 128 or 256 bits • Block sizes: 128 bits • Block cipher mode of operation ECB, CBC, CTR: Compliant with NIST SP 800-38A CMAC: Compliant with NIST SP 800-38B CCM: Compliant with NIST SP 800-38C GCM: Compliant with NIST SP 800-38D XTS: Compliant with NIST SP 800-38E GCTR • Number of cycles for execution*1 ECB, CBC, CTR, CMAC, GCTR, XTS: 44 cycles of PCLKB for 128-bit keys, 61 cycles of PCLKB for 256-bit keys CCM: 88 cycles of PCLKB for 128-bit keys AES-GCM • AES-GCM is realized by combining AES-GCTR and GHASH. Key management • Keys are only valid within the TSIP-Lite. • Only key generation information is output from the TSIP-Lite. • Keys can be regenerated by the input of key generation information to the TSIP-Lite. Endian • Big or little
Generation of random numbers	32-bit true random number generator • The TSIP-Lite library can assemble 32-bit true random numbers to generate 128- or 256-bit true random numbers. • The generated 128-bit and 256-bit true random numbers are used as keys in encrypting and decrypting data.
Protection against illicit key copying	• An ID unique to the MCU (unique ID) is accessible from the access management circuit through the dedicated bus. • Combining the unique ID with the key generation information prevents the illicit copying of the key to another MCU.
Supervisor mode	• The supervisor mode signal is connected to the access management circuit and is used to limit control of the TSIP-Lite module to supervisor mode only.
Interrupt sources	Three These can be used as triggers for data transfer by the DMAC or DTC.
Low power consumption	Setting of the module stop state is possible.

Note 1. This does not include the overhead for calling functions of the TSIP-Lite library.

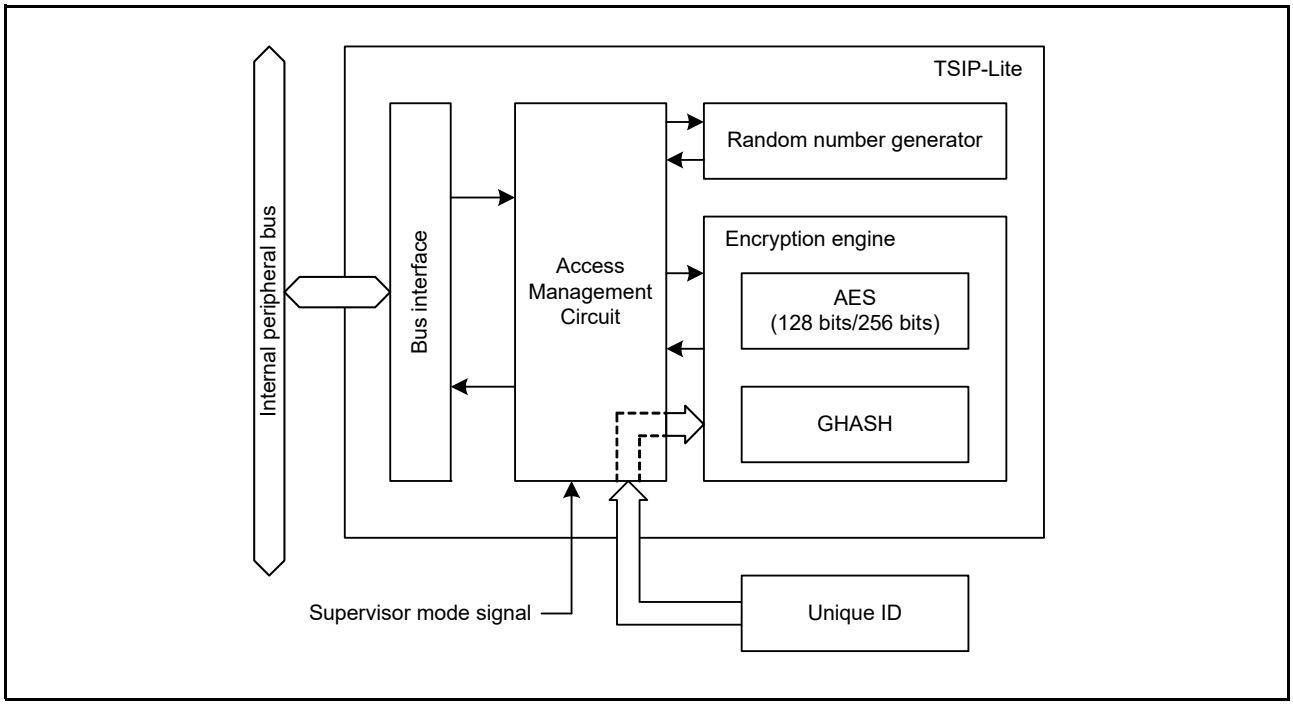


Figure 41.1 TSIP-Lite Block Diagram

41.2 Operation

41.2.1 Operating Modes and State Transitions

Figure 41.2 shows the state transitions of the TSIP-Lite.
Use of the TSIP-Lite security functions is only possible through use of the TSIP-Lite library provided by Renesas Electronics, in accordance with the state transitions as shown in the figure below.
When irregular access to the TSIP-Lite (access that violates the defined procedure) due to a falsified program or a program entering runaway execution, etc. is attempted, the access management circuit does not accept any subsequent access and stops the output of any data from the TSIP-Lite.

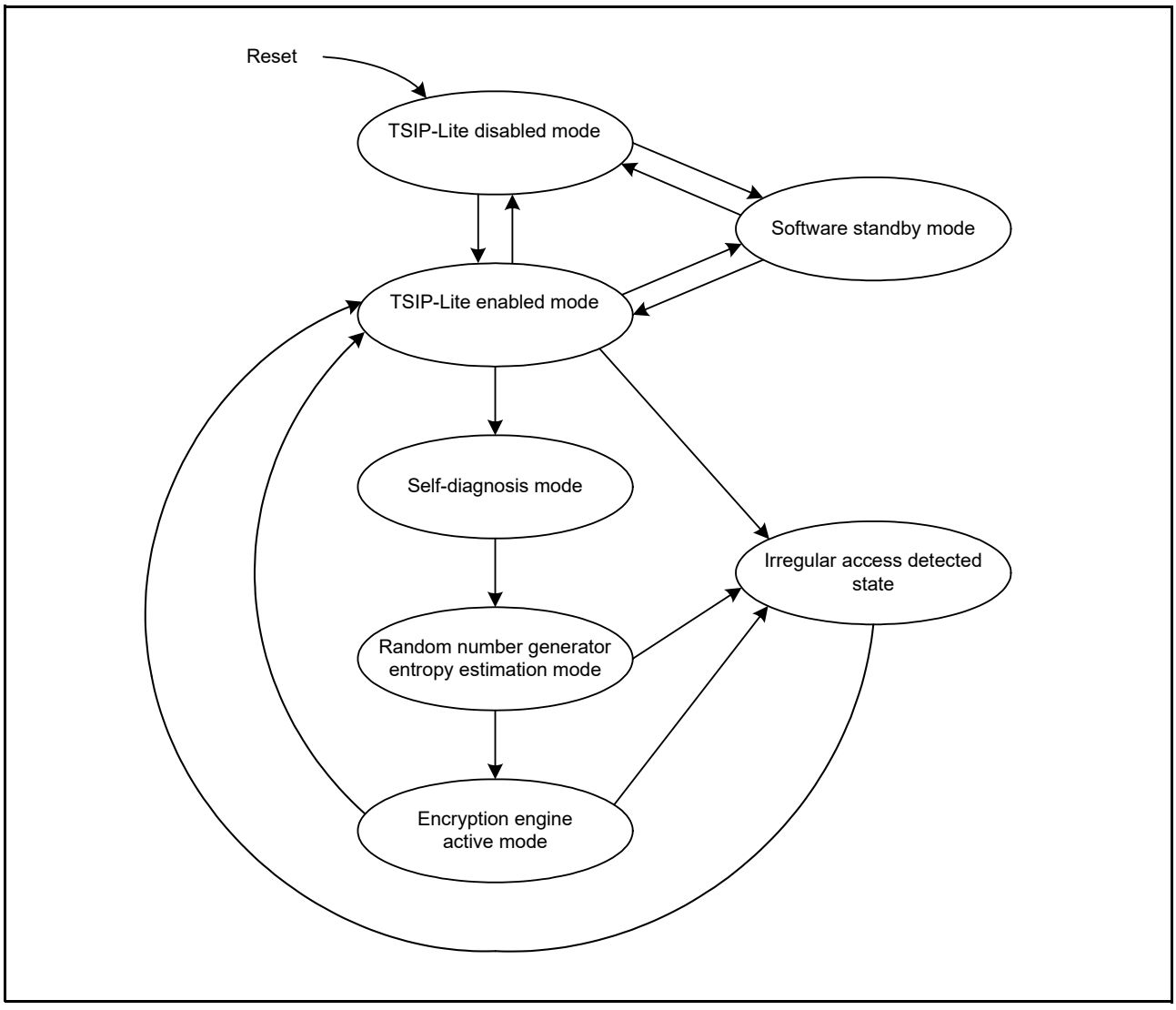


Figure 41.2 TSIP-Lite Operating Modes and State Transitions

Many of the security functions that the TSIP-Lite offers are applicable only in the encryption engine active mode. The operations that can be performed in this mode are given below.

(1) Key Installation
(2) Encryption and decryption
(3) Key generation
(4) Random number generation

41.2.2 Encryption Engine

Figure 41.3 shows processes of the encryption engine integrated in the TSIP-Lite.
The encryption engine, using the key generation information, performs plaintext to ciphertext encryption and ciphertext to plaintext decryption by hardware.
In no part of the encryption or decryption process, is key data or intermediate data ever exposed outside of the TSIP-Lite.

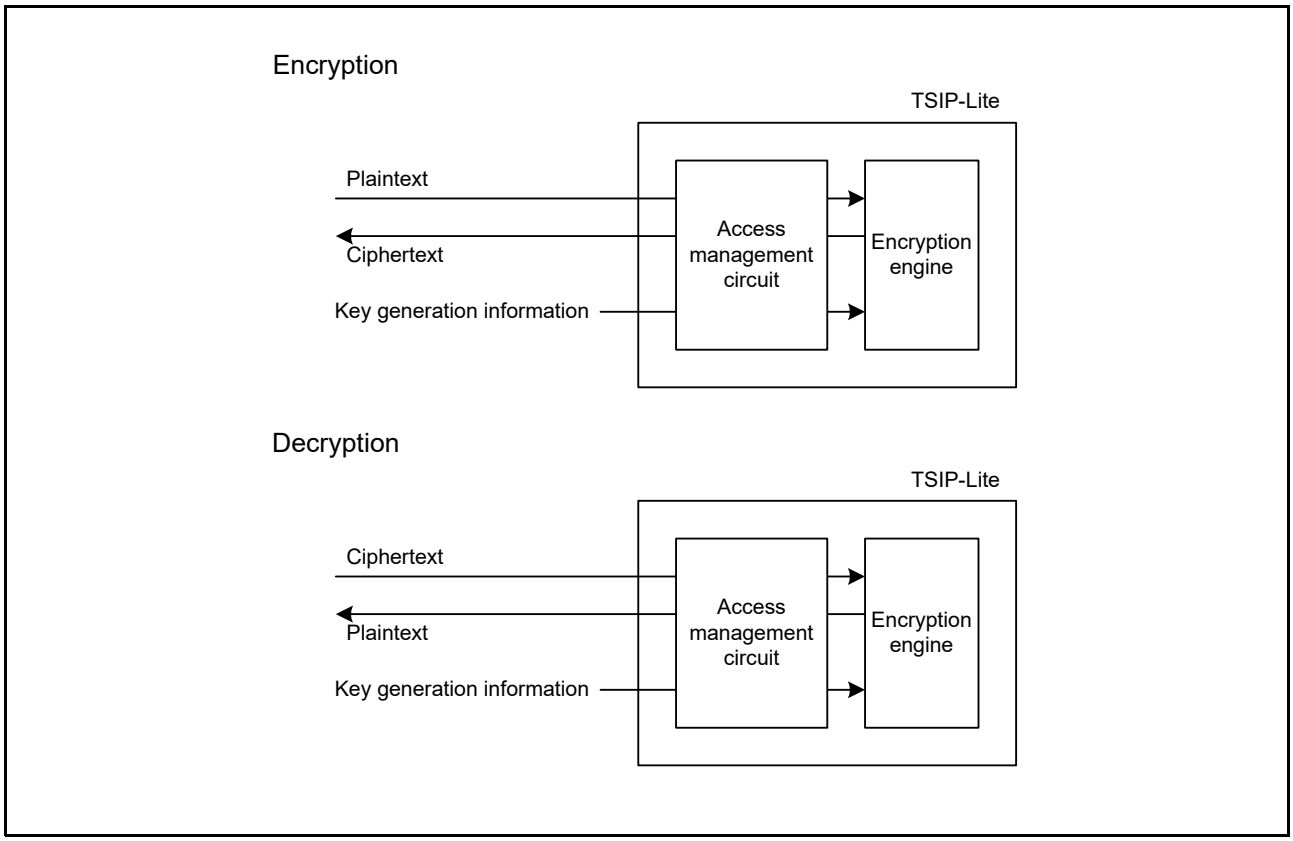


Figure 41.3 Encryption and Decryption processes by Encryption Engine

41.2.3 Key Installation

The key installation is the operation that safely converts the user key to the key generation information and stores it in flash memory. The procedure for installing the key data are given below.

(1) The user uses the key (Key-2) used for encrypting the user key to encrypt the user key (Key-1) producing eKey-1.
(2) The user sends the encrypted user key (eKey-1) to the TSIP-Lite over the serial interface.
(3) The key generation information of the Key-2 (Index-2) contained in the TSIP-Lite library is used to recover the Key-2, which is then used to decrypt the user key.
(4) The user key is converted to user key generation information (Index-1) using the unique ID and a random number, and stored in flash memory.

The installation process and flow chart are given in Figure 41.4 and Figure 41.5, respectively.
Once the key data is installed, the user key generation information (Index-1) can then be used to perform encryption or decryption.

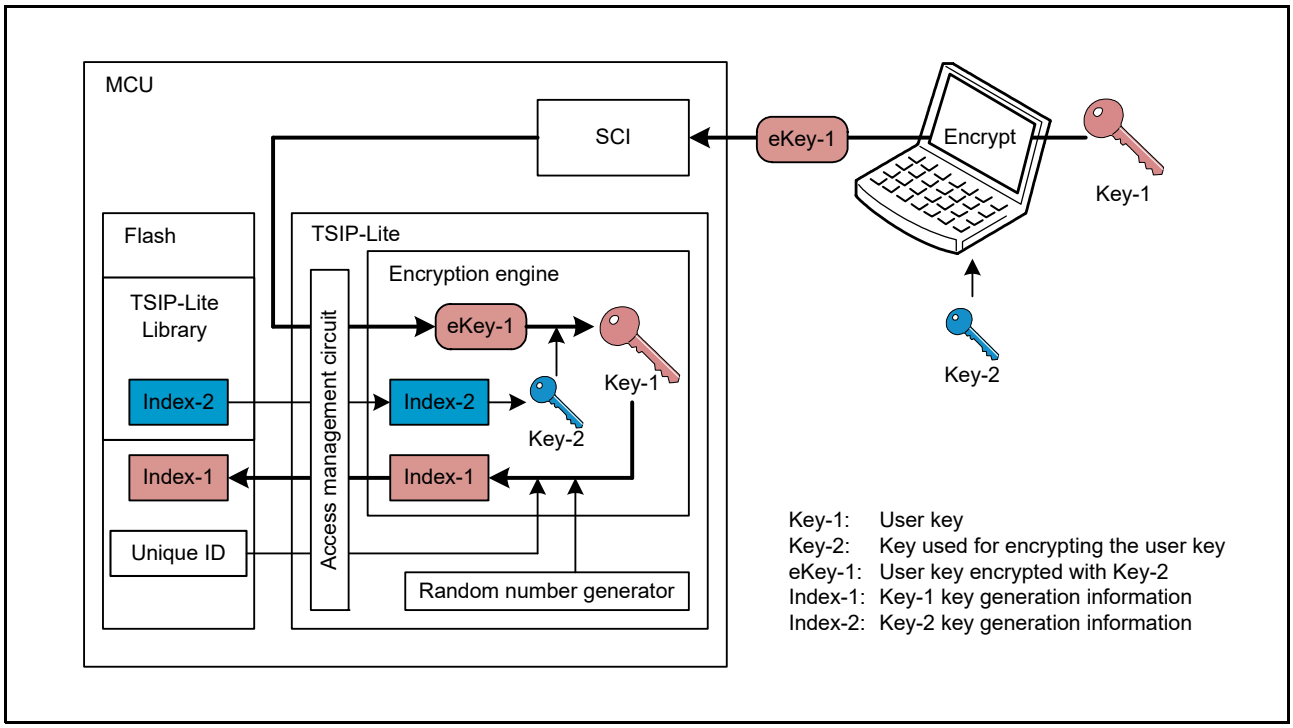


Figure 41.4 Key Installation Process

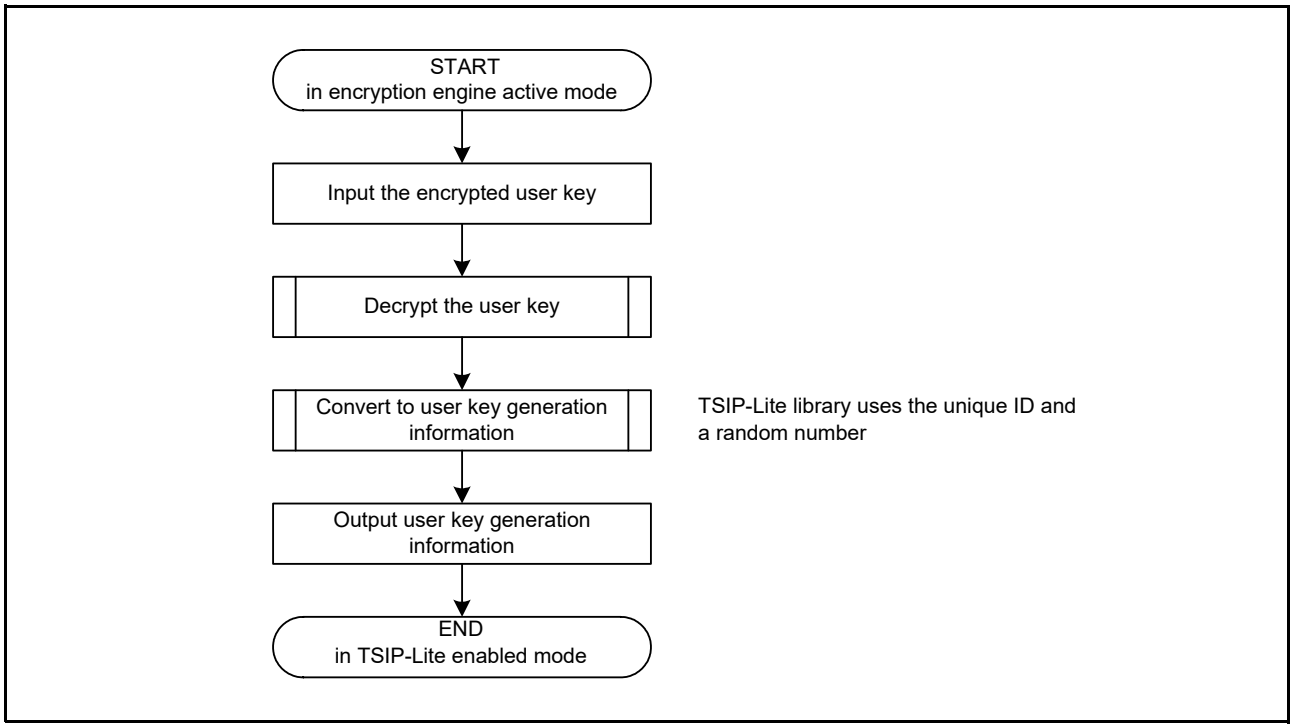


Figure 41.5 Key Installation Flow Chart

41.2.4 Encryption and Decryption

The procedures for encrypting and decrypting data are given below.

(1) Input the key generation information into the TSIP-Lite, and recover the key data.
(2) Input the data to encrypt or decrypt into the TSIP-Lite. This converts plaintext into ciphertext, and ciphertext into plaintext.
(3) Read the converted data.

The encryption engine has an input buffer and an output buffer, enabling encryption/decryption to proceed in parallel with data input/output.
Figure 41.6, Figure 41.7, and Figure 41.8 show the timing diagram, encryption flow, and decryption flow, respectively.

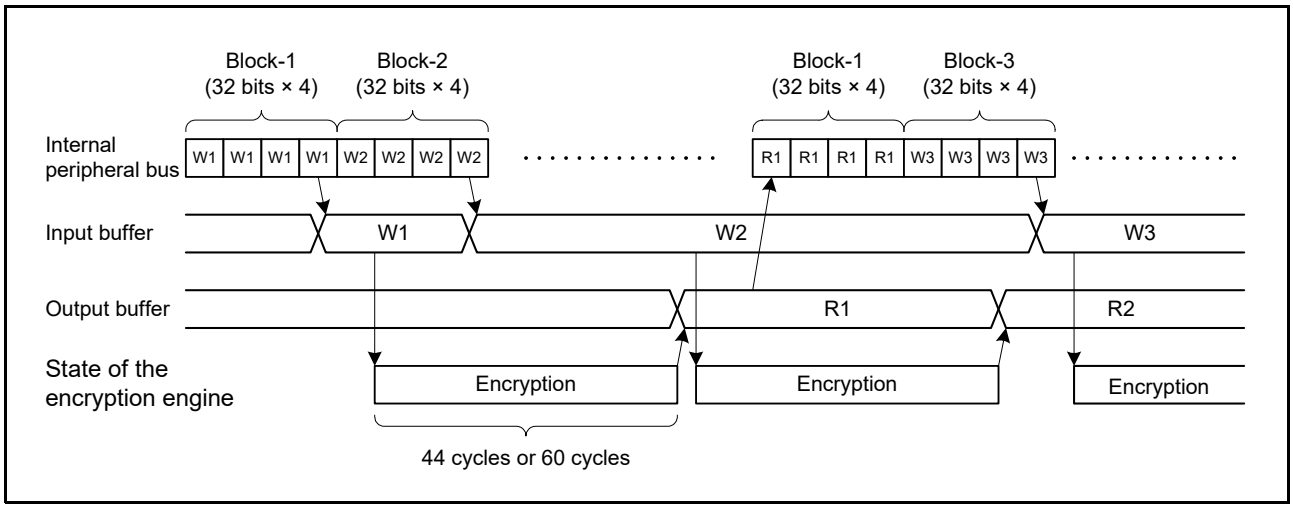


Figure 41.6 Encryption and Decryption Timing Diagram

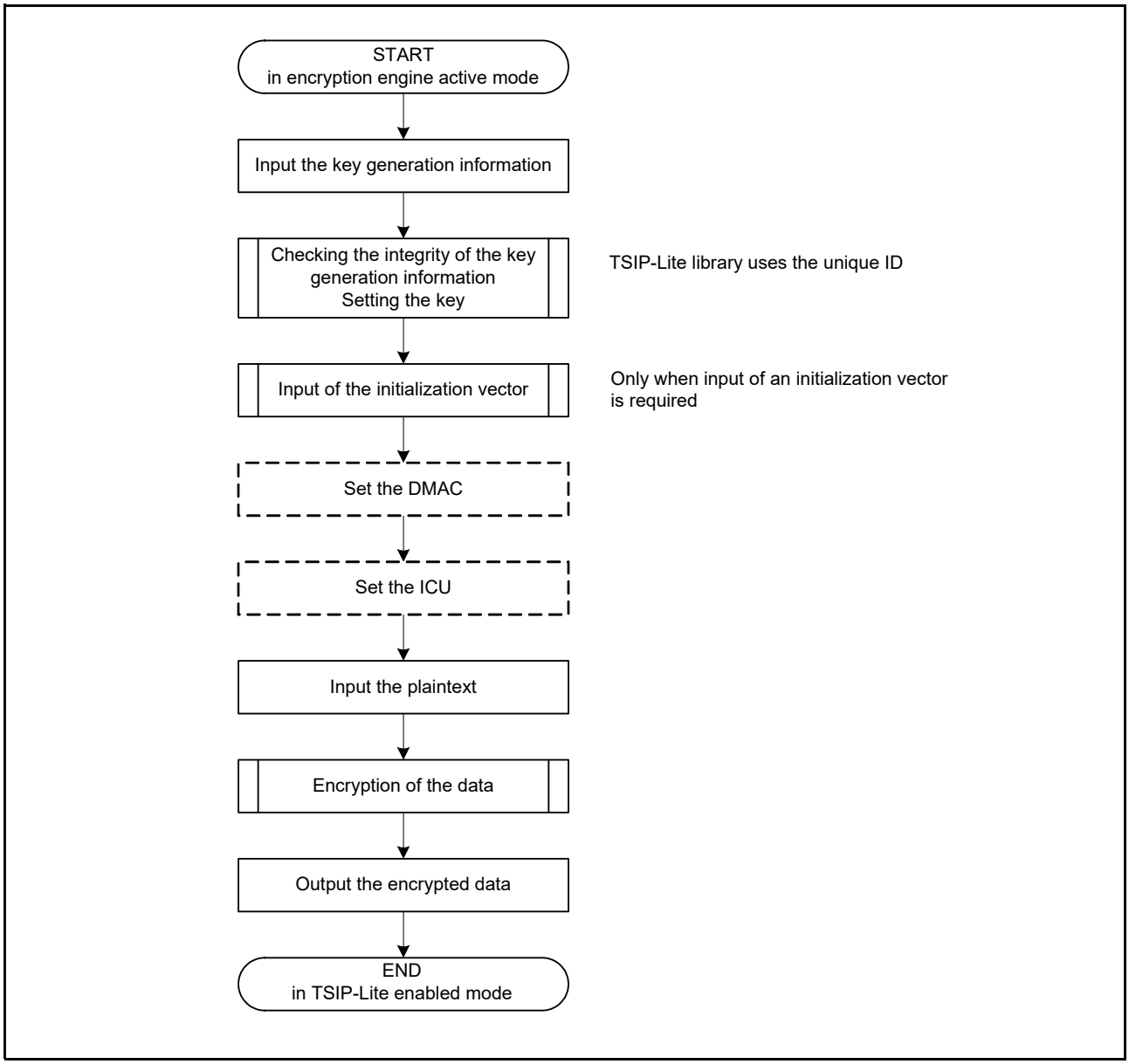


Figure 41.7 Encryption Flow Chart

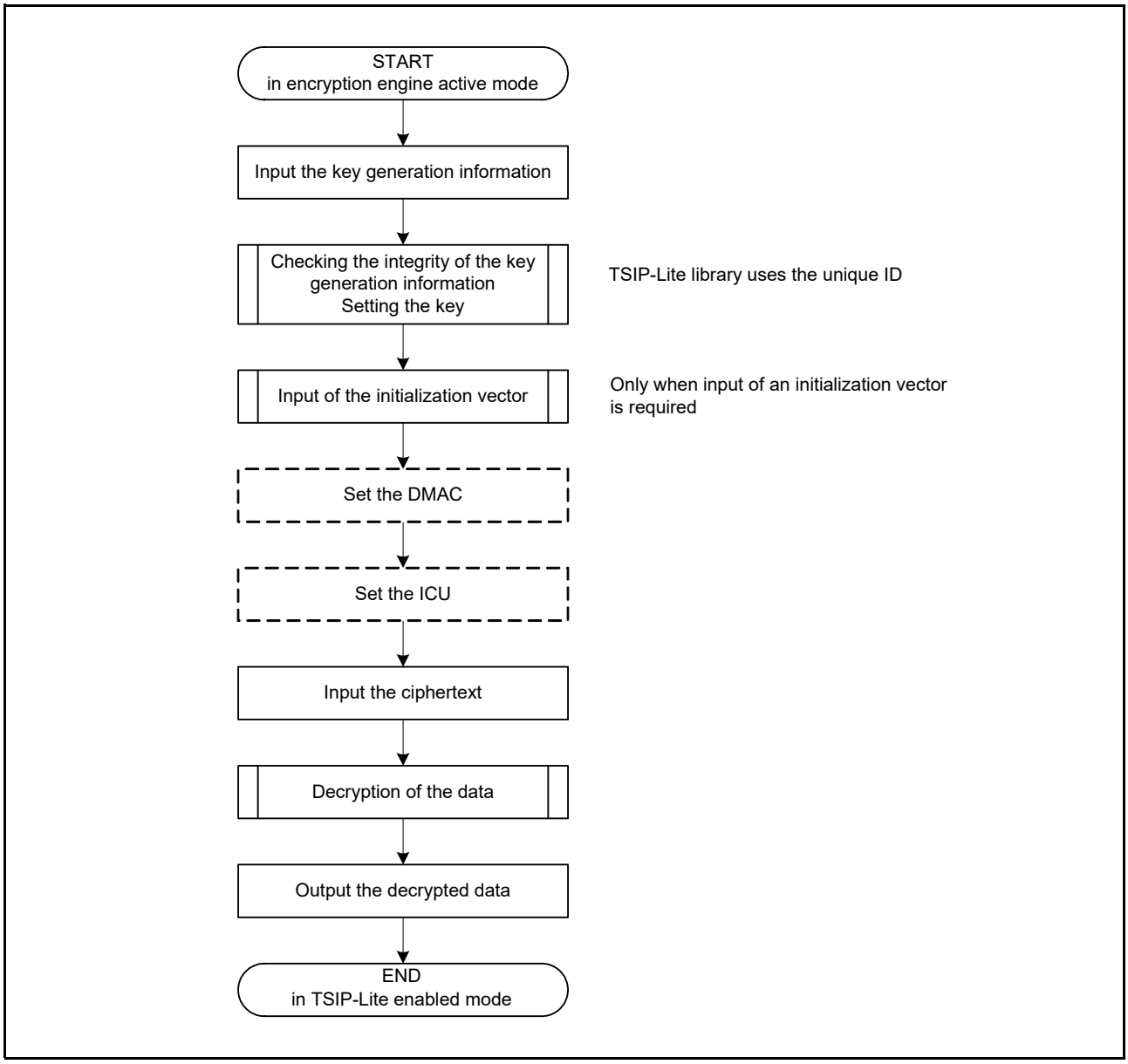


Figure 41.8 Decryption Flow Chart

41.2.5 Generating Key Generation Information (by Using Random Numbers)

Figure 41.9 shows the generating flow for the key generation information by using random numbers.

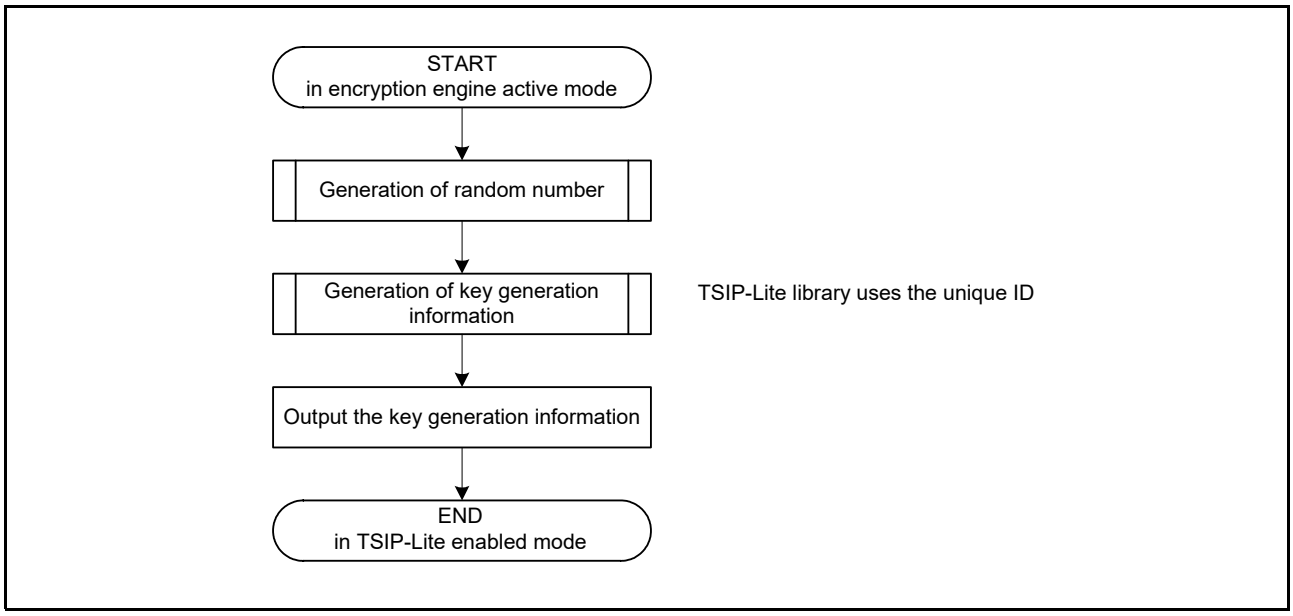


Figure 41.9 Key Generation Information Generating Flow Chart (Using Random Numbers)

41.2.6 Random Number Generation

Figure 41.10 shows the random number generation flow.

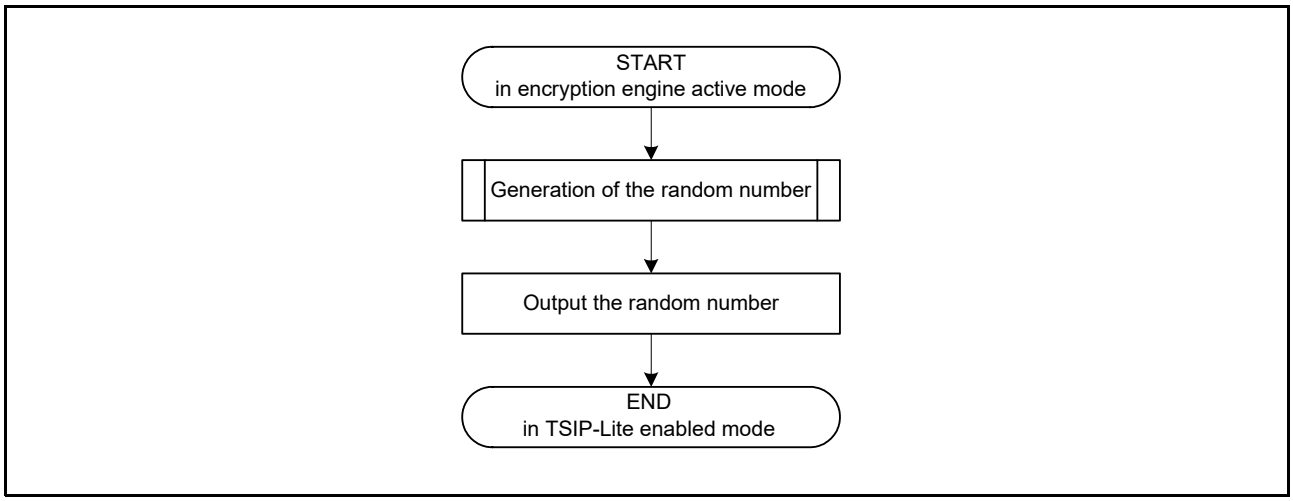


Figure 41.10 Random Number Generation Flow Chart

41.3 Interrupt

Table 41.2 lists the interrupt sources.
TSIP-Lite library uses interrupts caused by these interrupt sources. Do not set the ICU.IERm.IENj bits corresponding to these interrupt sources to 0.

Table 41.2 TSIP-Lite Interrupt Sources

Name	Interrupt Source	DTC Triggerable	DMAC Triggerable
RD	Data read ready	Yes	Yes
WR	Data write ready	Yes	Yes
Error	Illegal access detected	No	No

41.4 Usage Notes

41.4.1 Standby Mode

When standby mode is entered while the encryption engine is in processing, proper processing cannot be resumed after standby mode is exited. Standby mode should therefore be entered only after first entering TSIP-Lite disabled mode or TSIP-Lite enabled mode.

41.4.2 Setting the Module Stop Function

The module stop control register D (MSTPCRD) enables or disables operation of the TSIP-Lite. After a reset, the TSIP-Lite is stopped. After exiting the module stop state, the TSIP-Lite can be accessed. Refer to section 11, Low Power Consumption for details.

41.4.3 TSIP-Lite Library

Use of the TSIP-Lite requires the TSIP-Lite library provided by Renesas Electronics. Please contact our sales office for information regarding the TSIP-Lite library.

42. 12-Bit A/D Converter (S12ADHa)

42.1 Overview

This MCU has up to three units of 12-bit successive approximation A/D converters (S12AD, S12AD1, S12AD2).
The 12-bit A/D converter converts analog input of the selected channels, temperature sensor output, or internal reference voltage into a 12-bit digital value through successive approximation.
The A/D converter has three operating modes: single scan mode in which the arbitrarily selected analog inputs are converted in ascending channel order; continuous scan mode in which the arbitrarily selected analog inputs are continuously converted in ascending channel order; and group scan mode in which the arbitrarily selected analog inputs are arbitrarily divided into two groups (groups A and B) or three groups (groups A, B, and C) and converted in ascending channel order in each group.
In group scan mode, either two groups (groups A and B) or three groups (groups A, B, and C) is selected.
The conditions for scanning start of each group (A, B, and C) (synchronous trigger) can be independently selected, thus allowing A/D conversion of each group to be started independently.
During group priority operation, in addition to the above-mentioned operation, a trigger to start scanning for the priority group is accepted during scan for the low-priority group, and scan for the priority group is started after scan for the low-priority group was interrupted. The priority order is group A > group B > group C. Accordingly, as priority operation, when a trigger to start scanning for group B is accepted during scan for group C, group C scan is interrupted, and scan for group B is started. Likewise, when a trigger to start scanning for group A is accepted during scan for group C, group C scan is interrupted and scan for group A is started. In the same way, when a trigger to start scanning for group A is accepted during scan for group B, group B scan is interrupted and scan for group A is started.
The interrupted scan operation can be restarted after the scanning of the priority group is completed.
In double trigger mode, one analog input channel arbitrarily selected is converted in single scan mode or group scan mode (group A), and the resulting data of A/D conversion started by the first and second synchronous triggers are stored into different registers (duplication of A/D conversion data).
Self-diagnosis is executed once at the beginning of each scan, and one of the three voltages internally generated in the 12-bit A/D converter is converted.
The temperature sensor output or internal reference voltage can be selected in single scan mode and group scan mode. A/D conversion is performed on the temperature sensor output prior to internal reference voltage when both the temperature sensor output and internal reference voltage are selected.
The temperature sensor output or internal reference voltage cannot be selected in continuous scan mode.
The MCU incorporates the comparison function (with windows A and B). In addition, the value of A/D conversion and the reference value of the low side can be compared by a comparator.
Table 42.1 lists the specifications of the 12-bit A/D converter and Table 42.2 lists the functions of the 12-bit A/D converter. Figure 42.1 to Figure 42.4 show block diagrams of the 12-bit A/D converter.

Table 42.1 Specifications of 12-Bit A/D Converter (1/2)

Item	Description
Number of units	Three units (S12AD, S12AD1, and S12AD2) (products with 64 Kbytes of RAM) Two units (S12AD and S12AD2) (products with 48 Kbytes of RAM)
Input channels	Four channels for S12AD, four channels for S12AD1, and 14 channels for S12AD2 (products with 64 Kbytes of RAM) Seven channels for S12AD and eight channeld for S12AD2 (products with 48 Kbytes of RAM)
Extended analog function	Temperature sensor output, internal reference voltage (S12AD2 only)
A/D conversion method	Successive approximation method
Resolution	12 bits
Conversion time	0.9 µs per channel (when A/D conversion clock ADCLK = 60 MHz)
A/D conversion clock	Peripheral module clock PCLKB*1 and A/D conversion clock ADCLK*1 can be set so that the frequency ratio should be one of the following. PCLKB to ADCLK frequency ratio = 1:1, 2:1, 4:1, 1:2 ADCLK is set using the clock generation circuit. A/D conversion clock (ADCLK) can operate between 8 MHz at a minimum and 60 MHz at a maximum.
Data registers	• One register per channel for analog input, one register per unit for A/D-converted data duplication in double trigger mode, and two registers per unit for A/D-converted data duplication during extended operation in double trigger mode. • One register for temperature sensor (S12AD2) • One register for internal reference (S12AD2) • One register per unit for self-diagnosis • The results of A/D conversion are stored in 12-bit A/D data registers. • The value obtained by adding up A/D-converted results is stored as a value in the number of bit for conversion accuracy + 2 bits/4 bits*2 in the A/D data registers in A/D-converted value addition mode. • Double trigger mode (selectable in single scan and group scan modes): The first piece of A/D-converted analog-input data on one selected channel is stored in the data register for the channel, and the second piece is stored in the duplication register. • Extended operation in double trigger mode (available for specific triggers): A/D-converted analog-input data on one selected channel is stored in the duplication register that is prepared for each type of trigger.
Operating modes	Operating modes can be set individually for each unit. • Single scan mode: A/D conversion is performed only once on the analog inputs arbitrarily selected. A/D conversion is performed only once on the temperature sensor output (S12AD2). A/D conversion is performed only once on the internal reference voltage (S12AD2). • Continuous scan mode: A/D conversion is performed repeatedly on the analog inputs arbitrarily selected. • Group scan mode: Two (groups A and B) or three (groups A, B, and C) can be selected as the number of the groups to be used. (Only the combination of groups A and B can be selected when the number of the groups is two.) Analog inputs, temperature sensor output (S12AD2), and internal reference voltage (S12AD2) that are arbitrarily selected are divided into two groups (group A and B) or three groups (group A, B, and C), and A/D conversion of the analog input selected on a group basis is performed only once. The conditions for scanning start of groups A, B, and C (synchronous trigger) can be independently selected, thus allowing A/D conversion of each group to be started independently. • Group scan mode (when group priority control selected): If a priority-group trigger is input during scanning of the low-priority group, scan of the low-priority group is stopped and scan of the priority group is started. The priority order is group A (highest) > group B > group C (lowest). Whether or not to restart scanning of the low-priority group after processing for the high-priority group completes, is selectable. Rescan can also be set to start either from the beginning of the selected channel or the channel on which A/D conversion is not completed.
Conditions for A/D conversion start	• Software trigger • Synchronous trigger Trigger by the multi-function timer pulse unit (MTU), general PWM timer (GPTW), 8-bit timer (TMR), or event link controller (ELC). • Asynchronous trigger A/D conversion can be triggered by the external trigger ADTRG0# (S12AD), ADTRG1# (S12AD1), or ADTRG2# (S12AD2) pin (individually for each unit).

Table 42.1 Specifications of 12-Bit A/D Converter (2/2)

Item	Description
Functions	• Channel-dedicated sample-and-hold function (three channels each for S12AD and S12AD1) (Constant sampling can be set) • Variable sampling time (can be set per channel) • Self-diagnosis of 12-bit A/D converter • Selectable A/D-converted value addition mode or average mode • Analog input disconnection detection assist function (discharge function/precharge function) • Double trigger mode (duplication of A/D conversion data) • Automatic clear function of A/D data registers • Comparison function (windows A and B) • Order of channel conversion in each unit can be set. • Input signal amplification function by programmable gain amplifier (three channels each for S12AD and S12AD1) (only for products with 64 Kbytes of RAM)
Interrupt sources	• In the modes except double trigger mode and group scan mode, a scan end interrupt request (S12ADI, S12ADI1, or S12ADI2) can be generated on completion of single scan (individually for each unit). • In double trigger mode, a scan end interrupt request (S12ADI, S12ADI1, or S12ADI2) can be generated on completion of double scan (individually for each unit). • In group scan mode, a scan end interrupt request (S12ADI, S12ADI1, or S12ADI2) can be generated on completion of group A scan, whereas a group B scan end interrupt request (S12GBADI, S12GBADI1, or S12GBADI2) can be generated on completion of group B scan, and a group C scan end interrupt request (S12GCADI, S12GCADI1, or S12GCADI2) can be generated on completion of group C scan. • When double trigger mode is selected in group scan mode, an A/D scan end interrupt request (S12ADI, S12ADI1, or S12ADI2) can be generated on completion of double scan of group A, and the corresponding scan end interrupt request (S12GBADI/S12GCADI, S12GBADI1/S12GCADI1, or S12GBADI2/S12GCADI2) can be generated on completion of group B and group C scan. • A compare interrupt request (S12CMPAI, S12CMPAI1, S12CMPAI2, S12CMPBI, S12CMPBI1, or S12CMPBI2) can be generated upon a match with the comparison condition for the digital compare function. • The S12ADI/S12ADI1/S12ADI2, S12GBADI/S12GBADI1/S12GBADI2, and S12GCADI/S12GCADI1/S12GCADI2 interrupts can trigger the DMA controller (DMAC) and data transfer controller (DTC).
Event link	• The event signal is generated when all scans are finished. • The event signal is generated depending on conditions for comparison function window in single scan mode. • Able to start scanning by a trigger from the ELC
Low power consumption function	• Module stop state can be set.*3, *4

Note 1. The peripheral module clock PCLKB frequency is set according to the setting of the SCKCR.PCKB[3:0] bits and the A/D conversion clock ADCLK frequency is set according to the setting of the SCKCR.PCKD[3:0] bits.
Note 2. The number of extended bits during addition differs depending on the addition count.
2-bit extension: 1-time to 4-time conversion (add zero to three times)
4-bit extension: 16-time conversion (add 15 times)
Note 3. Refer to section 11, Low Power Consumption for details.
Note 4. Wait for 1 µs or longer to start A/D conversion after release from the module stop state.

Table 42.2 Functions of 12-Bit A/D Converter (1/3)

Item			Pin Name, Abbreviation		
			Unit 0 (S12AD)	Unit 1 (S12AD1)	Unit 2 (S12AD2)
Analog input channels			AN000 to AN006	AN100 to AN103	AN200 to AN211, AN216, AN217, internal reference voltage, temperature sensor output
Conditions for A/D conversion start	Software	Software trigger	Enabled		
	Asynchronous trigger	Trigger input pin	ADTRG0#	ADTRG1#	ADTRG2#
	Synchronous trigger	Compare match/input capture in MTU0.TGRA	TRGA0N		
		Compare match/input capture in MTU1.TGRA	TRGA1N		
		Compare match/input capture in MTU2.TGRA	TRGA2N		
		Compare match/input capture in MTU3.TGRA	TRGA3N		
		Compare match/input capture in MTU4.TGRA or underflow (trough) of MTU4.TCNT in complementary PWM mode	TRGA4N		
		Compare match/input capture in MTU6.TGRA	TRGA6N		
		Compare match/input capture in MTU7.TGRA or underflow (trough) of MTU7.TCNT in complementary PWM mode	TRGA7N		
		Compare match in MTU0.TGRE	TRG0N		
		Compare match between MTU4.TADCORA and MTU4.TCNT	TRG4AN		
		Compare match between MTU4.TADCORB and MTU4.TCNT	TRG4BN		
		Compare match between MTU4.TADCORA and MTU4.TCNT, or compare match between MTU4.TADCORB and MTU4.TCNT	TRG4AN or TRG4BN		
		Compare match between MTU4.TADCORA and MTU4.TCNT, and compare match between MTU4.TADCORB and MTU4.TCNT (when interrupt skipping function 2 is used)	TRG4ABN		
		Compare match between MTU7.TADCORA and MTU7.TCNT	TRG7AN		
		Compare match between MTU7.TADCORB and MTU7.TCNT	TRG7BN		
		Compare match between MTU7.TADCORA and MTU7.TCNT, or compare match between MTU7.TADCORB and MTU7.TCNT	TRG7AN or TRG7BN		
		Compare match between MTU7.TADCORA and MTU7.TCNT, and compare match between MTU7.TADCORB and MTU7.TCNT (when interrupt skipping function 2 is used)	TRG7ABN		
		Compare match/input capture in MTU9.TGRA	TRGA9N		
		Compare match in MTU9.TGRE	TRG9N		
		Compare match/input capture in MTU0.TGRA, or compare match in MTU0.TGRE	TRGA0N or TRG0N		
		Compare match/input capture in MTU9.TGRA, or compare match in MTU9.TGRE	TRGA9N or TRG9N		
		Compare match/input capture in MTU0.TGRA, or compare match/input capture in MTU9.TGRA	TRGA0N or TRGA9N		

Table 42.2 Functions of 12-Bit A/D Converter (2/3)

Item			Pin Name, Abbreviation		
			Unit 0 (S12AD)	Unit 1 (S12AD1)	Unit 2 (S12AD2)
Conditions for A/D conversion start	Synchronous trigger	Compare match in MTU0.TGRE, or compare match in MTU9.TGRE	TRG0N or TRG9N		
		Compare match/input capture in MTU9.TGRA, and compare match in MTU9.TGRE	TRG9AEN		
		Compare match/input capture in MTU0.TGRA, and compare match in MTU0.TGRE	TRG0AEN		
		Compare match/input capture in MTU0.TGRA, and compare match/input capture in MTU9.TGRA	TRGA09N		
		Compare match in MTU0.TGRE, and compare match in MTU9.TGRE	TRG09N		
		Compare match in GPTW0.GTADTRA	GTADTRA0N		
		Compare match in GPTW0.GTADTRB	GTADTRB0N		
		Compare match in GPTW1.GTADTRA	GTADTRA1N		
		Compare match in GPTW1.GTADTRB	GTADTRB1N		
		Compare match in GPTW2.GTADTRA	GTADTRA2N		
		Compare match in GPTW2.GTADTRB	GTADTRB2N		
		Compare match in GPTW3.GTADTRA	GTADTRA3N		
		Compare match in GPTW3.GTADTRB	GTADTRB3N		
		Compare match in GPTW0.GTADTRA, or compare match in GPTW0.GTADTRB	GTADTRA0N or GTADTRB0N		
		Compare match in GPTW1.GTADTRA, or compare match in GPTW1.GTADTRB	GTADTRA1N or GTADTRB1N		
		Compare match in GPTW2.GTADTRA, or compare match in GPTW2.GTADTRB	GTADTRA2N or GTADTRB2N		
		Compare match in GPTW3.GTADTRA, or compare match in GPTW3.GTADTRB	GTADTRA3N or GTADTRB3N		
		Compare match in GPTW4.GTADTRA	GTADTRA4N		
		Compare match in GPTW4.GTADTRB	GTADTRB4N		
		Compare match in GPTW5.GTADTRA	GTADTRA5N		
		Compare match in GPTW5.GTADTRB	GTADTRB5N		
		Compare match in GPTW6.GTADTRA	GTADTRA6N		
		Compare match in GPTW6.GTADTRB	GTADTRB6N		
		Compare match in GPTW7.GTADTRA	GTADTRA7N		
		Compare match in GPTW7.GTADTRB	GTADTRB7N		
		Compare match in GPTW4.GTADTRA, or compare match in GPTW4.GTADTRB	GTADTRA4N or GTADTRB4N		
		Compare match in GPTW5.GTADTRA, or compare match in GPTW5.GTADTRB	GTADTRA5N or GTADTRB5N		
		Compare match in GPTW6.GTADTRA, or compare match in GPTW6.GTADTRB	GTADTRA6N or GTADTRB6N		
		Compare match in GPTW7.GTADTRA, or compare match in GPTW7.GTADTRB	GTADTRA7N or GTADTRB7N		

Table 42.2 Functions of 12-Bit A/D Converter (3/3)

Item			Pin Name, Abbreviation		
			Unit 0 (S12AD)	Unit 1 (S12AD1)	Unit 2 (S12AD2)
Conditions for A/D conversion start	Synchronous trigger	Compare match between TMR0.TCORA and TMR0.TCNT	TMTRG0AN_0		
		Compare match between TMR2.TCORA and TMR2.TCNT	TMTRG0AN_1		
		Compare match between TMR4.TCORA and TMR4.TCNT	TMTRG0AN_2		
		Compare match between TMR6.TCORA and TMR6.TCNT	TMTRG0AN_3		
		ELC trigger	ELCTRG00N	ELCTRG10N	ELCTRG20N
			ELCTRG01N	ELCTRG11N	ELCTRG21N
			ELCTRG00N or ELCTRG01N	ELCTRG10N or ELCTRG11N	ELCTRG20N or ELCTRG21N
Channel-dedicated sample-and-hold function		Target channels	AN000 to AN002	AN100 to AN102	—
Interrupts			S12ADI, S12GBADI, S12GCADI, S12CMPAI, S12CMPBI interrupt	S12ADI1, S12GBADI1, S12GCADI1, S12CMPAI1, S12CMPBI1 interrupt	S12ADI2, S12GBADI2, S12GCADI2, S12CMPAI2, S12CMPBI2 interrupt
Setting of module stop function*1, *2			MSTPCRA.MSTPA17 bit	MSTPCRA.MSTPA16 bit	MSTPCRA.MSTPA23 bit

Note: When setting an A/D conversion start trigger to ADTRG0#, ADTRG1#, or ADTRG2#, set the pin mode control bit in the port mode register for the corresponding pin to 1 (peripheral functions), and set the pin function select bit in the pin function control register to ADTRG0#, ADTRG1#, or ADTRG2#. Refer to section 20, I/O Ports and section 21, Multi-Function Pin Controller (MPC) for details.

Note 1. Refer to section 11, Low Power Consumption for details.

Note 2. Wait for 1 µs or longer to start A/D conversion after release from the module stop state.

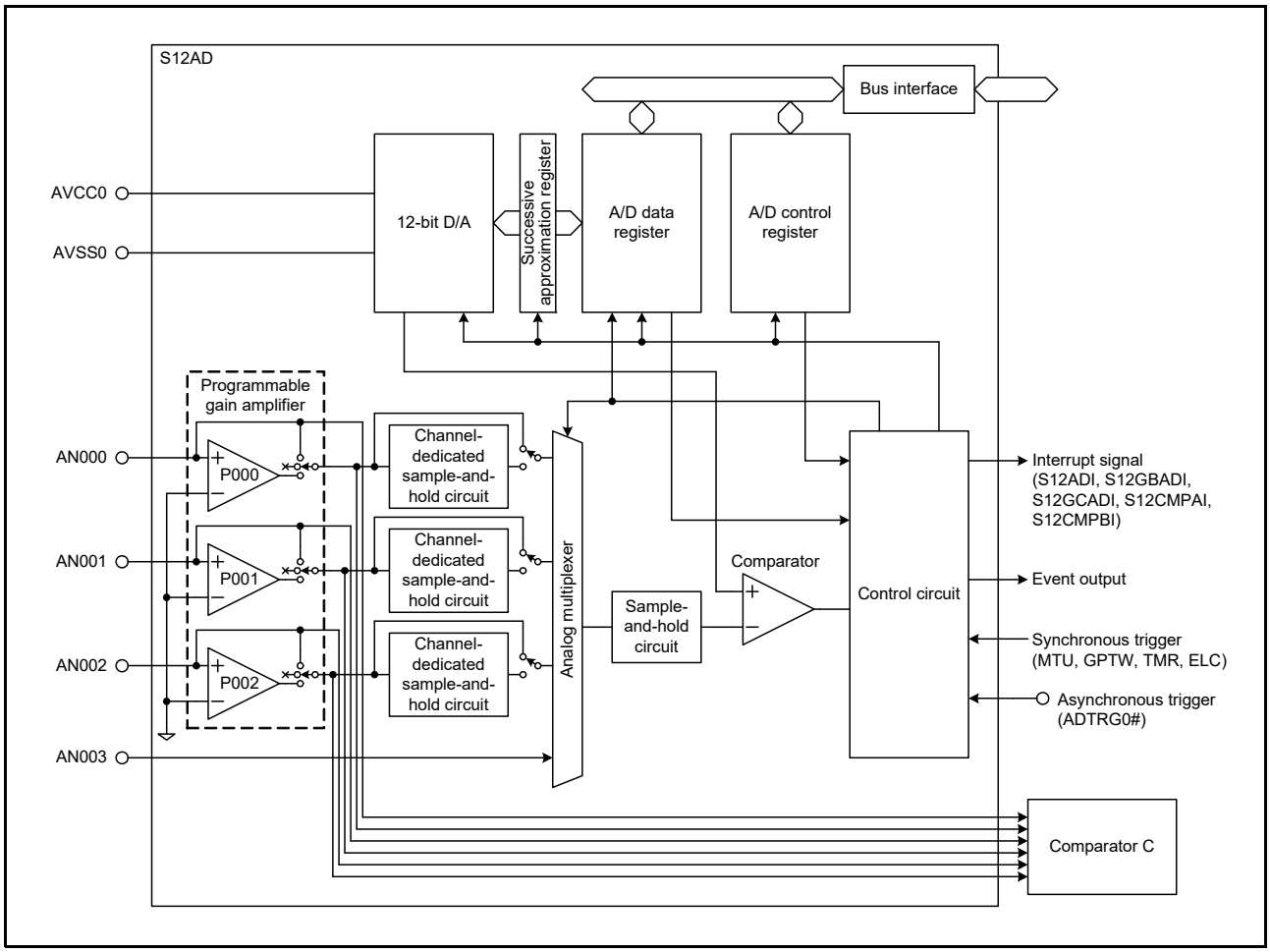


Figure 42.1 Block Diagram of 12-Bit A/D Converter (Products with 64 Kbytes of RAM, Unit 0)

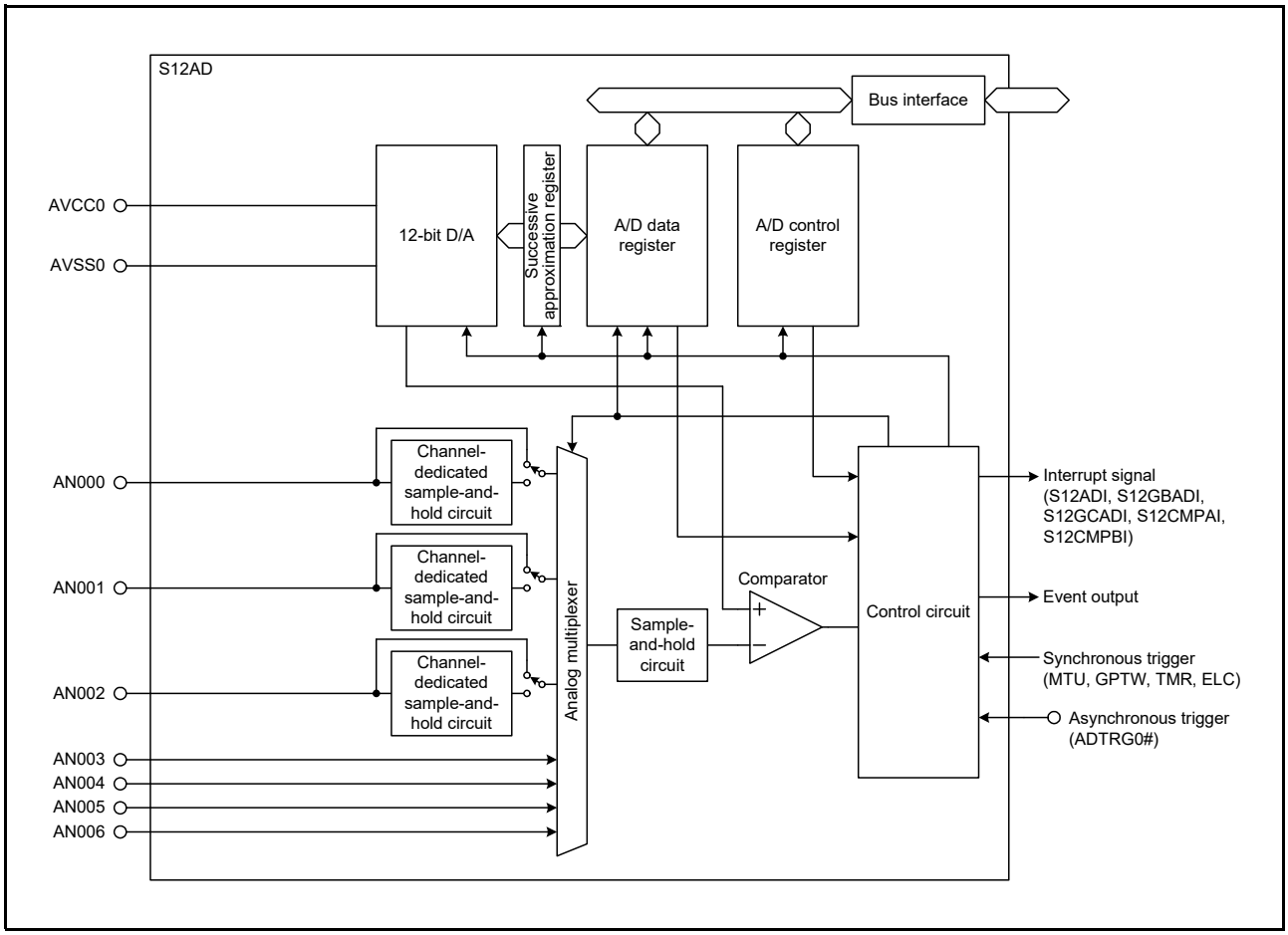


Figure 42.2 Block Diagram of 12-Bit A/D Converter (Products with 48 Kbytes of RAM, Unit 0)

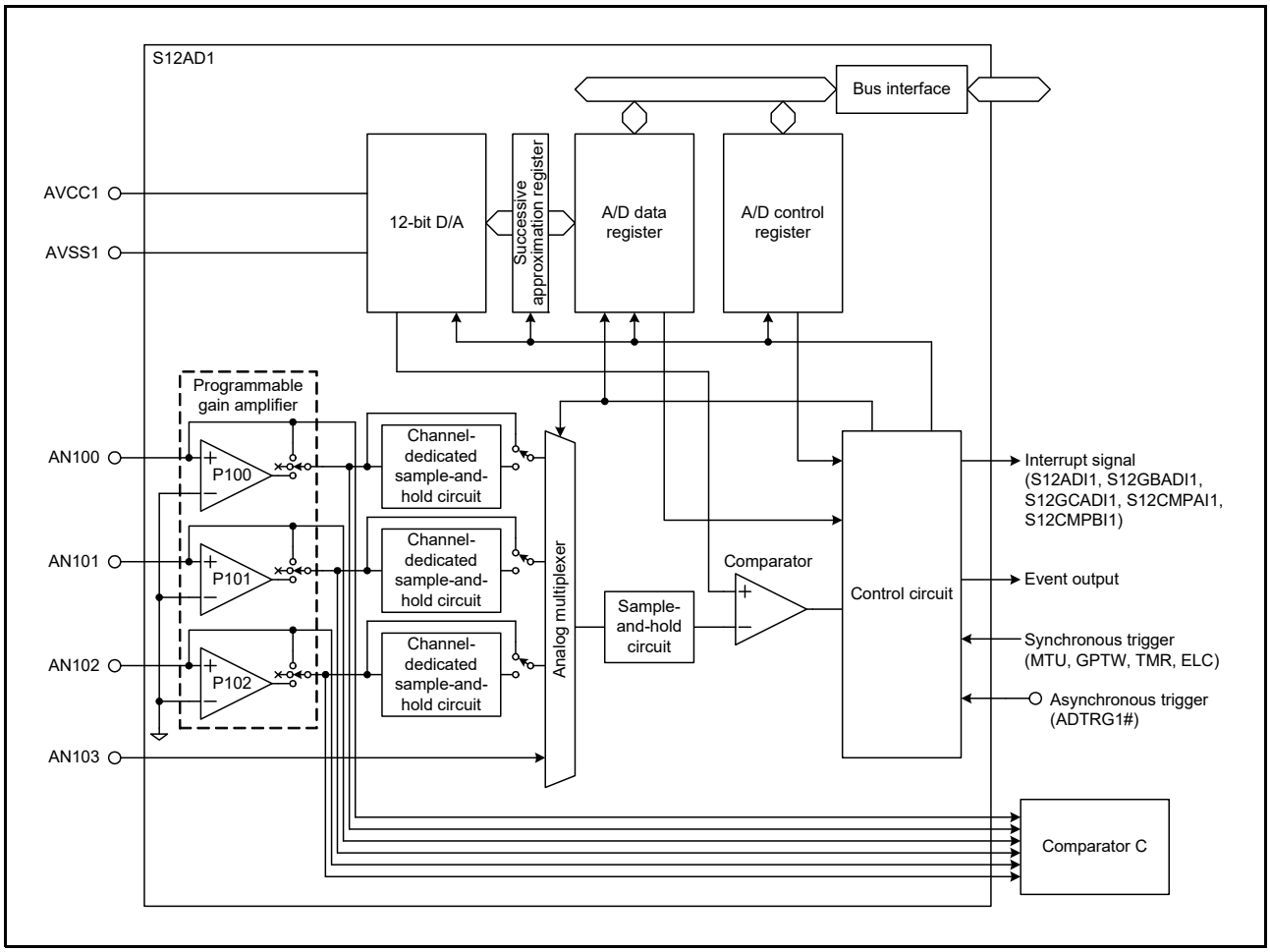


Figure 42.3 Block Diagram of 12-Bit A/D Converter (Products with 64 Kbytes of RAM, Unit 1)

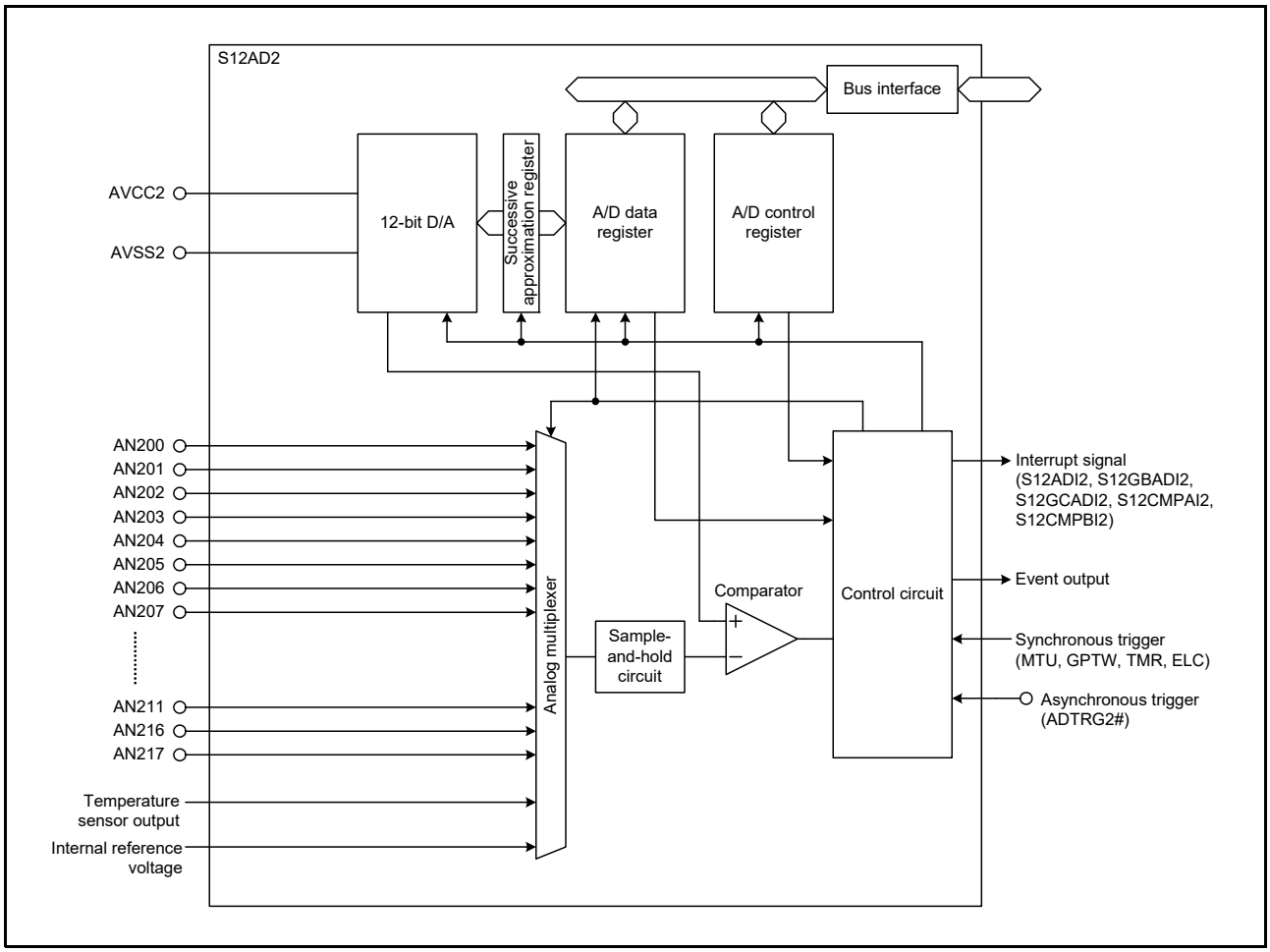


Figure 42.4 Block Diagram of 12-Bit A/D Converter (Unit 2)

Table 42.3 lists the input/output pins of the 12-bit A/D converter.

There are three 12-bit A/D converter units, unit 0 (S12AD), unit 1 (S12AD1), and unit 2 (S12AD2), with operation independent of each other. The input channels of S12AD, S12AD1, and S12AD2 can be divided into three groups for operation.

Table 42.3 Input/Output Pins of 12-Bit A/D Converter

Unit	Pin Name	I/O	Function
Unit 0 (S12AD)	AVCC0	—	Analog power supply pin
	AVSS0	—	Analog ground pin
	AN000 to AN006	Input	Analog input pins
	ADTRG0#	Input	External trigger input pin for starting A/D conversion
	ADST0	Output	Output pin for state of the ADST bit
Unit 1 (S12AD1)	AVCC1	—	Analog power supply pin
	AVSS1	—	Analog ground pin
	AN100 to AN103	Input	Analog input pins
	ADTRG1#	Input	External trigger input pin for starting A/D conversion
	ADST1	Output	Output pin for state of the ADST bit
Unit 2 (S12AD2)	AVCC2	—	Analog power supply pin
	AVSS2	—	Analog ground pin
	AN200 to AN211, AN216, AN217	Input	Analog input pins
	ADTRG2#	Input	External trigger input pin for starting A/D conversion
	ADST2	Output	Output pin for state of the ADST bit

42.2 Register Descriptions

Products with 48 Kbytes of RAM do not have unit 1 (S12AD1). Do not access the registers in unit 1.

42.2.1 A/D Data Registers y (ADDRy) (y = 0 to 11, 16, 17), A/D Data Duplication Register (ADDBLDR), A/D Data Duplication Register A (ADDBLDRA), A/D Data Duplication Register B (ADDBLDRB), A/D Temperature Sensor Data Register (ADTSDR), A/D Internal Reference Voltage Data Register (ADOCDR)

Address(es): S12AD.ADDR0 0008 9020h, S12AD.ADDR1 0008 9022h, S12AD.ADDR2 0008 9024h, S12AD.ADDR3 0008 9026h, S12AD.ADDR4 0008 9028h, S12AD.ADDR5 0008 902Ah, S12AD.ADDR6 0008 902Ch, S12AD.ADDBLDR 0008 9018h, S12AD.ADDBLDRA 0008 9084h, S12AD.ADDBLDRB 0008 9086h, S12AD1.ADDR0 0008 9220h, S12AD1.ADDR1 0008 9222h, S12AD1.ADDR2 0008 9224h, S12AD1.ADDR3 0008 9226h, S12AD1.ADDBLDR 0008 9218h, S12AD1.ADDBLDRA 0008 9284h, S12AD1.ADDBLDRB 0008 9286h, S12AD2.ADDR0 0008 9420h, S12AD2.ADDR1 0008 9422h, S12AD2.ADDR2 0008 9424h, S12AD2.ADDR3 0008 9426h, S12AD2.ADDR4 0008 9428h, S12AD2.ADDR5 0008 942Ah, S12AD2.ADDR6 0008 942Ch, S12AD2.ADDR7 0008 942Eh, S12AD2.ADDR8 0008 9430h, S12AD2.ADDR9 0008 9432h, S12AD2.ADDR10 0008 9434h, S12AD2.ADDR11 0008 9436h, S12AD2.ADDR16 0008 9440h, S12AD2.ADDR17 0008 9442h, S12AD2.ADDBLDR 0008 9418h, S12AD2.ADDBLDRA 0008 9484h, S12AD2.ADDBLDRB 0008 9486h, S12AD2.ADTSDR 0008 941Ah, S12AD2.ADOCDR 0008 941Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The ADDRy registers (y = 0 to 6: S12AD, y = 0 to 3: S12AD1, y = 0 to 11, 16, 17: S12AD2) are 16-bit read-only registers which store the A/D conversion results.
The ADDBLDR register is a 16-bit read-only register used in double trigger mode. The ADDBLDR register stores the results of A/D conversion when the conversion is started by the second trigger.
The ADDBLDRA and ADDBLDRB registers are 16-bit read-only registers that store the A/D conversion results in response to the respective triggers during extended operation in double trigger mode.
The ADTSDR register is a 16-bit read-only register that stores the A/D-converted value of the temperature sensor output.
The ADOCDR register is a 16-bit read-only register that stores the A/D conversion results of the internal reference voltage.
The format of each register differs depending on the conditions below.

- Settings of the A/D data register format select bit (ADCER.ADRFMT) (right-justified or left-justified)
- Settings of the addition count select bits (ADADC.ADC[2:0]) (2-, 3-, 4-, or 16-time conversion)
- Settings of the average mode enable bit (ADADC.AVEE) (add or average)

The data formats for each given condition are shown below.

(1) When A/D-Converted Value Addition/Average Mode is Disabled

- Right-justified format
  The A/D-converted value is stored in bits 11 to 0. Bits 15 to 12 are read as 0.
- Left-justified format
  The A/D-converted value is stored in bits 15 to 4. Bits 3 to 0 are read as 0.

(2) When A/D-Converted Average Mode is Selected

A/D-converted value average mode can be set only when 2- or 4-time conversion is selected in A/D-converted value addition mode.

- Right-justified format
  The mean value of the A/D-converted results of the same channel is stored in bits 11 to 0.
  Bits 15 to 12 are read as 0.

- Left-justified format
  The mean value of the A/D-converted results of the same channel is stored in bits 15 to 4.
  Bits 3 to 0 are read as 0.

(3) When A/D-Converted Value Addition Mode is Selected

When A/D-converted addition mode is selected, the value added by the A/D-converted value of the same channel is indicated. The number of A/D conversions can be set to 1, 2, 3, 4, or 16 times. If A/D-converted addition mode is selected, when the conversion count is set to 1 to 4 times, the value added by the A/D conversion result is retained in the A/D data register as 2-bit extended data of the conversion accuracy bits; when the conversion count is set to 16 times, the value added by the A/D conversion result is retained in the A/D data register as 4-bit extended data of the conversion accuracy bits. Even if A/D-converted value addition mode is selected, the value is stored in the A/D data register according to the settings of the A/D data register format select bits.

- Right-justified format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times)
  The value added by the A/D-converted value of the same channel is stored in bits 13 to 0.
  Bits 15 and 14 are read as 0.
- Right-justified format (in A/D-converted value addition mode and when number of conversions is selected for 16 times)
  The value added by the A/D-converted value of the same channel is stored in bits 15 to 0.
- Left-justified format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times)
  The value added by the A/D-converted value of the same channel is stored in bits 15 to 2.
  Bits 1 and 0 are read as 0.
- Left-justified format (in A/D-converted value addition mode and when number of conversions is selected for 16 times)
  The value added by the A/D-converted value of the same channel is stored in bits 15 to 0.

42.2.2 A/D Self-Diagnosis Data Register (ADRD)

Address(es): S12AD.ADRD 0008 901Eh, S12AD1.ADRD 0008 921Eh, S12AD2.ADRD 0008 941Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The ADRD register is a 16-bit read-only register that stores the A/D conversion results based on the 12-bit A/D converter's self-diagnosis. In addition to the A/D-converted value, the self-diagnosis status is included in. In the ADRD register, the different formats are used depending on the conditions below.

- Settings of the A/D data register format select bit (ADCER.ADRFMT) (right-justified or left-justified)

The A/D-converted value addition mode and A/D-converted value average mode cannot be applied to the A/D self-diagnosis function. If temperature sensor output or internal reference voltage is selected, self-diagnosis function cannot be selected. For details of self-diagnosis, refer to section 42.2.14, A/D Control Extended Register (ADCER).

The data formats for each given condition are shown below.

- Right-justified format
  The A/D-converted value is stored in bits 11 to 0. The self-diagnosis status is stored in bits 15 and 14.
  Bits 13 and 12 are read as 0.
- Left-justified format
  The A/D-converted value is stored in bits 15 to 4. The self-diagnosis status is stored in bits 1 and 0.
  Bits 3 and 2 are read as 0.

Table 42.4 Self-Diagnosis Status Description

Bits 15 and 14 for Right-Justified Format Setting Bits 1 and 0 for Left-Justified Format Setting	Self-Diagnosis Status
00b	Self-diagnosis has never been executed since power-on.
01b	Self-diagnosis using the voltage of 0 V has been executed.
10b	Self-diagnosis using the voltage of 1/2 × AVCC has been executed.
11b	Self-diagnosis using the voltage of AVCC has been executed.

Note: For details of self-diagnosis, refer to section 42.2.14, A/D Control Extended Register (ADCER).

42.2.3 A/D Control Register (ADCSR)

Address(es): S12AD.ADCSR 0008 9000h, S12AD1.ADCSR 0008 9200h, S12AD2.ADCSR 0008 9400h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ADST	ADCS[1:0]		ADIE	—	—	TRGE	EXTRG	DBLE	GBADIE	—	DBLANS[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	DBLANS[4:0]	Double Trigger Channel Select	These bits select one analog input channel for double triggered operation. The setting is only effective while double trigger mode is selected.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	GBADIE	Group B Scan End Interrupt Enable	0: Group B scan end interrupt is disabled. 1: Group B scan end interrupt is enabled.	R/W
b7	DBLE	Double Trigger Mode Select	0: Deselects double trigger mode. 1: Selects double trigger mode.	R/W
b8	EXTRG	Trigger Select	0: A/D conversion is started by synchronous trigger. 1: A/D conversion is started by asynchronous trigger.	R/W
b9	TRGE	Trigger Start Enable	0: Disables A/D conversion to be started by synchronous or asynchronous trigger. 1: Enables A/D conversion to be started by synchronous or asynchronous trigger.	R/W
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	ADIE	Scan End Interrupt Enable	0: Scan end interrupt is disabled. 1: Scan end interrupt is enabled.	R/W
b14, b13	ADCS[1:0]	Scan Mode Select	b14 b13 0 0: Single scan mode 0 1: Group scan mode 1 0: Continuous scan mode 1 1: Setting prohibited	R/W
b15	ADST	A/D Conversion Start	0: Stops A/D conversion process. 1: Starts A/D conversion process.	R/W

The ADCSR register sets double trigger mode, A/D conversion start trigger; enables/disables scan end interrupt; selects the scan mode; and starts or stops A/D conversion.

DBLANS[4:0] Bits (Double Trigger Channel Select)

The DBLANS[4:0] bits select one of the channels for A/D conversion data duplication in double trigger mode. The A/D conversion results of the analog input of the channel selected by the DBLANS[4:0] bits are stored into the A/D data register y when conversion is started by the first trigger, and into the A/D data duplication register when started by the second trigger. Table 42.5 shows selection of the channel for double triggered operation.

When double trigger mode is selected, the channels selected by the ADANSA0 and ADANSA1 registers are invalid, and the channel selected by the DBLANS[4:0] bits is subjected to A/D conversion instead.

When double trigger mode is selected in group scan mode, double trigger mode operation is performed for group A only and not performed for group B or C. Also, in double trigger mode, the analog inputs of multiple channels, temperature sensor outputs, or internal reference voltage cannot be selected for group A, but can be selected for groups B and C.

The DBLANS[4:0] bits should be set while the ADST bit is 0. They should not be set simultaneously when 1 is written to the ADST bit.

Table 42.5 Relationship Between DBLANS[4:0] Bits Settings and Double Trigger Enabled Channels

	Duplication Channel		
DBLANS[4:0]	S12AD (Unit 0)	S12AD1 (Unit 1)	S12AD2 (Unit 2)
00000b	AN000	AN100	AN200
00001b	AN001	AN101	AN201
00010b	AN002	AN102	AN202
00011b	AN003	AN103	AN203
00100b	AN004	—	AN204
00101b	AN005	—	AN205
00110b	AN006	—	AN206
00111b	—	—	AN207
01000b	—	—	AN208
01001b	—	—	AN209
01010b	—	—	AN210
01011b	—	—	AN211
10000b	—	—	AN216
10001b	—	—	AN217

—: Do not specify this value.

Note: Duplication cannot be selected for the A/D conversion data of self-diagnosis, temperature sensor output, and internal reference voltage.

GBADIE Bit (Group B Scan End Interrupt Enable)

The GBADIE bit enables or disables group B scan end interrupt in group scan mode.

A group B scan end interrupt is provided individually for each unit. Table 42.6 shows the relationship between each unit and the scan end interrupt for group B.

Table 42.6 Relationship Between Each Unit and Group B Scan End Interrupt

Unit	Group B Scan End Interrupt
S12AD	S12GBADI
S12AD1	S12GBADI1
S12AD2	S12GBADI2

DBLE Bit (Double Trigger Mode Select)

Double trigger mode has a function to store the resulting data of A/D conversion started by the first and second synchronous triggers into separate registers.

When double trigger mode is selected, the channels specified in the ADANSA0 and ADANSA1 registers are invalid and the channel selected by the DBLANS[4:0] bits is effective instead. Double trigger mode can be only operated by the synchronous trigger selected by the ADSTRGR.TRSA[6:0] bits. Do not generate an asynchronous or software trigger.

The A/D conversion results started by the first trigger are stored into the A/D data register y and those started by the second trigger are stored into the A/D data duplication register. In this case, when the ADIE bit is set to 1, the interrupt request is generated upon completion of the second conversion instead of upon completion of the first conversion.

In continuous scan mode, double trigger mode should not be selected. In addition, double trigger mode should not be used for self-diagnosis, or conversion of the temperature sensor output and internal reference voltage. When using double trigger mode in group scan mode, A/D conversion of the temperature sensor output or internal reference voltage should not be selected for group A.

The DBLE bit should be set after the ADST bit has been set to 0.

EXTRG Bit (Trigger Select)

The EXTRG bit selects the synchronous trigger or the asynchronous trigger as the trigger for starting A/D conversion. To use an external pin (asynchronous trigger) to start A/D conversion, set the ADCSR.TRGE and ADCSR.EXTRG bits to 1 while the external trigger pin (ADTRG0#, ADTRG1#, or ADTRG2#) is being driven high. Then, if the ADTRG0#, ADTRG1#, or ADTRG2# signal is changed to low, the falling edge is detected and the scan process is started. In this case, the pulse width of the low-level input must be at least 1.5 clock cycles of PCLKB.
Table 42.7 lists the units and the corresponding external pins (asynchronous triggers).

Table 42.7 Relationship Between Each Unit and External Pin (Asynchronous Trigger)

Unit	External Pin (Asynchronous Trigger)
S12AD	ADTRG0#
S12AD1	ADTRG1#
S12AD2	ADTRG2#

TRGE Bit (Trigger Start Enable)

The TRGE bit enables or disables A/D conversion by the synchronous trigger and the asynchronous trigger.
This bit should be set to 1 in group scan mode.

ADIE Bit (Scan End Interrupt Enable)

The ADIE bit enables or disables the A/D scan end interrupt in scans except for groups B and C scan in group scan mode.
With double trigger mode deselected, the A/D scan conversion end interrupt is generated after the first scan is completed if the ADIE bit is set to 1.
With double trigger mode selected, the A/D scan conversion end interrupt is generated after the second scan is completed if the ADIE bit is set to 1 as long as the scan is started by the synchronous trigger selected by the ADSTRGR.TRSA[6:0] bits.
When scan is started by a software trigger, even with double trigger mode selected, the A/D scan conversion end interrupt is generated if the ADIE bit is set to 1 when the scan is completed. The A/D scan conversion end interrupt is provided individually for each unit. Table 42.8 shows the relationship between each unit and the A/D scan conversion end interrupt.

Table 42.8 Relationship Between Each Unit and A/D Scan Conversion End Interrupt

Unit	A/D Scan Conversion End Interrupt
S12AD	S12ADI
S12AD1	S12ADI1
S12AD2	S12ADI2

ADCS[1:0] Bits (Scan Mode Select)

The ADCS[1:0] bits selects the scan mode.
In single scan mode, A/D conversion is performed for the analog inputs for up to seven channels for A/D converter unit 0, four channels for unit 1, and 14 channels for unit 2 selected with the ADANSA0 and ADANSA1 registers in the ascending order of arbitrary channel number, and the A/D conversion is stopped when one cycle of the A/D scan conversion is completed for all the selected channels.
The temperature sensor output or internal reference voltage can be selected; however, it cannot be selected with a channel input. Conversion of the temperature sensor output takes priority over that of internal reference voltage when both the temperature sensor output and internal reference voltage are selected.
In continuous scan mode, A/D conversion is performed for the analog inputs for up to seven channels for A/D converter unit 0, four channels for unit 1, and 14 channels for unit 2 selected with the ADANSA0 and ADANSA1 registers in the ascending order of arbitrary channel number, and the A/D conversion is repeated from the first channel when one cycle

of the A/D scan conversion is completed for all the selected channels.
In continuous scan mode, selecting the temperature sensor output and internal reference voltage is prohibited.
In group scan mode, A/D conversion is performed for the analog inputs (group A) for up to seven channels for A/D converter unit 0, four channels for unit 1, and 14 channels for unit 2 selected with the ADANSA0 and ADANSA1 registers in the ascending order of arbitrary channel number after scanning is started by the synchronous trigger selected by the ADSTRGR.TRSA[6:0] bits, and the A/D conversion is stopped when one cycle of the A/D scan conversion is completed for all the selected channels.
A/D conversion is also performed for the analog inputs (group B or C) for up to seven channels for A/D converter unit 0, four channels for unit 1, and 14 channels for unit 2 selected with the ADANSB0 and ADANSB1 registers and the ADANSC0 and ADANSC1 registers in the ascending order of arbitrary channel number after A/D conversion is started by the synchronous trigger selected by the ADSTRGR.TRSB[6:0] bits and the ADGCTRGR2.TRSC6 and ADGCTRGR.TRSC[5:0] bits, and when A/D conversion is completed for all the selected channels, A/D conversion is stopped.
The temperature sensor output or internal reference voltage can be selected; however, it cannot be selected with a channel input in the same group. Conversion of the temperature sensor output takes priority over that of internal reference voltage when both the temperature sensor output and internal reference voltage are selected.
When selecting group scan mode, different channels and triggers should be selected for groups A, B, and C.
When using two groups while group scan mode is set, use groups A and B (ADGCTRGR.GRCE bit = 0). When using three groups, use groups A, B, and C (ADGCTRGR.GRCE bit = 1).
The ADCS[1:0] bits should be set while the ADST bit is 0. They should not be set simultaneously when 1 is written to the ADST bit.

Table 42.9 Selection of Scan Mode, Double Trigger Mode, and Targets for A/D Conversion

Scan Mode Setup	Double Trigger Mode Setting	Targets for A/D Conversion				
		Self-Diagnosis	Analog Input (Including Group A)	Analog Input (Group B and Group C)	Temperature Sensor Output	Internal Reference Voltage
Single scan	DBLE = 0	✓	✓	×	✓	✓
	DBLE = 1	×	✓ (1 channel only)	×	×	×
Continuous scan	DBLE = 0	✓	✓	×	×	×
	DBLE = 1	×	×	×	×	×
Group scan	DBLE = 0	✓	✓	✓	✓	✓
	DBLE = 1	×	✓ (1 channel only)	✓	✓ (Group B and Group C)	✓ (Group B and Group C)

✓: Selectable, ×: Not selectable

ADST Bit (A/D Conversion Start)

The ADST bit starts or stops A/D conversion process.
Before the ADST bit is set to 1, set the A/D conversion clock, the conversion mode, and conversion target analog input.
[Setting conditions]

- 1 is written by software.
- The synchronous trigger selected by the ADSTRGR.TRSA[6:0] bits is detected with the ADCSR.EXTRG and ADCSR.TRGE bits being set to 0 and 1, respectively.
- The synchronous trigger selected by the ADSTRGR.TRSB[6:0] bits is detected with the ADCSR.TRGE bit being set to 1 in group scan mode.
- The asynchronous trigger is detected with the ADCSR.TRGE and ADCSR.EXTRG bits being set to 1 and the ADSTRGR.TRSA[6:0] bits being set to 0000000b.
- With group priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1),

a group B or C trigger is detected and A/D conversion of group B or C is started.

- With group priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), the ADGSPCR.GBRP bit is set to 1 and A/D conversion of the lowest-priority group is started.

[Clearing conditions]

- 0 is written by software.
- The A/D conversion of all the selected channels, temperature sensor output, or the internal reference voltage (for unit 2 only) is completed in single scan mode.
- Group A scan is completed in group scan mode.
- Group B scan is completed in group scan mode.
- Group C scan is completed in group scan mode.
- With group priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), the ADGSPCR.GBRSCN bit is set to 1 and the scanning of the low-priority group started by a trigger is stopped.

Note: When group priority control operation mode has been enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), do not set the ADST bit to 1.

Note: When group priority control operation mode has been enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1) and the ADGSPCR.GBRP bit = 1, do not set the ADST bit to 0. When forcibly terminating A/D conversion, follow the procedure for clearing the ADST bit.

Note: When the single-scan continuous function is used (ADGSPCR.GBRP bit = 1) with group priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), the ADST bit remains as 1.

42.2.4 A/D Channel Select Register A0 (ADANSA0)

(1) S12AD.ADANSA0

Address(es): S12AD.ADANSA0 0008 9004h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	ANSA006	ANSA005	ANSA004	ANSA003	ANSA002	ANSA001	ANSA000
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ANSA000	A/D Conversion Channel Select	Select whether the voltages on AN000 to AN006 are to be converted or not. 0: Not selected for conversion 1: Selected for conversion	R/W
b1	ANSA001			R/W
b2	ANSA002			R/W
b3	ANSA003			R/W
b4	ANSA004			R/W
b5	ANSA005			R/W
b6	ANSA006			R/W
b15 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The S12AD.ADANSA0 register selects analog input channels for A/D conversion from among AN000 to AN006. In group scan mode, this register selects group A channels.

ANSA0n Bit (A/D Conversion Channel Select) (n = 00 to 06)

The ANSA0n bit selects analog input channels for A/D conversion from among AN000 to AN006. The channels to be selected and the number of channels can be arbitrarily set. The ANSA000 bit corresponds to AN000 and the ANSA006 bit corresponds to AN006.

When double trigger mode is selected, the channel selected by the S12AD.ADCSR.DBLANS[4:0] bits is selected in group A, and the ANSA0n bit setting is invalid.

The ANSA0n bit should be set while the S12AD.ADCSR.ADST bit is 0.

(2) S12AD1.ADANSA0

Address(es): S12AD1.ADANSA0 0008 9204h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	ANSA003	ANSA002	ANSA001	ANSA000
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ANSA000	A/D Conversion Channel Select	Select whether the voltages on AN100 to AN103 are to be converted or not. 0: Not selected for conversion 1: Selected for conversion	R/W
b1	ANSA001			R/W
b2	ANSA002			R/W
b3	ANSA003			R/W
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The S12AD1.ADANSA0 register selects analog input channels for A/D conversion from among AN100 to AN103. In group scan mode, this register selects group A channels.

ANSA0n Bit (A/D Conversion Channel Select) (n = 00 to 03)

The ANSA0n bit selects analog input channels for A/D conversion from among AN100 to AN103. The channels to be selected and the number of channels can be arbitrarily set. The ANSA000 bit corresponds to AN100 and the ANSA003 bit corresponds to AN103.

When double trigger mode is selected, the channel selected by the S12AD1.ADCSR.DBLANS[4:0] bits is selected in group A, and the ANSA0n bit setting is invalid.

The ANSA0n bit should be set while the S12AD1.ADST bit is 0.

(3) S12AD2.ADANSA0

Address(es): S12AD2.ADANSA0 0008 9404h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	ANSA011	ANSA010	ANSA009	ANSA008	ANSA007	ANSA006	ANSA005	ANSA004	ANSA003	ANSA002	ANSA001	ANSA000
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ANSA000	A/D Conversion Channel Select	Select whether the voltages on AN200 to AN211 are to be converted or not. 0: Not selected for conversion 1: Selected for conversion	R/W
b1	ANSA001			R/W
b2	ANSA002			R/W
b3	ANSA003			R/W
b4	ANSA004			R/W
b5	ANSA005			R/W
b6	ANSA006			R/W
b7	ANSA007			R/W
b8	ANSA008			R/W
b9	ANSA009			R/W
b10	ANSA010			R/W
b11	ANSA011			R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The S12AD2.ADANSA0 register selects analog input channels for A/D conversion from among AN200 to AN211. In group scan mode, this register selects group A channels.

ANSA0n Bit (A/D Conversion Channel Select) (n = 00 to 11)

The ANSA0n bit selects analog input channels for A/D conversion from among AN200 to AN211. The channels to be selected and the number of channels can be arbitrarily set. The ANSA000 bit corresponds to AN200 and the ANSA011 bit corresponds to AN211.

When performing A/D conversion of the temperature sensor output or internal reference voltage in single scan mode or for group A in group scan mode, do not select analog input channels (set this register to 0000h).

When double trigger mode is selected, the channel selected by the S12AD2.ADCSR.DBLANS[4:0] bits is selected in group A, and the ANSA0n bit setting is invalid.

The ANSA0n bit should be set while the S12AD2.ADCSR.ADST bit is 0.

42.2.5 A/D Channel Select Register A1 (ADANSA1)

Address(es): S12AD2.ADANSA1 0008 9406h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ANSA101	ANSA100
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ANSA100	A/D Conversion Channel Select	Select whether the voltages on AN216 to AN217 are to be converted or not. 0: Not selected for conversion 1: Selected for conversion	R/W
b1	ANSA101			R/W
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The S12AD2.ADANSA1 register selects analog input channels for A/D conversion from among AN216 to AN217. In group scan mode, this register selects group A channels.

ANSA1n Bit (A/D Conversion Channel Select) (n = 00, 01)

The ANSA1n bit selects analog input channels for A/D conversion from among AN216 to AN217. The channels to be selected and the number of channels can be arbitrarily set. The ANSA100 bit corresponds to AN216 and the ANSA101 bit corresponds to AN217.

When performing A/D conversion of the temperature sensor output or internal reference voltage in single scan mode or for group A in group scan mode, do not select analog input channels (set this register to 0000h).

When double trigger mode is selected, the channel selected by the S12AD2.ADCSR.DBLANS[4:0] bits is selected in group A, and the ANSA1n bit setting is invalid.

The ANSA1n bit should be set while the S12AD2.ADCSR.ADST bit is 0.

42.2.6 A/D Channel Select Register B0 (ADANSB0)

(1) S12AD.ADANSB0

Address(es): S12AD.ADANSB0 0008 9014h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	ANSB006	ANSB005	ANSB004	ANSB003	ANSB002	ANSB001	ANSB000
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ANSB000	A/D Conversion Channel Select	Select whether the voltages on AN000 to AN006 are to be converted or not. 0: Not selected for conversion 1: Selected for conversion	R/W
b1	ANSB001			R/W
b2	ANSB002			R/W
b3	ANSB003			R/W
b4	ANSB004			R/W
b5	ANSB005			R/W
b6	ANSB006			R/W
b15 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The S12AD.ADANSB0 register selects analog input channels for A/D conversion from among AN000 to AN006 in group B when group scan mode is selected. The S12AD.ADANSB0 register is not used in any scan mode other than group scan mode.

ANSB0n Bit (A/D Conversion Channel Select) (n = 00 to 06)

The ANSB0n bit selects analog input channels for A/D conversion from among AN000 to AN006 in group B when group scan mode is selected. The S12AD.ADANSB0 register is used for group scan mode only; not used for any other modes. The channels specified in group A (the S12AD.ADANSA0 register), the channels specified in group C (the S12AD.ADANSC0 register), and the channels corresponding to group A, selected with the S12AD.ADCSR.DBLANS[4:0] bits in double trigger mode should be excluded as the channels to be selected and the number of channels to be set.

The ANSB000 bit corresponds to AN000 and the ANSB006 bit corresponds to AN006.

The ANSB0n bit should be set while the S12AD.ADCSR.ADST bit is 0.

(2) S12AD1.ADANSB0

Address(es): S12AD1.ADANSB0 0008 9214h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	ANSB003	ANSB002	ANSB001	ANSB000
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ANSB000	A/D Conversion Channel Select	Select whether the voltages on AN100 to AN103 are to be converted or not. 0: Not selected for conversion 1: Selected for conversion	R/W
b1	ANSB001			R/W
b2	ANSB002			R/W
b3	ANSB003			R/W
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The S12AD1.ADANSB0 register selects analog input channels for A/D conversion from among AN100 to AN103 in group B when group scan mode is selected. The S12AD1.ADANSB0 register is not used in any scan mode other than group scan mode.

ANSB0n Bit (A/D Conversion Channel Select) (n = 00 to 03)

The ANSB0n bit selects analog input channels for A/D conversion from among AN100 to AN103 in group B when group scan mode is selected. The S12AD1.ADANSB0 register is used for group scan mode only; not used for any other modes. The channels specified in group A (the S12AD1.ADANSA0 register), the channels specified in group C (the S12AD1.ADANSC0 register), and the channels corresponding to group A, selected with the S12AD1.ADCSR.DBLANS[4:0] bits in double trigger mode should be excluded as the channels to be selected and the number of channels to be set.

The ANSB000 bit corresponds to AN100 and the ANSB003 bit corresponds to AN103.

The ANSB0n bit should be set while the S12AD1.ADCSR.ADST bit is 0.

(3) S12AD2.ADANSB0

Address(es): S12AD2.ADANSB0 0008 9414h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	ANSB011	ANSB010	ANSB009	ANSB008	ANSB007	ANSB006	ANSB005	ANSB004	ANSB003	ANSB002	ANSB001	ANSB000
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ANSB000	A/D Conversion Channel Select	Select whether the voltages on AN200 to AN211 are to be converted or not. 0: Not selected for conversion 1: Selected for conversion	R/W
b1	ANSB001			R/W
b2	ANSB002			R/W
b3	ANSB003			R/W
b4	ANSB004			R/W
b5	ANSB005			R/W
b6	ANSB006			R/W
b7	ANSB007			R/W
b8	ANSB008			R/W
b9	ANSB009			R/W
b10	ANSB010			R/W
b11	ANSB011			R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The S12AD2.ADANSB0 register selects analog input channels for A/D conversion from among AN200 to AN211 in group B when group scan mode is selected. The S12AD2.ADANSB0 register is not used in any scan mode other than group scan mode.

ANSB0n Bit (A/D Conversion Channel Select) (n = 00 to 11)

The ANSB0n bit selects analog input channels for A/D conversion from among AN200 to AN211 in group B when group scan mode is selected. The S12AD2.ADANSB0 register is used for group scan mode only; not used for any other modes. The channels specified in group A (the S12AD2.ADANSA0 and S12AD2.ADANSA1 registers), the channels specified in group C (the S12AD2.ADANSC0 and S12AD2.ADANSC1 registers), and the channels corresponding to group A, selected with the S12AD2.ADCSR.DBLANS[4:0] bits in double trigger mode should be excluded as the channels to be selected and the number of channels to be set.

When performing A/D conversion of the temperature sensor output or internal reference voltage for group B in group scan mode, do not select analog input channels (set this register to 0000h).

The ANSB000 bit corresponds to AN200 and the ANSB011 bit corresponds to AN211.

The ANSB0n bit should be set while the S12AD2.ADCSR.ADST bit is 0.

42.2.7 A/D Channel Select Register B1 (ADANSB1)

Address(es): S12AD2.ADANSB1 0008 9416h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ANSB101	ANSB100
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ANSB100	A/D Conversion Channel Select	Select whether the voltages on AN216 to AN217 are to be converted or not. 0: Not selected for conversion 1: Selected for conversion	R/W
b1	ANSB101			R/W
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The S12AD2.ADANSB1 register selects analog input channels for A/D conversion from among AN216 to AN217 in group B when group scan mode is selected. The S12AD2.ADANSB1 register is not used in any scan mode other than group scan mode.

ANSB1n Bit (A/D Conversion Channel Select) (n = 00, 01)

The ANSB1n bit selects analog input channels for A/D conversion from among AN216 to AN217 in group B when group scan mode is selected. The S12AD2.ADANSB1 register is used for group scan mode only; not used for any other modes. The channels specified in group A (the S12AD2.ADANSA0 and S12AD2.ADANSA1 registers), the channels specified in group C (the S12AD2.ADANSC0 and S12AD2.ADANSC1 registers), and the channels corresponding to group A, selected with the S12AD2.ADCSR.DBLANS[4:0] bits in double trigger mode should be excluded as the channels to be selected and the number of channels to be set.

When performing A/D conversion of the temperature sensor output or internal reference voltage for group B in group scan mode, do not select analog input channels (set this register to 0000h).

The ANSB100 bit corresponds to AN216 and the ANSB101 bit corresponds to AN217.

The ANSB1n bit should be set while the S12AD2.ADCSR.ADST bit is 0.

42.2.8 A/D Channel Select Register C0 (ADANSC0)

(1) S12AD.ADANSC0

Address(es): S12AD.ADANSC0 0008 90D4h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	ANSC006	ANSC005	ANSC004	ANSC003	ANSC002	ANSC001	ANSC000
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ANSC000	A/D Conversion Channel Select	Select whether the voltages on AN000 to AN006 are to be converted or not. 0: Not selected for conversion 1: Selected for conversion	R/W
b1	ANSC001			R/W
b2	ANSC002			R/W
b3	ANSC003			R/W
b4	ANSC004			R/W
b5	ANSC005			R/W
b6	ANSC006			R/W
b15 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The S12AD.ADANSC0 register selects analog input channels for A/D conversion from among AN000 to AN006 in group C when group scan mode is selected. The S12AD.ADANSC0 register is not used in any scan mode other than group scan mode.

ANSC0n Bit (A/D Conversion Channel Select) (n = 00 to 06)

The ANSC0n bit selects analog input channels for A/D conversion from among AN000 to AN006 in group C when group scan mode is selected. The S12AD.ADANSC0 register is used for group scan mode only; not used for any other modes. The channels specified in group A (the S12AD.ADANSA0 register), the channels specified in group B (the S12AD.ADANSB0 register), and the channels corresponding to group A, selected with the S12AD.ADCSR.DBLANS[4:0] bits in double trigger mode should be excluded as the channels to be selected and the number of channels to be set.

The ANSC000 bit corresponds to AN000 and the ANSC006 bit corresponds to AN006.

The ANSC0n bit should be set while the S12AD.ADCSR.ADST bit is 0.

(2) S12AD1.ADANSC0

Address(es): S12AD1.ADANSC0 0008 92D4h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	ANSC003	ANSC002	ANSC001	ANSC000
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ANSC000	A/D Conversion Channel Select	Select whether the voltages on AN100 to AN103 are to be converted or not. 0: Not selected for conversion 1: Selected for conversion	R/W
b1	ANSC001			R/W
b2	ANSC002			R/W
b3	ANSC003			R/W
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The S12AD1.ADANSC0 register selects analog input channels for A/D conversion from among AN100 to AN103 in group C when group scan mode is selected. The S12AD1.ADANSC0 register is not used in any scan mode other than group scan mode.

ANSC0n Bit (A/D Conversion Channel Select) (n = 00 to 03)

The ANSC0n bit selects analog input channels for A/D conversion from among AN100 to AN103 in group C when group scan mode is selected. The S12AD1.ADANSC0 register is used for group scan mode only; not used for any other modes. The channels specified in group A (the S12AD1.ADANSA0 register), the channels specified in group B (the S12AD1.ADANSB0 register), and the channels corresponding to group A, selected with the S12AD1.ADCSR.DBLANS[4:0] bits in double trigger mode should be excluded as the channels to be selected and the number of channels to be set.

The ANSC000 bit corresponds to AN100 and the ANSC003 bit corresponds to AN103.

The ANSC0n bit should be set while the S12AD1.ADCSR.ADST bit is 0.

(3) S12AD2.ADANSC0

Address(es): S12AD2.ADANSC0 0008 94D4h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	ANSC011	ANSC010	ANSC009	ANSC008	ANSC007	ANSC006	ANSC005	ANSC004	ANSC003	ANSC002	ANSC001	ANSC000
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ANSC000	A/D Conversion Channel Select	Select whether the voltages on AN200 to AN211 are to be converted or not. 0: Not selected for conversion 1: Selected for conversion	R/W
b1	ANSC001			R/W
b2	ANSC002			R/W
b3	ANSC003			R/W
b4	ANSC004			R/W
b5	ANSC005			R/W
b6	ANSC006			R/W
b7	ANSC007			R/W
b8	ANSC008			R/W
b9	ANSC009			R/W
b10	ANSC010			R/W
b11	ANSC011			R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The S12AD2.ADANSC0 register selects analog input channels for A/D conversion from among AN200 to AN211 in group C when group scan mode is selected. The S12AD2.ADANSC0 register is not used in any scan mode other than group scan mode.

ANSC0n Bit (A/D Conversion Channel Select) (n = 00 to 11)

The ANSC0n bit selects analog input channels for A/D conversion from among AN200 to AN211 in group C when group scan mode is selected. The S12AD2.ADANSC0 register is used for group scan mode only; not used for any other modes. The channels specified in group A (the S12AD2.ADANSA0 and S12AD2.ADANSA1 registers), the channels specified in group B (the S12AD2.ADANSB0 and S12AD2.ADANSB1 registers), and the channels corresponding to group A, selected with the S12AD2.ADCSR.DBLANS[4:0] bits in double trigger mode should be excluded as the channels to be selected and the number of channels to be set.

When performing A/D conversion of the temperature sensor output or internal reference voltage or for group C in group scan mode, do not select analog input channels (set this register to 0000h).

The ANSC000 bit corresponds to AN200 and the ANSC011 bit corresponds to AN211.

The ANSC0n bit should be set while the S12AD2.ADCSR.ADST bit is 0.

42.2.9 A/D Channel Select Register C1 (ADANSC1)

Address(es): S12AD2.ADANSC1 0008 94D6h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ANSC101	ANSC100
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ANSC100	A/D Conversion Channel Select	Select whether the voltages on AN216 to AN217 are to be converted or not. 0: Not selected for conversion 1: Selected for conversion	R/W
b1	ANSC101			R/W
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The S12AD2.ADANSC1 register selects analog input channels for A/D conversion from among AN216 to AN217 in group C when group scan mode is selected. The S12AD2.ADANSC1 register is not used in any scan mode other than group scan mode.

ANSC1n Bit (A/D Conversion Channel Select) (n = 00, 01)

The ANSC1n bit selects analog input channels for A/D conversion from among AN216 to AN217 in group C when group scan mode is selected. The S12AD2.ADANSC1 register is used for group scan mode only; not used for any other modes. The channels specified in group A (the S12AD2.ADANSA0 and S12AD2.ADANSA1 registers), the channels specified in group B (the S12AD2.ADANSB0 and S12AD2.ADANSB1 registers), and the channels corresponding to group A, selected with the S12AD2.ADCSR.DBLANS[4:0] bits in double trigger mode should be excluded as the channels to be selected and the number of channels to be set.

When performing A/D conversion of the temperature sensor output or internal reference voltage or for group C in group scan mode, do not select analog input channels (set this register to 0000h).

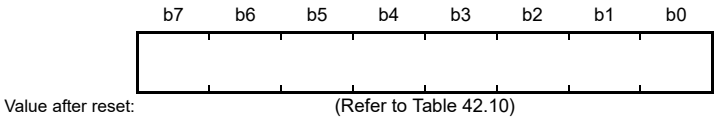
The ANSC100 bit corresponds to AN216 and the ANSC101 bit corresponds to AN217.

The ANSC1n bit should be set while the S12AD2.ADCSR.ADST bit is 0.

42.2.10 A/D Channel Conversion Order Setting Register n (ADSCSn) (n = 0 to 13)

(1) S12AD.ADSCSn (n = 0 to 6)

Address(es): S12AD.ADSCS0 0008 91C0h, S12AD.ADSCS1 0008 91C1h, S12AD.ADSCS2 0008 91C2h, S12AD.ADSCS3 0008 91C3h, S12AD.ADSCS4 0008 91C4h, S12AD.ADSCS5 0008 91C5h, S12AD.ADSCS6 0008 91C6h



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	—	Specify the order of channels for conversion by the 12-bit A/D converter. The channel with the number set in an ADSCSn register is the (n+1)th to be converted (refer to Table 42.10).	R/W

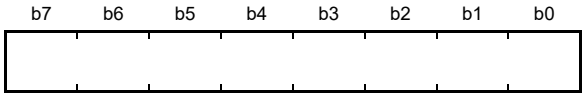
The ADSCSn register specifies the order of channels for conversion by the 12-bit A/D converter unit 0.
The order of conversion is set in the order AN000, AN001, ..., AN006 after a reset.
To change the order, change the values in the ADSCSn registers.
To use the channel-dedicated sample-and-hold circuit for AN000 to AN002, set the corresponding channel in any of the ADSCS0, ADSCS1, or ADSCS2 register. When setting them to the ADSCS3 register, the results of conversion for the channels are not guaranteed.
If a specified channel is not selected for conversion in the channel selection registers (ADANSA0, ADANSB0, and ADANSC0), A/D conversion of the corresponding channel does not proceed. Set the channels for conversion in order starting from the ADSCS0 register, and set all the channels other than those above in the rest of the registers. Do not set the same value to plural registers.

Table 42.10 Relationship Between the Value After Reset of the A/D Channel Conversion Order Setting Register n (ADSCSn) and Conversion Order

Register Symbol	Order of Conversion	Value After Reset	Channel Numbers that Can be Set:
ADSCS0	First	00h	00h to 06h (for AN000 to AN006)
ADSCS1	Second	01h	
ADSCS2	Third	02h	
ADSCS3	Fourth	03h	
ADSCS4	Fifth	04h	
ADSCS5	Sixth	05h	
ADSCS6	Seventh	06h	

(2) S12AD1.ADSCSn (n = 0 to 3)

Address(es): S12AD1.ADSCS0 0008 93C0h, S12AD1.ADSCS1 0008 93C1h, S12AD1.ADSCS2 0008 93C2h, S12AD1.ADSCS3 0008 93C3h



Value after reset: (Refer to Table 42.11)

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	—	Specify the order of channels for conversion by the 12-bit A/D converter. The channel with the number set in an ADSCSn register is the (n+1)th to be converted (refer to Table 42.11).	R/W

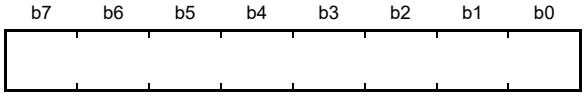
The ADSCSn register specifies the order of channels for conversion by the 12-bit A/D converter unit 1.
The order of conversion is set in the order AN100, AN101, ..., AN103 after a reset.
To change the order, change the values in the ADSCSn registers.
To use the channel-dedicated sample-and-hold circuit for AN100 to AN102, set the corresponding channel in any of the ADSCS0, ADSCS1, or ADSCS2 register. When setting them to the ADSCS3 register, the results of conversion for the channels are not guaranteed.
If a specified channel is not selected for conversion in the channel selection registers (ADANSA0, ADANSB0, and ADANSC0), A/D conversion of the corresponding channel does not proceed. Set the channels for conversion in order starting from the ADSCS0 register, and set all the channels other than those above in the rest of the registers. Do not set the same value to plural registers.

Table 42.11 Relationship Between the Value After Reset of the A/D Channel Conversion Order Setting Register n (ADSCSn) and Conversion Order

Register Symbol	Order of Conversion	Value After Reset	Channel Numbers that Can be Set:
ADSCS0	First	00h	00h to 03h (for AN100 to AN103)
ADSCS1	Second	01h	
ADSCS2	Third	02h	
ADSCS3	Fourth	03h	

(3) S12AD2.ADSCSn (n = 0 to 13)

Address(es): S12AD2.ADSCS0 0008 95C0h, S12AD2.ADSCS1 0008 95C1h, S12AD2.ADSCS2 0008 95C2h, S12AD2.ADSCS3 0008 95C3h, S12AD2.ADSCS4 0008 95C4h, S12AD2.ADSCS5 0008 95C5h, S12AD2.ADSCS6 0008 95C6h, S12AD2.ADSCS7 0008 95C7h, S12AD2.ADSCS8 0008 95C8h, S12AD2.ADSCS9 0008 95C9h, S12AD2.ADSCS10 0008 95CAh, S12AD2.ADSCS11 0008 95CBh, S12AD2.ADSCS12 0008 95D0h, S12AD2.ADSCS13 0008 95D1h



Value after reset: (Refer to Table 42.12)

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	—	Specify the order of channels for conversion by the 12-bit A/D converter. The channel with the number set in an ADSCSn register is the (n+1)th to be converted (refer to Table 42.12).	R/W

The ADSCSn register specifies the order of channels for conversion by the 12-bit A/D converter unit 2.
The order of conversion is set in the order AN200, AN201, ..., AN211, AN216, and AN217 after a reset.
To change the order, change the values in the ADSCSn registers.
If a specified channel is not selected for conversion in the channel selection registers (ADANSA0/1, ADANSB0/1, and ADANSC0/1), A/D conversion of the corresponding channel does not proceed. Set the channels for conversion in order starting from the ADSCS0 register, and set all the channels other than those above in the rest of the registers. Do not set the same value to plural registers.
Furthermore, the places of the temperature sensor output and internal reference voltage in the order of conversion are fixed.

Table 42.12 Relationship Between the Value After Reset of the A/D Channel Conversion Order Setting Register n (ADSCSn) and Conversion Order

Register Symbol	Order of Conversion	Value After Reset	Channel Numbers that Can be Set:
ADSCS0	First	00h	00h to 0Bh, 10h, and 11h (for AN200 to AN211, AN216, AN217)
ADSCS1	Second	01h	
ADSCS2	Third	02h	
ADSCS3	Fourth	03h	
ADSCS4	Fifth	04h	
ADSCS5	Sixth	05h	
ADSCS6	Seventh	06h	
ADSCS7	Eighth	07h	
ADSCS8	Ninth	08h	
ADSCS9	Tenth	09h	
ADSCS10	Eleventh	0Ah	
ADSCS11	Twelfth	0Bh	
ADSCS12	Thirteenth	10h	
ADSCS13	Fourteenth	11h	

42.2.11 A/D-Converted Value Addition/Average Function Channel Select Register 0 (ADADS0)

(1) S12AD.ADADS0

Address(es): S12AD.ADADS0 0008 9008h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	ADS006	ADS005	ADS004	ADS003	ADS002	ADS001	ADS000
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ADS000	A/D-Converted Value Addition/Average Channel Select	Set an A/D-converted value addition/average mode for AN000 to AN006. 0: A/D-converted value addition/average mode is disabled 1: A/D-converted value addition/average mode is enabled	R/W
b1	ADS001			R/W
b2	ADS002			R/W
b3	ADS003			R/W
b4	ADS004			R/W
b5	ADS005			R/W
b6	ADS006			R/W
b15 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The S12AD.ADADS0 register selects channels AN000 to AN006 on which A/D conversion is performed successively 2, 3, 4, or 16 times and then converted values are added (integrated) or averaged.

ADS0n Bit (A/D-Converted Value Addition/Average Channel Select) (n = 00 to 06)

When the ADS0n bit of the number that is the same as that of A/D-converted channel selected by the S12AD.ADANSA0.ANSA0n bit or the S12AD.ADCSR.DBLANS[4:0] bits and the S12AD.ADANSB0.ANSB0n bit and the S12AD.ADANSC0.ANSC0n bit is set to 1, A/D conversion of analog input of the selected channels is performed successively 2, 3, 4, or 16 times that is set with the S12AD.ADADC.ADC[2:0] bits.

When the S12AD.ADADC.AVEE bit is 0, the value obtained by addition (integration) is stored in the A/D data register.

When the S12AD.ADADC.AVEE bit is 1, the mean value of the results obtained by addition (integration) is stored in the A/D data register.

An A/D conversion channel with addition/average mode disabled executes a regular one-time conversion and stores the value in the A/D data register.

The ADS0n bit should be set while the S12AD.ADCSR.ADST bit is 0.

(2) S12AD1.ADADS0

Address(es): S12AD1.ADADS0 0008 9208h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	ADS003	ADS002	ADS001	ADS000
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ADS000	A/D-Converted Value Addition/Average Channel Select	Set an A/D-converted value addition/average mode for AN100 to AN103. 0: A/D-converted value addition/average mode is disabled 1: A/D-converted value addition/average mode is enabled	R/W
b1	ADS001			R/W
b2	ADS002			R/W
b3	ADS003			R/W
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The S12AD1.ADADS0 register selects channels AN100 to AN103 on which A/D conversion is performed successively 2, 3, 4, or 16 times and then converted values are added (integrated) or averaged.

ADS0n Bit (A/D-Converted Value Addition/Average Channel Select) (n = 00 to 03)

When the ADS0n bit of the number that is the same as that of A/D-converted channel selected by the S12AD1.ADANSA0.ANSA0n bit or the S12AD1.ADCSR.DBLANS[4:0] bits and the S12AD1.ADANSB0.ANSB0n bit and the S12AD1.ADANSC0.ANSC0n bit is set to 1, A/D conversion of analog input of the selected channels is performed successively 2, 3, 4, or 16 times that is set with the S12AD1.ADADC.ADC[2:0] bits.

When the S12AD1.ADADC.AVEE bit is 0, the value obtained by addition (integration) is stored in the A/D data register.

When the S12AD1.ADADC.AVEE bit is 1, the mean value of the results obtained by addition (integration) is stored in the A/D data register.

An A/D conversion channel with addition/average mode disabled executes a regular one-time conversion and stores the value in the A/D data register.

The ADS0n bit should be set while the S12AD1.ADCSR.ADST bit is 0.

(3) S12AD2.ADADS0

Address(es): S12AD2.ADADS0 0008 9408h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	ADS011	ADS010	ADS009	ADS008	ADS007	ADS006	ADS005	ADS004	ADS003	ADS002	ADS001	ADS000
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ADS000	A/D-Converted Value Addition/Average Channel Select	Set an A/D-converted value addition/average mode for AN200 to AN211. 0: A/D-converted value addition/average mode is disabled 1: A/D-converted value addition/average mode is enabled	R/W
b1	ADS001			R/W
b2	ADS002			R/W
b3	ADS003			R/W
b4	ADS004			R/W
b5	ADS005			R/W
b6	ADS006			R/W
b7	ADS007			R/W
b8	ADS008			R/W
b9	ADS009			R/W
b10	ADS010			R/W
b11	ADS011			R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The S12AD2.ADADS0 register selects channels AN200 to AN211 on which A/D conversion is performed successively 2, 3, 4, or 16 times and then converted values are added (integrated) or averaged.

ADS0n Bit (A/D-Converted Value Addition/Average Channel Select) (n = 00 to 11)

When the ADS0n bit of the number that is the same as that of A/D-converted channel selected by the S12AD2.ADANSA0.ANSA0n bit or the S12AD2.ADCSR.DBLANS[4:0] bits and the S12AD2.ADANSB0.ANSB0n bit and the S12AD2.ADANSC0.ANSC0n bit is set to 1, A/D conversion of analog input of the selected channels is performed successively 2, 3, 4, or 16 times that is set with the S12AD2.ADADC.ADC[2:0] bits.

When the S12AD2.ADADC.AVEE bit is 0, the value obtained by addition (integration) is stored in the A/D data register.

When the S12AD2.ADADC.AVEE bit is 1, the mean value of the results obtained by addition (integration) is stored in the A/D data register.

An A/D conversion channel with addition/average mode disabled executes a regular one-time conversion and stores the value in the A/D data register.

The ADS0n bit should be set while the S12AD2.ADCSR.ADST bit is 0.

Figure 42.5 shows a scanning operation sequence in which both the S12AD2.ADADS0.ADS002 and S12AD2.ADADS0.ADS006 bits are set to 1.
It is assumed that addition mode is selected (S12AD2.ADADC.AVEE bit = 0) in continuous scan mode (S12AD2.ADCSR.ADCS[1:0] bits = 10b), the addition count is set to three times (S12AD2.ADADC.ADC[2:0] bits = 011b), and channels AN200 to AN207 are selected (S12AD2.ADANSA0 register = 00FFh). The conversion process begins with AN200. The AN202 conversion is performed successively four times (add three times), and the added (integrated) value is stored in A/D data register 2. After that the AN203 conversion is started. The AN206 conversion is performed successively four times and the added (integrated) value is stored in A/D data register 6. After conversion of AN207, the conversion operation is once again performed in the same sequence from AN200.

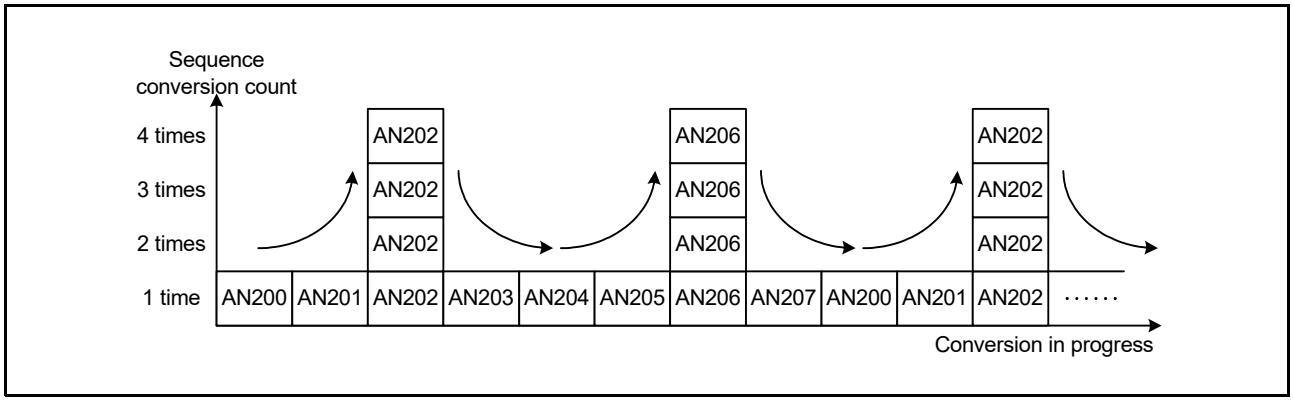


Figure 42.5 Scan Conversion Sequence with S12AD2.ADADC.ADC[2:0] = 011b, S12AD2.ADADC.AVEE = 0, S12AD2.ADADS0.ADS002 = 1, and S12AD2.ADADS0.ADS006 = 1

42.2.12 A/D-Converted Value Addition/Average Function Channel Select Register 1 (ADADS1)

Address(es): S12AD2.ADADS1 0008 940Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADS101	ADS100
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ADS100	A/D-Converted Value Addition/Average Channel Select	Set an A/D-converted value addition/average mode for AN216 to AN217. 0: A/D-converted value addition/average mode is disabled 1: A/D-converted value addition/average mode is enabled	R/W
b1	ADS101			R/W
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The S12AD2.ADADS1 register selects channels AN216 to AN217 on which A/D conversion is performed successively 2, 3, 4, or 16 times and then converted values are added (integrated) or averaged.

ADS1n Bit (A/D-Converted Value Addition/Average Channel Select) (n = 00, 01)

When the ADS1n bit of the number that is the same as that of A/D-converted channel selected by the S12AD2.ADANSA1.ANSA1n bit or the S12AD2.ADCSR.DBLANS[4:0] bits, the S12AD2.ADANSB1.ANSB1n bit, and the S12AD2.ADANSC1.ANSC1n bit is set to 1, A/D conversion of analog input of the selected channels is performed successively 2, 3, 4, or 16 times that is set with the S12AD2.ADADC.ADC[2:0] bits. When the S12AD2.ADADC.AVEE bit is 0, the value obtained by addition (integration) is stored in the A/D data register. When the S12AD2.ADADC.AVEE bit is 1, the mean value of the results obtained by addition (integration) is stored in the A/D data register.

An A/D conversion channel with addition/average mode disabled executes a regular one-time conversion and stores the value in the A/D data register.

The ADS1n bit should be set while the S12AD2.ADCSR.ADST bit is 0.

42.2.13 A/D-Converted Value Addition/Average Count Select Register (ADADC)

Address(es): S12AD.ADADC 0008 900Ch, S12AD1.ADADC 0008 920Ch, S12AD2.ADADC 0008 940Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	AVEE	—	—	—	—	ADC[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	ADC[2:0]	Addition Count Select	b2 b0 0 0 0: 1-time conversion (no addition; same as normal conversion) 0 0 1: 2-time conversion (addition once) 0 1 0: 3-time conversion (addition twice)*1 0 1 1: 4-time conversion (addition three times) 1 0 1: 16-time conversion (addition 15 times)*1 Settings other than above are prohibited.	R/W
b6 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	AVEE	Average Mode Enable	0: Addition mode is selected. 1: Average mode is selected.	R/W

Note 1. The AVEE bit is enabled only when 2-time or 4-time conversion is selected. When average mode is selected (ADADC.AVEE bit = 1), do not set 3-time conversion (ADADC.ADC[2:0] bits = 010b) nor 16-time conversion (ADADC.ADC[2:0] bits = 101b).

When the A/D-converted value addition/average modes are enabled for A/D conversion of the channels, temperature sensor output, or internal reference voltage, the number of addition and whether they are to be added or averaged is specified in the ADADC register.

ADC[2:0] Bits (Addition Count Select)

The number of addition from channels including those channels selected in double-trigger mode (by the ADCSR.DBLANS[4:0] bits), temperature sensor output, or internal reference voltage for which A/D conversion and addition or averaging is enabled, is selected by the setting of the ADC[2:0] bits.
The ADC[2:0] bits should be set while the ADCSR.ADST bit is 0.

AVEE Bit (Average Mode Enable)

The AVEE bit selects whether the results of A/D conversion from channels, including those selected in double-trigger mode (by the DBLANS[4:0] bits), temperature sensor output, or internal reference voltage are to be averaged or added when A/D conversion and addition/average modes are enabled.
When average mode is selected by setting the AVEE bit to 1, do not set the addition count to one time (ADC[2:0] bits = 000b), three times (ADC[2:0] bits = 010b), or 16 times (ADC[2:0] bits = 101b). The mean value of 1-time, 3-time, and 16-time conversion cannot be obtained.
The AVEE bit should be set while the ADCSR.ADST bit is 0.

42.2.14 A/D Control Extended Register (ADCER)

Address(es): S12AD.ADCER 0008 900Eh, S12AD1.ADCER 0008 920Eh, S12AD2.ADCER 0008 940Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ADRFMT	—	—	—	DIAGM	DIAGLD	DIAGVAL[1:0]		—	ASE	ACE	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	ACE	A/D Data Register Automatic Clearing Enable	0: Disables automatic clearing. 1: Enables automatic clearing.	R/W
b6	ASE	A/D Data Register Automatic Setting Enable	0: Disables automatic setting. 1: Enables automatic setting.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b9, b8	DIAGVAL[1:0]	Self-Diagnosis Conversion Voltage Select	b9 b8 0 0: Setting prohibited in self-diagnosis voltage fixed mode 0 1: Uses the voltage of 0 V for self-diagnosis. 1 0: Uses the voltage of 1/2 × AVCC for self-diagnosis. 1 1: Uses the voltage of AVCC for self-diagnosis.	R/W
b10	DIAGLD	Self-Diagnosis Mode Select	0: Rotation mode for self-diagnosis voltage 1: Fixed mode for self-diagnosis voltage	R/W
b11	DIAGM	Self-Diagnosis Enable	0: Disables self-diagnosis of 12-bit A/D converter. 1: Enables self-diagnosis of 12-bit A/D converter.	R/W
b14 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	ADRFMT	A/D Data Register Format Select	0: Right-justified is selected for the A/D data register format. 1: Left-justified is selected for the A/D data register format.	R/W

Note 1. If both the ACE and the ASE bits are set to 1, neither automatic clearing nor automatic setting is performed.

The ADCER register sets self-diagnosis mode, format of A/D data registers y (ADDRy), and automatic clearing of A/D data registers.

ACE Bit (A/D Data Register Automatic Clearing Enable)

The ACE bit enables or disables automatic clearing (all 0) of the ADDRy, ADRD, ADDBLDR, ADDBLDRA, ADDBLDRB, ADTSDR, or ADOCDR register after any of these registers have been read by the CPU, DTC, or DMAC. Automatic clearing of the A/D data register is enabled to detect a failure which has not been updated in the A/D data register.

ASE Bit (A/D Data Register Automatic Setting Enable)

The ACE bit enables or disables automatic setting (all 1) of the ADDRy, ADRD, ADDBLDR, ADDBLDRA, ADDBLDRB, ADTSDR, or ADOCDR register after any of these registers have been read by the CPU, DTC, or DMAC. Automatic setting of the A/D data register is enabled to detect a failure which has not been updated in the A/D data register.

DIAGVAL[1:0] Bits (Self-Diagnosis Conversion Voltage Select)

The DIAGVAL[1:0] bits select the voltage value used in self-diagnosis voltage fixed mode. For details, refer to the descriptions of the DIAGLD bit.
Self-diagnosis should not be executed by setting the DIAGLD bit to 1 when the DIAGVAL[1:0] bits are set to 00b.

DIAGLD Bit (Self-Diagnosis Mode Select)

The DIAGLD bit selects whether the three voltage values are rotated or the fixed voltage is used in self-diagnosis.

Setting this bit (DIAGLD) to 0 allows conversion of the voltages in rotation mode where 0 V, 1/2 × AVCC, and AVCC are converted in that order. When self-diagnosis rotation mode is selected after a reset, self-diagnosis is performed from 0 V. When self-diagnosis voltage fixed mode is selected, the fixed voltage specified by the DIAGVAL[1:0] bits is converted. In self-diagnosis voltage rotation mode, the self-diagnosis voltage value does not return to 0 when scan conversion is completed. When scan conversion is restarted, therefore, rotation starts at the voltage value following the previous value. If fixed mode is switched to rotation mode, rotation starts at the fixed voltage value.
The DIAGLD bit should be set while the ADCSR.ADST bit is 0.

DIAGM Bit (Self-Diagnosis Enable)

The DIAGM bit enables or disables self-diagnosis.
Self-diagnosis is used to detect a failure of the 12-bit A/D converter. Specifically, one from among the internally generated voltages 0, 1/2 × AVCC, or AVCC is selected for conversion. When conversion is completed, information on the converted voltage and the conversion result is stored into the self-diagnosis data register (ADRD). The ADRD register can then be read out by software to determine whether the conversion result falls within the normal range (normal) or not (abnormal). Self-diagnosis is executed once at the beginning of each scan, and one of the three voltages is converted. When self-diagnosis is selected in group scan mode, self-diagnosis is separately executed in groups A, B, and C.
The DIAGM bit should be set while the ADCSR.ADST bit is 0.
When performing the self-diagnosis, A/D conversion of the temperature sensor output and internal reference voltage is prohibited regardless of scanning groups.

ADRFMT Bit (A/D Data Register Format Select)

The ADRFMT bit specifies right-justified or left-justified for the data to be stored in the ADDRy, ADDBLDR, ADDBLDRA, ADDBLDRB, ADTSDR, ADOCDR, ADRD, ADCMPDR0, ADCMPDR1, ADWINLLB, or ADWINULB register.
The ADRFMT bit should be set while the ADCSR.ADST bit is 0.

For details on the format of each data register, refer to section 42.2.1, A/D Data Registers y (ADDRy) (y = 0 to 11, 16, 17), A/D Data Duplication Register (ADDBLDR), A/D Data Duplication Register A (ADDBLDRA), A/D Data Duplication Register B (ADDBLDRB), A/D Temperature Sensor Data Register (ADTSDR), A/D Internal Reference Voltage Data Register (ADOCDR), section 42.2.2, A/D Self-Diagnosis Data Register (ADRD), section 42.2.33, A/D Comparison Function Window A Lower Level Setting Register (ADCMPDR0), section 42.2.34, A/D Comparison Function Window A Upper Level Setting Register (ADCMPDR1), section 42.2.40, A/D Comparison Function Window B Lower Level Setting Register (ADWINLLB), and section 42.2.41, A/D Comparison Function Window B Upper Level Setting Register (ADWINULB).

42.2.15 A/D Conversion Start Trigger Select Register (ADSTRGR)

Address(es): S12AD.ADSTRGR 0008 9010h, S12AD1.ADSTRGR 0008 9210h, S12AD2.ADSTRGR 0008 9410h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	TRSA[6:0]							—	TRSB[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	TRSB[6:0]	Group B A/D Conversion Start Trigger Select	Select the A/D conversion start trigger for group B in group scan mode.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b14 to b8	TRSA[6:0]	A/D Conversion Start Trigger Select	Select the A/D conversion start trigger in single scan mode and continuous mode. In group scan mode, the A/D conversion start trigger for group A is selected.	R/W
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

The ADSTRGR register selects the A/D conversion start trigger.

TRSB[6:0] Bits (Group B A/D Conversion Start Trigger Select)

The TRSB[6:0] bits select the trigger to start scanning of the analog input selected in group B. The TRSB[6:0] bits require to be set only in group scan mode and are not used in any other scan mode. For the scan conversion start trigger for group B, setting a software trigger or an asynchronous trigger is prohibited. Therefore, the TRSB[6:0] bits should be set to the value other than 00h and the ADCSR.TRGE bit should be set to 1 in group scan mode.

When two groups are selected (ADGCTRGR.GRCE bit = 0) during group priority operation in group scan mode, setting the ADGSPCR.GBRP bit to 1 allows group B to continuously operate in single scan mode. When setting the GBRP bit to 1, set the TRSB[6:0] bits to 3Fh or 7Fh. Note that the issuance period of trigger for A/D conversion must be more than or equal to the actual scan conversion time (t_{SCAN}). If the issuance period is less than t_{SCAN}, A/D conversion by the trigger may have no effect.

When selecting a trigger from the MTU operated on a PCLKC cycle as an A/D conversion start trigger, a delay of the period for synchronization processing occurs. Refer to section 42.3.7, Analog Input Sampling Time and Scan Conversion Time for details.

Table 42.13 lists the A/D conversion startup sources selected by the TRSB[6:0] bits.

TRSA[6:0] Bits (A/D Conversion Start Trigger Select)

The TRSA[6:0] bits select the trigger to start A/D conversion in single scan mode and continuous scan mode. In group scan mode, the trigger to start scanning of the analog input selected in group A is selected. When performing scanning in group scan mode or double trigger mode, set the ADCSR.TRGE bit to 1 and set the ADCSR.EXTRG bit to 0.

- When using the A/D conversion startup source of a synchronous trigger, set the ADCSR.TRGE bit to 1 and set the ADCSR.EXTRG bit to 0.
- When using the asynchronous trigger, set the ADCSR.TRGE bit to 1 and set the ADCSR.EXTRG bit to 1.
- Software trigger (ADCSR.ADST) is enabled regardless of the settings of the ADCSR.TRGE bit, the ADCSR.EXTRG bit, and the TRSA[6:0] bits.

Note that the issuance period of trigger for A/D conversion must be more than or equal to the actual scan conversion time (t_{SCAN}). If the issuance period is less than t_{SCAN}, A/D conversion by a trigger may have no effect. When selecting a trigger from the MTU operated on a PCLKC cycle as an A/D conversion start trigger, a delay of the period for synchronization processing occurs. Refer to section 42.3.7, Analog Input Sampling Time and Scan Conversion

Time for details.

Table 42.14 lists the selection of A/D conversion start sources selected by the TRSA[6:0] bits.

Table 42.13 Selection of A/D Trigger Sources by the TRSB[6:0] Bits (1/3)

Module	Source	Remarks	TRSB[6]	TRSB[5]	TRSB[4]	TRSB[3]	TRSB[2]	TRSB[1]	TRSB[0]
Trigger source deselection state			0	1	1	1	1	1	1
			1	1	1	1	1	1	1
MTU	TRGA0N	Compare match/input capture in MTU0.TGRA	0	0	0	0	0	0	1
	TRGA1N	Compare match/input capture in MTU1.TGRA	0	0	0	0	0	1	0
	TRGA2N	Compare match/input capture in MTU2.TGRA	0	0	0	0	0	1	1
	TRGA3N	Compare match/input capture in MTU3.TGRA	0	0	0	0	1	0	0
	TRGA4N	Compare match/input capture in MTU4.TGRA or underflow (trough) of MTU4.TCNT in complementary PWM mode	0	0	0	0	1	0	1
	TRGA6N	Compare match/input capture in MTU6.TGRA	0	0	0	0	1	1	0
	TRGA7N	Compare match/input capture in MTU7.TGRA or underflow (trough) of MTU7.TCNT in complementary PWM mode	0	0	0	0	1	1	1
	TRG0N	Compare match in MTU0.TGRE	0	0	0	1	0	0	0
	TRG4AN	Compare match between MTU4.TADCORA and MTU4.TCNT	0	0	0	1	0	0	1
	TRG4BN	Compare match between MTU4.TADCORB and MTU4.TCNT	0	0	0	1	0	1	0
	TRG4AN or TRG4BN	Compare match between MTU4.TADCORA and MTU4.TCNT, or compare match between MTU4.TADCORB and MTU4.TCNT	0	0	0	1	0	1	1
	TRG4ABN	Compare match between MTU4.TADCORA and MTU4.TCNT, and compare match between MTU4.TADCORB and MTU4.TCNT (when interrupt skipping function 2 is used)	0	0	0	1	1	0	0
	TRG7AN	Compare match between MTU7.TADCORA and MTU7.TCNT	0	0	0	1	1	0	1
	TRG7BN	Compare match between MTU7.TADCORB and MTU7.TCNT	0	0	0	1	1	1	0
	TRG7AN or TRG7BN	Compare match between MTU7.TADCORA and MTU7.TCNT, or compare match between MTU7.TADCORB and MTU7.TCNT	0	0	0	1	1	1	1
	TRG7ABN	Compare match between MTU7.TADCORA and MTU7.TCNT, and compare match between MTU7.TADCORB and MTU7.TCNT (when interrupt skipping function 2 is used)	0	0	1	0	0	0	0
	TRGA9N	Compare match/input capture in MTU9.TGRA	0	0	1	0	0	1	1
	TRG9N	Compare match in MTU9.TGRE	0	0	1	0	1	0	0
	TRGA0N or TRG0N	Compare match/input capture in MTU0.TGRA, or compare match in MTU0.TGRE	0	0	1	1	0	0	1
	TRGA9N or TRG9N	Compare match/input capture in MTU9.TGRA, or compare match in MTU9.TGRE	0	0	1	1	0	1	0
	TRGA0N or TRGA9N	Compare match/input capture in MTU0.TGRA, or compare match/input capture in MTU9.TGRA	0	0	1	1	0	1	1
	TRG0N or TRG9N	Compare match in MTU0.TGRE, or compare match in MTU9.TGRE	0	0	1	1	1	0	0
	TRG9AEN	Compare match/input capture in MTU9.TGRA, and compare match in MTU9.TGRE	0	1	0	0	0	0	1
	TRG0AEN	Compare match/input capture in MTU0.TGRA, and compare match in MTU0.TGRE	0	1	0	0	0	1	0

Table 42.13 Selection of A/D Trigger Sources by the TRSB[6:0] Bits (2/3)

Module	Source	Remarks	TRSB[6]	TRSB[5]	TRSB[4]	TRSB[3]	TRSB[2]	TRSB[1]	TRSB[0]
MTU	TRGA09N	Compare match/input capture in MTU0.TGRA, and compare match/input capture in MTU9.TGRA	0	1	0	0	0	1	1
	TRG09N	Compare match in MTU0.TGRE, and compare match in MTU9.TGRE	0	1	0	0	1	0	0
GPTW	GTADTRA0N	Compare match in GPTW0.GTADTRA	1	0	0	0	0	0	0
	GTADTRB0N	Compare match in GPTW0.GTADTRB	1	0	0	0	0	0	1
	GTADTRA1N	Compare match in GPTW1.GTADTRA	1	0	0	0	0	1	0
	GTADTRB1N	Compare match in GPTW1.GTADTRB	1	0	0	0	0	1	1
	GTADTRA2N	Compare match in GPTW2.GTADTRA	1	0	0	0	1	0	0
	GTADTRB2N	Compare match in GPTW2.GTADTRB	1	0	0	0	1	0	1
	GTADTRA3N	Compare match in GPTW3.GTADTRA	1	0	0	0	1	1	0
	GTADTRB3N	Compare match in GPTW3.GTADTRB	1	0	0	0	1	1	1
	GTADTRA0N or GTADTRB0N	Compare match in GPTW0.GTADTRA, or compare match in GPTW0.GTADTRB	1	0	0	1	0	0	0
	GTADTRA1N or GTADTRB1N	Compare match in GPTW1.GTADTRA, or compare match in GPTW1.GTADTRB	1	0	0	1	0	0	1
	GTADTRA2N or GTADTRB2N	Compare match in GPTW2.GTADTRA, or compare match in GPTW2.GTADTRB	1	0	0	1	0	1	0
	GTADTRA3N or GTADTRB3N	Compare match in GPTW3.GTADTRA, or compare match in GPTW3.GTADTRB	1	0	0	1	0	1	1
	GTADTRA4N	Compare match in GPTW4.GTADTRA	1	0	0	1	1	0	0
	GTADTRB4N	Compare match in GPTW4.GTADTRB	1	0	0	1	1	0	1
	GTADTRA5N	Compare match in GPTW5.GTADTRA	1	0	0	1	1	1	0
	GTADTRB5N	Compare match in GPTW5.GTADTRB	1	0	0	1	1	1	1
	GTADTRA6N	Compare match in GPTW6.GTADTRA	1	0	1	0	0	0	0
	GTADTRB6N	Compare match in GPTW6.GTADTRB	1	0	1	0	0	0	1
	GTADTRA7N	Compare match in GPTW7.GTADTRA	1	0	1	0	0	1	0
	GTADTRB7N	Compare match in GPTW7.GTADTRB	1	0	1	0	0	1	1
	GTADTRA4N or GTADTRB4N	Compare match in GPTW4.GTADTRA, or compare match in GPTW4.GTADTRB	1	0	1	0	1	0	0
	GTADTRA5N or GTADTRB5N	Compare match in GPTW5.GTADTRA, or compare match in GPTW5.GTADTRB	1	0	1	0	1	0	1
	GTADTRA6N or GTADTRB6N	Compare match in GPTW6.GTADTRA, or compare match in GPTW6.GTADTRB	1	0	1	0	1	1	0
	GTADTRA7N or GTADTRB7N	Compare match in GPTW7.GTADTRA, or compare match in GPTW7.GTADTRB	1	0	1	0	1	1	1
TMR	TMTRG0AN_0	Compare match between TMR0.TCORA and TMR0.TCNT	0	0	1	1	1	0	1
	TMTRG0AN_1	Compare match between TMR2.TCORA and TMR2.TCNT	0	0	1	1	1	1	0
	TMTRG0AN_2	Compare match between TMR4.TCORA and TMR4.TCNT	0	0	1	1	1	1	1
	TMTRG0AN_3	Compare match between TMR6.TCORA and TMR6.TCNT	0	1	0	0	0	0	0

Table 42.13 Selection of A/D Trigger Sources by the TRSB[6:0] Bits (3/3)

Module	Source	Remarks	TRSB[6]	TRSB[5]	TRSB[4]	TRSB[3]	TRSB[2]	TRSB[1]	TRSB[0]
ELC	ELCTRG00N*1 ELCTRG10N*2 ELCTRG20N*3	A/D Startup source 0 from ELC	0	1	1	0	0	1	0
	ELCTRG01N*1 ELCTRG11N*2 ELCTRG21N*3	A/D Startup source 1 from ELC	0	1	1	0	0	1	1
	ELCTRG00N or ELCTRG01N*1 ELCTRG10N or ELCTRG11N*2 ELCTRG20N or ELCTRG21N*3	A/D Startup source 0 from ELC, or A/D Startup source 1 from ELC	0	1	1	1	0	1	0

Note 1. Unit 0
Note 2. Unit 1
Note 3. Unit 2

Table 42.14 Selection of A/D Trigger Sources by the TRSA[6:0] Bits (1/3)

Module	Source	Remarks	TRSA[6]	TRSA[5]	TRSA[4]	TRSA[3]	TRSA[2]	TRSA[1]	TRSA[0]
Trigger source deselection state			0	1	1	1	1	1	1
			1	1	1	1	1	1	1
External pin	ADTRGn#	Trigger input pin	0	0	0	0	0	0	0
MTU	TRGA0N	Compare match/input capture in MTU0.TGRA	0	0	0	0	0	0	1
	TRGA1N	Compare match/input capture in MTU1.TGRA	0	0	0	0	0	1	0
	TRGA2N	Compare match/input capture in MTU2.TGRA	0	0	0	0	0	1	1
	TRGA3N	Compare match/input capture in MTU3.TGRA	0	0	0	0	1	0	0
	TRGA4N	Compare match/input capture in MTU4.TGRA or underflow (trough) of MTU4.TCNT in complementary PWM mode	0	0	0	0	1	0	1
	TRGA6N	Compare match/input capture in MTU6.TGRA	0	0	0	0	1	1	0
	TRGA7N	Compare match/input capture in MTU7.TGRA or underflow (trough) of MTU7.TCNT in complementary PWM mode	0	0	0	0	1	1	1
	TRG0N	Compare match in MTU0.TGRE	0	0	0	1	0	0	0
	TRG4AN	Compare match between MTU4.TADCORA and MTU4.TCNT	0	0	0	1	0	0	1
	TRG4BN	Compare match between MTU4.TADCORB and MTU4.TCNT	0	0	0	1	0	1	0
	TRG4AN or TRG4BN	Compare match between MTU4.TADCORA and MTU4.TCNT, or compare match between MTU4.TADCORB and MTU4.TCNT	0	0	0	1	0	1	1
	TRG4ABN	Compare match between MTU4.TADCORA and MTU4.TCNT, and compare match between MTU4.TADCORB and MTU4.TCNT (when interrupt skipping function 2 is used)	0	0	0	1	1	0	0
	TRG7AN	Compare match between MTU7.TADCORA and MTU7.TCNT	0	0	0	1	1	0	1
	TRG7BN	Compare match between MTU7.TADCORB and MTU7.TCNT	0	0	0	1	1	1	0
	TRG7AN or TRG7BN	Compare match between MTU7.TADCORA and MTU7.TCNT, or compare match between MTU7.TADCORB and MTU7.TCNT	0	0	0	1	1	1	1
	TRG7ABN	Compare match between MTU7.TADCORA and MTU7.TCNT, and compare match between MTU7.TADCORB and MTU7.TCNT (when interrupt skipping function 2 is used)	0	0	1	0	0	0	0
	TRGA9N	Compare match/input capture in MTU9.TGRA	0	0	1	0	0	1	1
	TRG9N	Compare match in MTU9.TGRE	0	0	1	0	1	0	0
	TRGA0N or TRG0N	Compare match/input capture in MTU0.TGRA, or compare match in MTU0.TGRE	0	0	1	1	0	0	1
	TRGA9N or TRG9N	Compare match/input capture in MTU9.TGRA, or compare match in MTU9.TGRE	0	0	1	1	0	1	0
	TRGA0N or TRGA9N	Compare match/input capture in MTU0.TGRA, or compare match/input capture in MTU9.TGRA	0	0	1	1	0	1	1
	TRG0N or TRG9N	Compare match in MTU0.TGRE, or compare match in MTU9.TGRE	0	0	1	1	1	0	0
	TRG9AEN	Compare match/input capture in MTU9.TGRA, and compare match in MTU9.TGRE	0	1	0	0	0	0	1
	TRG0AEN	Compare match/input capture in MTU0.TGRA, and compare match in MTU0.TGRE	0	1	0	0	0	1	0

Table 42.14 Selection of A/D Trigger Sources by the TRSA[6:0] Bits (2/3)

Module	Source	Remarks	TRSA[6]	TRSA[5]	TRSA[4]	TRSA[3]	TRSA[2]	TRSA[1]	TRSA[0]
MTU	TRGA09N	Compare match/input capture in MTU0.TGRA, and compare match/input capture in MTU9.TGRA	0	1	0	0	0	1	1
	TRG09N	Compare match in MTU0.TGRE, and compare match in MTU9.TGRE	0	1	0	0	1	0	0
GPTW	GTADTRA0N	Compare match in GPTW0.GTADTRA	1	0	0	0	0	0	0
	GTADTRB0N	Compare match in GPTW0.GTADTRB	1	0	0	0	0	0	1
	GTADTRA1N	Compare match in GPTW1.GTADTRA	1	0	0	0	0	1	0
	GTADTRB1N	Compare match in GPTW1.GTADTRB	1	0	0	0	0	1	1
	GTADTRA2N	Compare match in GPTW2.GTADTRA	1	0	0	0	1	0	0
	GTADTRB2N	Compare match in GPTW2.GTADTRB	1	0	0	0	1	0	1
	GTADTRA3N	Compare match in GPTW3.GTADTRA	1	0	0	0	1	1	0
	GTADTRB3N	Compare match in GPTW3.GTADTRB	1	0	0	0	1	1	1
	GTADTRA0N or GTADTRB0N	Compare match in GPTW0.GTADTRA, or compare match in GPTW0.GTADTRB	1	0	0	1	0	0	0
	GTADTRA1N or GTADTRB1N	Compare match in GPTW1.GTADTRA, or compare match in GPTW1.GTADTRB	1	0	0	1	0	0	1
	GTADTRA2N or GTADTRB2N	Compare match in GPTW2.GTADTRA, or compare match in GPTW2.GTADTRB	1	0	0	1	0	1	0
	GTADTRA3N or GTADTRB3N	Compare match in GPTW3.GTADTRA, or compare match in GPTW3.GTADTRB	1	0	0	1	0	1	1
	GTADTRA4N	Compare match in GPTW4.GTADTRA	1	0	0	1	1	0	0
	GTADTRB4N	Compare match in GPTW4.GTADTRB	1	0	0	1	1	0	1
	GTADTRA5N	Compare match in GPTW5.GTADTRA	1	0	0	1	1	1	0
	GTADTRB5N	Compare match in GPTW5.GTADTRB	1	0	0	1	1	1	1
	GTADTRA6N	Compare match in GPTW6.GTADTRA	1	0	1	0	0	0	0
	GTADTRB6N	Compare match in GPTW6.GTADTRB	1	0	1	0	0	0	1
	GTADTRA7N	Compare match in GPTW7.GTADTRA	1	0	1	0	0	1	0
	GTADTRB7N	Compare match in GPTW7.GTADTRB	1	0	1	0	0	1	1
	GTADTRA4N or GTADTRB4N	Compare match in GPTW4.GTADTRA, or compare match in GPTW4.GTADTRB	1	0	1	0	1	0	0
	GTADTRA5N or GTADTRB5N	Compare match in GPTW5.GTADTRA, or compare match in GPTW5.GTADTRB	1	0	1	0	1	0	1
	GTADTRA6N or GTADTRB6N	Compare match in GPTW6.GTADTRA, or compare match in GPTW6.GTADTRB	1	0	1	0	1	1	0
	GTADTRA7N or GTADTRB7N	Compare match in GPTW7.GTADTRA, or compare match in GPTW7.GTADTRB	1	0	1	0	1	1	1
TMR	TMTRG0AN_0	Compare match between TMR0.TCORA and TMR0.TCNT	0	0	1	1	1	0	1
	TMTRG0AN_1	Compare match between TMR2.TCORA and TMR2.TCNT	0	0	1	1	1	1	0
	TMTRG0AN_2	Compare match between TMR4.TCORA and TMR4.TCNT	0	0	1	1	1	1	1
	TMTRG0AN_3	Compare match between TMR6.TCORA and TMR6.TCNT	0	1	0	0	0	0	0

Table 42.14 Selection of A/D Trigger Sources by the TRSA[6:0] Bits (3/3)

Module	Source	Remarks	TRSA[6]	TRSA[5]	TRSA[4]	TRSA[3]	TRSA[2]	TRSA[1]	TRSA[0]
ELC	ELCTRG00N *1 ELCTRG10N *2 ELCTRG20N *3	A/D Startup source 0 from ELC	0	1	1	0	0	1	0
	ELCTRG01N *1 ELCTRG11N *2 ELCTRG21N *3	A/D Startup source 1 from ELC	0	1	1	0	0	1	1
	ELCTRG00N or ELCTRG01N *1 ELCTRG10N or ELCTRG11N *2 ELCTRG20N or ELCTRG21N *3	A/D Startup source 0 from ELC, or A/D Startup source 1 from ELC	0	1	1	1	0	1	0

Note 1. Unit 0
Note 2. Unit 1
Note 3. Unit 2

42.2.16 A/D Conversion Extended Input Control Register (ADEXICR)

Address(es): S12AD2.ADEXICR 0008 9412h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	OCSB	TSSB	OCSA	TSSA	—	—	—	—	—	—	OCSAD	TSSAD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TSSAD	Temperature Sensor Output A/D-Converted Value Addition/Average Mode Select	0: Temperature sensor output A/D-converted value addition/average mode is disabled. 1: Temperature sensor output A/D-converted value addition/average mode is enabled.	R/W
b1	OCSAD	Internal Reference Voltage A/D-Converted Value Addition/Average Mode Select	0: Internal reference voltage A/D-converted value addition/average mode is disabled. 1: Internal reference voltage A/D-converted value addition/average mode is enabled.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	TSSA	Temperature Sensor Output A/D Conversion Select	0: Temperature sensor output is not A/D-converted 1: Temperature sensor output is A/D-converted	R/W
b9	OCSA	Internal Reference Voltage A/D Conversion Select	0: Internal reference voltage is not A/D-converted 1: Internal reference voltage is A/D-converted	R/W
b10	TSSB	Group B Temperature Sensor Output A/D Conversion Select	0: Temperature sensor output is not A/D-converted 1: Temperature sensor output is A/D-converted	R/W
b11	OCSB	Group B Internal Reference Voltage A/D Conversion Select	0: Internal reference voltage is not A/D-converted 1: Internal reference voltage is A/D-converted	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADEXICR register specifies the settings of A/D conversion of the temperature sensor output, internal reference voltage.

TSSAD Bit (Temperature Sensor Output A/D-Converted Value Addition/Average Mode Select)

When A/D conversion of the temperature sensor output is selected and the TSSAD bit is set to 1, A/D conversion of the temperature sensor output is performed in sequence for the count set in the ADADC.ADC[2:0] bits (2, 3, 4, or 16 times). The added (accumulated) total value is returned to the A/D temperature sensor data register (ADTSDR) when the ADADC.AVEE bit is 0 and the average is returned to the ADTSDR register when the ADADC.AVEE bit is 1. The TSSAD bit should be set while the ADCSR.ADST bit is 0.

OCSAD Bit (Internal Reference Voltage A/D-Converted Value Addition/Average Mode Select)

When A/D conversion of the internal reference voltage is selected and the OCSAD bit is set to 1, A/D conversion of the internal reference voltage is performed in sequence for the count set in the ADADC.ADC[2:0] bits (2, 3, 4, or 16 times). When the ADADC.AVEE bit is 0, the value obtained by addition (integration) is stored in the A/D internal reference voltage data register (ADOCDR). When the ADADC.AVEE bit is 1, the mean value is stored in the ADOCDR register. The OCSAD bit should be set while the ADCSR.ADST bit is 0.

TSSA Bit (Temperature Sensor Output A/D Conversion Select)

The TSSA bit selects A/D conversion of the temperature sensor output for group A in single scan mode or group scan mode. When A/D conversion of the temperature sensor output is performed, set the ADCSR.DBLE bit to 0.
The TSSA bit should be set while the ADCSR.ADST bit is 0. When setting the TSSB or ADGCEXCR.TSSC bit to 1, set this bit to 0.

The temperature sensor output can be selected in single scan mode and group scan mode. Set the sampling time to at least 4 µs when A/D-converting the temperature sensor output. When setting the TSSA bit to 1, the temperature sensor is automatically activated. After activation, only start A/D conversion after waiting 200 µs for the sensor output to become stable. When A/D conversion is started, auto-discharging for 15 cycles of ADCLK proceeds before sampling.
A/D conversion of the temperature sensor outputs in groups B and C should be handled in the same way.

OCSA Bit (Internal Reference Voltage A/D Conversion Select)

The OCSA bit selects A/D conversion of the internal reference voltage for group A in single scan mode or group scan mode. When A/D conversion of the internal reference voltage is performed, set the ADCSR.DBLE bit to 0.
The OCSA bit should be set while the ADCSR.ADST bit is 0. When setting the OCSB or ADGCEXCR.OCSC bit to 1, set this bit to 0.
To perform A/D conversion of the internal reference voltage, set the OCSA bit to 1, wait for at least 400 ns, and then start A/D conversion. When A/D conversion is started, auto-discharging for 15 cycles of ADCLK proceeds before sampling.
When the disconnection detection assist function is also in use, the setting of the ADNDIS[4:0] bits is ignored.
A/D conversion of the internal reference voltages in groups B and C should be handled in the same way.

TSSB Bit (Group B Temperature Sensor Output A/D Conversion Select)

The TSSB bit selects A/D conversion of the temperature sensor output for group B in group scan mode.
This bit should be set while the ADCSR.ADST bit is 0. When setting the TSSA or ADGCEXCR.TSSC bit to 1, set this bit to 0.
Refer to description of the TSSA bit for A/D-converting the temperature sensor output.

OCSB Bit (Group B Internal Reference Voltage A/D Conversion Select)

The OCSB bit selects A/D conversion of the internal reference voltage for group B in group scan mode.
This bit should be set while the ADCSR.ADST bit is 0. When setting the OCSA or ADGCEXCR.OCSC bit to 1, set this bit to 0.
Refer to description of the OCSA bit for A/D-converting the internal reference voltage.

42.2.17 A/D Group C Extended Input Control Register (ADGCEXCR)

Address(es): S12AD2.ADGCEXCR 0008 94D8h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	OCSC	TSSC
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TSSC	Group C Temperature Sensor Output A/D Conversion Select	0: Temperature sensor output is not A/D-converted 1: Temperature sensor output is A/D-converted	R/W
b1	OCSC	Group C Internal Reference Voltage A/D Conversion Select	0: Internal reference voltage is not A/D-converted 1: Internal reference voltage is A/D-converted	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADGCEXCR register specifies extended input for group C.

TSSC Bit (Group C Temperature Sensor Output A/D Conversion Select)

The TSSC bit selects A/D conversion of the temperature sensor output for group C in group scan mode.
The TSSC bit should be set while the ADCSR.ADST bit is 0. When setting the ADEXICR.TSSA or ADEXICR.TSSB bit to 1, set this bit to 0.
Refer to description of the ADEXICR.TSSA bit for A/D-converting the temperature sensor output.

OCSC Bit (Group C Internal Reference Voltage A/D Conversion Select)

The OCSC bit selects A/D conversion of the internal reference voltage for group C in group scan mode.
The OCSC bit should be set while the ADCSR.ADST bit is 0. When setting the ADEXICR.OCSA or ADEXICR.OCSB bit to 1, set this bit to 0.
Refer to description of the ADEXICR.OCSA bit for A/D-converting the internal reference voltage.

42.2.18 A/D Group C Trigger Select Register (ADGCTRGR)

Address(es): S12AD.ADGCTRGR 0008 90D9h, S12AD1.ADGCTRGR 0008 92D9h, S12AD2.ADGCTRGR 0008 94D9h

	b7	b6	b5	b4	b3	b2	b1	b0
	GRCE	GCADIE	TRSC[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	TRSC[5:0]	Group C A/D Conversion Start Trigger Select	In combination with the ADGCTRGR2.TRSC6 bit, these bits select the A/D conversion start trigger for group C in group scan mode.	R/W
b6	GCADIE	Group C Scan End Interrupt Enable	0: Group C scan end interrupt is disabled. 1: Group C scan end interrupt is enabled.	R/W
b7	GRCE	Group C A/D Conversion Operation Enable	Enables A/D conversion operation for group C. 0: Group C is not used 1: Group C is used	R/W

The ADGCTRGR register enables operation for group C and selects the A/D conversion start trigger. For details on group priority operation, refer to Table 42.21 and Table 42.22.

TRSC[5:0] Bits (Group C A/D Conversion Start Trigger Select)

In combination with the ADGCTRGR2.TRSC6 bit, the TRSC[5:0] bits select the trigger to start scanning of the analog input selected in group C. These bits are used for group scan mode only; not used for any other modes. Software trigger or asynchronous trigger cannot be set as the scan conversion trigger for group C. When using group C in group scan mode, set the TRSC6:TRSC[5:0] bits to a value other than 00h, set the ADCSR.TRGE bit to 1, and set the GRCE bit to 1. When group C is used during group priority control in group scan mode and the ADGSPCR.GBRP bit is set to 1, group C can be continuously operated in single scan mode. When continuously operating group C in single scan mode, set the TRSC[5:0] bits to 3Fh to disable trigger selection.

Note that the issuance period of trigger for A/D conversion must be more than or equal to the actual scan conversion time (t_{SCAN}). If the issuance period is less than t_{SCAN}, A/D conversion by a trigger may have no effect.

When selecting a trigger from the MTU operated on a PCLKC cycle as an A/D conversion start trigger, a delay of the period for synchronization processing occurs. Refer to section 42.3.7, Analog Input Sampling Time and Scan Conversion Time for details.

Table 42.15 lists the selection of A/D conversion start sources selected by the TRSC6 and TRSC[5:0] bits for group C.

Table 42.15 Selection of A/D Trigger Sources by the TRSC6 and TRSC[5:0] Bits (Group C) (1/2)

Module	Source	Remarks	TRSC6	TRSC[5]	TRSC[4]	TRSC[3]	TRSC[2]	TRSC[1]	TRSC[0]
Trigger source deselection state			0	1	1	1	1	1	1
			1	1	1	1	1	1	1
MTU	TRGA0N	Compare match/input capture in MTU0.TGRA	0	0	0	0	0	0	1
	TRGA1N	Compare match/input capture in MTU1.TGRA	0	0	0	0	0	1	0
	TRGA2N	Compare match/input capture in MTU2.TGRA	0	0	0	0	0	1	1
	TRGA3N	Compare match/input capture in MTU3.TGRA	0	0	0	0	1	0	0
	TRGA4N	Compare match/input capture in MTU4.TGRA or underflow (trough) of MTU4.TCNT in complementary PWM mode	0	0	0	0	1	0	1
	TRGA6N	Compare match/input capture in MTU6.TGRA	0	0	0	0	1	1	0
	TRGA7N	Compare match/input capture in MTU7.TGRA or underflow (trough) of MTU7.TCNT in complementary PWM mode	0	0	0	0	1	1	1
	TRG0N	Compare match in MTU0.TGRE	0	0	0	1	0	0	0
	TRG4AN	Compare match between MTU4.TADCORA and MTU4.TCNT	0	0	0	1	0	0	1
	TRG4BN	Compare match between MTU4.TADCORB and MTU4.TCNT	0	0	0	1	0	1	0
	TRG4AN or TRG4BN	Compare match between MTU4.TADCORA and MTU4.TCNT, or compare match between MTU4.TADCORB and MTU4.TCNT	0	0	0	1	0	1	1
	TRG4ABN	Compare match between MTU4.TADCORA and MTU4.TCNT, and compare match between MTU4.TADCORB and MTU4.TCNT (when interrupt skipping function 2 is used)	0	0	0	1	1	0	0
	TRG7AN	Compare match between MTU7.TADCORA and MTU7.TCNT	0	0	0	1	1	0	1
	TRG7BN	Compare match between MTU7.TADCORB and MTU7.TCNT	0	0	0	1	1	1	0
	TRG7AN or TRG7BN	Compare match between MTU7.TADCORA and MTU7.TCNT, or compare match between MTU7.TADCORB and MTU7.TCNT	0	0	0	1	1	1	1
	TRG7ABN	Compare match between MTU7.TADCORA and MTU7.TCNT, and compare match between MTU7.TADCORB and MTU7.TCNT (when interrupt skipping function 2 is used)	0	0	1	0	0	0	0
	TRGA9N	Compare match/input capture in MTU9.TGRA	0	0	1	0	0	1	1
	TRG9N	Compare match in MTU9.TGRE	0	0	1	0	1	0	0
	TRGA0N or TRG0N	Compare match/input capture in MTU0.TGRA, or compare match in MTU0.TGRE	0	0	1	1	0	0	1
	TRGA9N or TRG9N	Compare match/input capture in MTU9.TGRA, or compare match in MTU9.TGRE	0	0	1	1	0	1	0
	TRGA0N or TRGA9N	Compare match/input capture in MTU0.TGRA, or compare match/input capture in MTU9.TGRA	0	0	1	1	0	1	1
	TRG0N or TRG9N	Compare match in MTU0.TGRE, or compare match in MTU9.TGRE	0	0	1	1	1	0	0
	TRG9AEN	Compare match/input capture in MTU9.TGRA, and compare match in MTU9.TGRE	0	1	0	0	0	0	1
	TRG0AEN	Compare match/input capture in MTU0.TGRA, and compare match in MTU0.TGRE	0	1	0	0	0	1	0
	TRGA09N	Compare match/input capture in MTU0.TGRA, and compare match/input capture in MTU9.TGRA	0	1	0	0	0	1	1
	TRG09N	Compare match in MTU0.TGRE, and compare match in MTU9.TGRE	0	1	0	0	1	0	0

Table 42.15 Selection of A/D Trigger Sources by the TRSC6 and TRSC[5:0] Bits (Group C) (2/2)

Module	Source	Remarks	TRSC6	TRSC[5]	TRSC[4]	TRSC[3]	TRSC[2]	TRSC[1]	TRSC[0]
GPTW	GTADTRA0N	Compare match in GPTW0.GTADTRA	1	0	0	0	0	0	0
	GTADTRB0N	Compare match in GPTW0.GTADTRB	1	0	0	0	0	0	1
	GTADTRA1N	Compare match in GPTW1.GTADTRA	1	0	0	0	0	1	0
	GTADTRB1N	Compare match in GPTW1.GTADTRB	1	0	0	0	0	1	1
	GTADTRA2N	Compare match in GPTW2.GTADTRA	1	0	0	0	1	0	0
	GTADTRB2N	Compare match in GPTW2.GTADTRB	1	0	0	0	1	0	1
	GTADTRA3N	Compare match in GPTW3.GTADTRA	1	0	0	0	1	1	0
	GTADTRB3N	Compare match in GPTW3.GTADTRB	1	0	0	0	1	1	1
	GTADTRA0N or GTADTRB0N	Compare match in GPTW0.GTADTRA, or compare match in GPTW0.GTADTRB	1	0	0	1	0	0	0
	GTADTRA1N or GTADTRB1N	Compare match in GPTW1.GTADTRA, or compare match in GPTW1.GTADTRB	1	0	0	1	0	0	1
	GTADTRA2N or GTADTRB2N	Compare match in GPTW2.GTADTRA, or compare match in GPTW2.GTADTRB	1	0	0	1	0	1	0
	GTADTRA3N or GTADTRB3N	Compare match in GPTW3.GTADTRA, or compare match in GPTW3.GTADTRB	1	0	0	1	0	1	1
	GTADTRA4N	Compare match in GPTW4.GTADTRA	1	0	0	1	1	0	0
	GTADTRB4N	Compare match in GPTW4.GTADTRB	1	0	0	1	1	0	1
	GTADTRA5N	Compare match in GPTW5.GTADTRA	1	0	0	1	1	1	0
	GTADTRB5N	Compare match in GPTW5.GTADTRB	1	0	0	1	1	1	1
	GTADTRA6N	Compare match in GPTW6.GTADTRA	1	0	1	0	0	0	0
	GTADTRB6N	Compare match in GPTW6.GTADTRB	1	0	1	0	0	0	1
	GTADTRA7N	Compare match in GPTW7.GTADTRA	1	0	1	0	0	1	0
	GTADTRB7N	Compare match in GPTW7.GTADTRB	1	0	1	0	0	1	1
	GTADTRA4N or GTADTRB4N	Compare match in GPTW4.GTADTRA, or compare match in GPTW4.GTADTRB	1	0	1	0	1	0	0
	GTADTRA5N or GTADTRB5N	Compare match in GPTW5.GTADTRA, or compare match in GPTW5.GTADTRB	1	0	1	0	1	0	1
	GTADTRA6N or GTADTRB6N	Compare match in GPTW6.GTADTRA, or compare match in GPTW6.GTADTRB	1	0	1	0	1	1	0
	GTADTRA7N or GTADTRB7N	Compare match in GPTW7.GTADTRA, or compare match in GPTW7.GTADTRB	1	0	1	0	1	1	1
TMR	TMTRG0AN_0	Compare match between TMR0.TCORA and TMR0.TCNT	0	0	1	1	1	0	1
	TMTRG0AN_1	Compare match between TMR2.TCORA and TMR2.TCNT	0	0	1	1	1	1	0
	TMTRG0AN_2	Compare match between TMR4.TCORA and TMR4.TCNT	0	0	1	1	1	1	1
	TMTRG0AN_3	Compare match between TMR6.TCORA and TMR6.TCNT	0	1	0	0	0	0	0
ELC	ELCTRG00N *1 ELCTRG10N *2 ELCTRG20N *3	A/D Startup source 0 from ELC	0	1	1	0	0	1	0
	ELCTRG01N *1 ELCTRG11N *2 ELCTRG21N *3	A/D Startup source 1 from ELC	0	1	1	0	0	1	1
	ELCTRG00N or ELCTRG01N *1 ELCTRG10N or ELCTRG11N *2 ELCTRG20N or ELCTRG21N *3	A/D Startup source 0 from ELC, or A/D Startup source 1 from ELC	0	1	1	1	0	1	0

Note 1. Unit 0
Note 2. Unit 1
Note 3. Unit 2

GCADIE Bit (Group C Scan End Interrupt Enable)

The GCADIE bit enables or disables scan end interrupt generation for group C. A scan end interrupt for group C is provided individually for each unit. Table 42.16 shows the relationship between each unit and the scan end interrupt for group C.

Table 42.16 Relationship Between Each Unit and Group C Scan End Interrupt

Unit	Group C Scan End Interrupt
S12AD	S12GCADI
S12AD1	S12GCADI1
S12AD2	S12GCADI2

GRCE Bit (Group C A/D Conversion Operation Enable)

When using group C in group scan mode, set the GRCE bit to 1.
When the GRCE bit is 0, trigger input for group C is disabled.
During group priority operation (the ADGSPCR.PGS bit is 1) with group C used, when the ADGSPCR.GBRP bit is set to 1, single scan for group C is continuously operated. When the GRCE bit is set to 1, single scan for group B is not continuously operated.
The GRCE bit should be set while the ADCSR.ADST bit is 0.

42.2.19 A/D Group C Trigger Select Register 2 (ADGCTRGR2)

Address(es): S12AD.ADGCTRGR2 0008 90DCh, S12AD1.ADGCTRGR2 0008 92DCh, S12AD2.ADGCTRGR2 0008 94DCh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	TRSC6
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TRSC6	Group C A/D Conversion Start Trigger Select	In combination with the ADGCTRGR.TRSC[5:0] bits, this bit selects the A/D conversion start trigger for group C in group scan mode.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADGCTRGR2 register is used in combination with the ADGCTRGR register to select the A/D conversion start trigger for group C. For details, refer to section 42.2.18, A/D Group C Trigger Select Register (ADGCTRGR).

42.2.20 A/D Sampling State Register n (ADSSTRn) (n = 0 to 11, L, T, O)

Address(es): S12AD.ADSSTR0 0008 90E0h, S12AD.ADSSTR1 0008 90E1h, S12AD.ADSSTR2 0008 90E2h, S12AD.ADSSTR3 0008 90E3h, S12AD.ADSSTR4 0008 90E4h, S12AD.ADSSTR5 0008 90E5h, S12AD.ADSSTR6 0008 90E6h, S12AD1.ADSSTR0 0008 92E0h, S12AD1.ADSSTR1 0008 92E1h, S12AD1.ADSSTR2 0008 92E2h, S12AD1.ADSSTR3 0008 92E3h, S12AD2.ADSSTR0 0008 94E0h, S12AD2.ADSSTR1 0008 94E1h, S12AD2.ADSSTR2 0008 94E2h, S12AD2.ADSSTR3 0008 94E3h, S12AD2.ADSSTR4 0008 94E4h, S12AD2.ADSSTR5 0008 94E5h, S12AD2.ADSSTR6 0008 94E6h, S12AD2.ADSSTR7 0008 94E7h, S12AD2.ADSSTR8 0008 94E8h, S12AD2.ADSSTR9 0008 94E9h, S12AD2.ADSSTR10 0008 94EAh, S12AD2.ADSSTR11 0008 94EBh, S12AD2.ADSSTRL 0008 94DDh, S12AD2.ADSSTRT 0008 94DEh, S12AD2.ADSSTRO 0008 94DFh

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	1	1	0	1	1

The ADSSTRn register sets the sampling time for analog input.
Set the sampling time based on the number of clock cycles for ADCLK (A/D conversion clock). The initial value is 27 clock cycles. If the impedance of analog input signal source is too high to secure sufficient sampling time or if the ADCLK is slow, the sampling time can be adjusted. The ADSSTRn register should be set while the ADCSR.ADST bit is 0.
Specify the value for the register as a multiples of 3 in the range from 12 to 252 (clock cycles).
A sampling time can be determined by the following formula:

$$\text{Sampling time} = \text{ADSSTR} \times t_C(\text{ADCLK})$$

Table 42.17 shows the relationship between the A/D sampling state register and the relevant channels.
For details, refer to section 42.3.7, Analog Input Sampling Time and Scan Conversion Time.

Table 42.17 Relationship Between A/D Sampling State Register and Relevant Channels

Unit	Register Name	Relevant Channels
S12AD	ADSSTR0 register	AN000, Self-Diagnosis
	ADSSTR1 register	AN001
	ADSSTR2 register	AN002
	ADSSTR3 register	AN003
	ADSSTR4 register	AN004
	ADSSTR5 register	AN005
	ADSSTR6 register	AN006
S12AD1	ADSSTR0 register	AN100, Self-Diagnosis
	ADSSTR1 register	AN101
	ADSSTR2 register	AN102
	ADSSTR3 register	AN103
S12AD2	ADSSTR0 register	AN200, Self-Diagnosis
	ADSSTR1 register	AN201
	ADSSTR2 register	AN202
	ADSSTR3 register	AN203
	ADSSTR4 register	AN204
	ADSSTR5 register	AN205
	ADSSTR6 register	AN206
	ADSSTR7 register	AN207
	ADSSTR8 register	AN208
	ADSSTR9 register	AN209
	ADSSTR10 register	AN210
	ADSSTR11 register	AN211
	ADSSTRL register	AN216, AN217
	ADSSTRT register	Temperature sensor output*1
	ADSSTRO register	Internal reference voltage*2

Note 1. When performing A/D conversion of the temperature sensor output, it is necessary to set the sampling time that meets the specification described in section 49, Electrical Characteristics.
Note 2. When performing A/D conversion of the internal reference voltage, the sampling time should be 4 µs or longer.

42.2.21 A/D Sample-and-Hold Circuit Control Register (ADSHCR)

Address(es): S12AD.ADSHCR 0008 9066h, S12AD1.ADSHCR 0008 9266h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SHANS[2:0]			SSTSH[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	SSTSH[7:0]	Channel-Dedicated Sample-and-Hold Circuit Sampling Time Setting	Set a sampling time between 12 and 252 clock cycles.	R/W
b8	SHANS[0]	Channel-Dedicated Sample-and-Hold Circuit Bypass Select	Select whether to use or not use (bypass) the channel-dedicated sample-and-hold circuits either for AN000 or AN100. 0: Channel-dedicated sample-and-hold circuit is disabled. 1: Channel-dedicated sample-and-hold circuit is enabled.	R/W
b9	SHANS[1]		Select whether to use or not use (bypass) the channel-dedicated sample-and-hold circuits either for AN001 or AN101. 0: Channel-dedicated sample-and-hold circuit is disabled. 1: Channel-dedicated sample-and-hold circuit is enabled.	R/W
b10	SHANS[2]		Select whether to use or not use (bypass) the channel-dedicated sample-and-hold circuits either for AN002 or AN102. 0: Channel-dedicated sample-and-hold circuit is disabled. 1: Channel-dedicated sample-and-hold circuit is enabled.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADSHCR register sets the parameters related to channel-dedicated sample-and-hold circuits.

SSTSH[7:0] Bits (Channel-Dedicated Sample-and-Hold Circuit Sampling Time Setting)

The SSTSH[7:0] bits set the sampling time for the channel-dedicated sample-and-hold circuits when the ADSHMSR.SHMD bit is 0. Set the sampling time in clock cycles of ADCLK (A/D conversion clock). The initial value is 27 clock cycles. If the impedance of analog input signal source is too high to secure sufficient sampling time or if ADCLK is slow, the sampling time can be adjusted. Set the SSTSH[7:0] bits while the ADCSR.ADST bit is 0. Set a value between 12 and 252 clock cycles as the setting for sampling time.
Also, set a value such that the sampling time will be no less than 0.4 µs.

SHANS[2:0] Bits (Channel-Dedicated Sample-and-Hold Circuit Bypass Select)

The SHANS[2:0] bits select whether to use or bypass the channel-dedicated sample-and-hold circuits of AN000 to AN002 of unit 0 or AN100 to AN102 of unit 1. Set the SHANS[2:0] bits while both the ADCSR.ADST and ADSHMSR.SHMD bits are 0. If any of channels AN000 to AN002 of unit 0 or AN100 to AN102 of unit 1 is selected for group B or group C while operation is under group A priority control in group scan mode, the channel-dedicated sample-and-hold circuit should be disabled.
No channels of unit 2 have channel-dedicated sample-and-hold circuits.

42.2.22 A/D Sample-and-Hold Operating Mode Select Register (ADSHMSR)

Address(es): S12AD.ADSHMSR 0008 907Ch, S12AD1.ADSHMSR 0008 927Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SHMD
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SHMD	Channel-Dedicated Sample-and-Hold Circuit Operating Mode Setting	0: Constant sampling of the channel-dedicated sample-and-hold circuits is disabled. 1: Constant sampling of the channel-dedicated sample-and-hold circuits is enabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADSHMSR register enables or disables constant sampling of the sample-and-hold circuit dedicated for channels.

SHMD Bit (Channel-Dedicated Sample-and-Hold Circuit Operating Mode Setting)

When the SHMD bit is set to 1, constant sampling of the sample-and-hold circuit for the channels selected by the ADSHCR.SHANS[2:0] bits is enabled. The SHMD bit should be set while the ADCSR.ADST bit is 0.

When the constant sampling function is enabled, constant sampling is performed while the 12-bit A/D converter is in the wait state and holding is performed when the A/D conversion is in progress.

Enable use of the channel-dedicated sample-and-hold circuit (ADSHCR.SHANS[2:0] ≠ 000b), enable normal sampling by the channel-dedicated sample-and-hold circuit (ADSHMSR.SHMD = 1), and wait for 6 cycles of ADCLK to pass before input of a trigger signal.

42.2.23 A/D Disconnection Detection Control Register (ADDISCR)

Address(es): S12AD.ADDISCR 0008 907Ah, S12AD1.ADDISCR 0008 927Ah, S12AD2.ADDISCR 0008 947Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	ADNDIS[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	ADNDIS[3:0]	A/D Disconnection Detection Assist Setting	Specify the period for discharging or precharging as a number of ADCLK cycles. b3 b0 0 0 0 0: No charging (disconnection detection assist function is disabled.) 0 0 1 1: Charging period of 3 clock cycles 0 1 1 0: Charging period of 6 clock cycles 1 0 0 1: Charging period of 9 clock cycles 1 1 0 0: Charging period of 12 clock cycles 1 1 1 1: Charging period of 15 clock cycles Settings other than above are prohibited.	R/W
b4	ADNDIS[4]		0: Discharge 1: Precharge	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADDISCR register sets the disconnection detection assist function.

ADNDIS[4:0] Bits (A/D Disconnection Detection Assist Setting)

The ADNDIS[4:0] bits select either precharge or discharge and the period of precharge/discharge for the A/D disconnection detection assist function. Setting the ADNDIS[4] bit = 1 allows to select precharge and setting the ADNDIS[4] bit = 0 allows to select discharge. The period of precharge/discharge can be set with the ADNDIS[3:0] bits. When ADNDIS[3:0] bits = 0000b, the disconnection detection assist function is disabled. Set the ADNDIS[3:0] bits to 0000b or a multiple of 3. A value other than those above cannot be set. The value set in the ADNDIS[3:0] bits is used to set the number of clock cycles that determines the precharging or discharging period.

The ADNDIS[4:0] bits should be set when the ADCSR.ADST bit is 0.

The disconnection detection assist function cannot be used when the PGA output, temperature sensor output, or internal reference voltage is converted, or when the self-diagnosis function is used. Set the ADNDIS[3:0] bits to 0000b.

When A/D conversion of a temperature sensor output or internal reference voltage proceeds, auto-discharging is also executed beforehand. Therefore, the ADNDIS[4:0] bits are automatically fixed to 0Fh during conversion for a scan group that includes a temperature sensor output or internal reference voltage, and the corresponding analog input paths in the A/D converter are discharged. Sampling starts after discharging is completed.

After the A/D conversion of a temperature sensor output or internal reference voltage is finished, the ADNDIS[4:0] bits automatically return to the setting before the conversion.

42.2.24 A/D Event Link Control Register (ADELCCR)

Address(es): S12AD.ADELCCR 0008 907Dh, S12AD1.ADELCCR 0008 927Dh, S12AD2.ADELCCR 0008 947Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	ELCC[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	ELCC[2:0]	Event Link Control	b2 b0 0 0 0: An event signal is generated when scanning for group A is completed. 0 0 1: An event signal is generated when scanning for group B is completed. 0 1 0: An event signal is generated when scanning for group A, group B, or group C is completed. 1 0 0: An event signal is generated when scanning for group C is completed. Settings other than above are prohibited.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADELCCR register is used to set an event signal generation condition at completion of scanning.

ELCC[2:0] Bits (Event Link Control)

The ELCC[2:0] bits are used to select an event signal generation condition at completion of scanning.

42.2.25 A/D Group Scan Priority Control Register (ADGSPCR)

Address(es): S12AD.ADGSPCR 0008 9080h, S12AD1.ADGSPCR 0008 9280h, S12AD2.ADGSPCR 0008 9480h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	GBRP	LGRRS	—	—	—	—	—	—	—	—	—	—	—	—	GBRSCN	PGS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PGS	Group Priority Control Setting *1	0: Operation is without group priority control 1: Operation is with group priority control	R/W
b1	GBRSCN	Low-Priority Group Restart Setting	(Enabled only when PGS = 1. Reserved when PGS = 0.) 0: Scanning for the group is not restarted after having been interrupted due to group priority control. 1: Scanning for the group is restarted after having been interrupted due to group priority control.	R/W
b13 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	LGRRS	Restart Channel Select	(Enabled when PGS = 1 and GBRSCN = 1. Reserved when PGS = 0 or GBRSCN = 0.) 0: Scanning is restarted from the scan start channel. 1: Scanning is restarted from the channel on which A/D conversion is not completed.	R/W
b15	GBRP	Single Scan Continuous Start *2	(Enabled only when PGS = 1. Reserved when PGS = 0.) 0: Single scan is not continuously started. 1: Single scan for the lowest-priority group is continuously started.	R/W

Note 1. When the PGS bit is to be set to 1, the ADCSR.ADCS[1:0] bits must be set to 01b (group scan mode). If the bits are set to any other values, proper operation is not guaranteed.
Note 2. When the GBRP bit has been set to 1, single scan is performed continuously for the lowest-priority group regardless of the setting of the GBRSCN bit.

The ADGSPCR register is used to interrupted scanning of the low-priority group and make settings for priority control of scanning for the priority group in group scan mode.
For the settings on group priority operation, refer to Table 42.21 and Table 42.22.

PGS Bit (Group Priority Control Setting)

The PGS bit sets the priority of operation in group scan mode. Set this bit to 1 when giving priority to operation on the group.
When the PGS bit is to be set to 1, the ADCSR.ADCS[1:0] bits must be set to 01b (group scan mode).
During group priority operation, a trigger to start scanning for the priority group is accepted during scan for the low-priority group, and scan for the priority group is started after scan for the low-priority group was interrupted. The priority order is group A > group B > group C.
When a trigger to start scanning for group B is accepted during scan for group C, group C scan is interrupted, and scan for group B is started. When a trigger to start scanning for group A is accepted during scan for group C, group C scan is interrupted, and scan for group A is started.
Likewise, when a trigger to start scanning for group A is accepted during scan for group B, group B scan is interrupted and scan for group A is started.
When setting the PGS bit to 0, clearing should be performed by software according to section 42.6.2, Notes on Stopping A/D Conversion. When setting the PGS bit to 1, follow the procedure described in section 42.3.5.3, Operation under Group Priority Control.

GBRSCN Bit (Low-Priority Group Restart Setting)

The GBRSCN bit controls the restarting of scan operation during group priority control.
If a scan operation on the low-priority group has been stopped by a priority group trigger input with the GBRSCN bit set to 1, the scan operation is restarted after the scanning of the priority group is completed. Also, if a low-priority trigger is input during scan for the priority group, the scan operation on the low-priority group is restarted after the scan for the priority group is completed.
If the GBRSCN bit has been set to 0, triggers that are input during A/D conversion are ignored. Also, the ADCSR.ADST bit must be 0 when the GBRSCN bit is to be set.
The setting of the GBRSCN bit is enabled when the PGS bit is set to 1.

LGRRS Bit (Restart Channel Select)

The LGRRS bit sets the channel on which scan is restarted during group priority control. The setting of the LGRRS bit is enabled when the PGS and GBRSCN bits are set to 1.
If a scan operation on the low-priority group has been stopped due to group priority operation with the LGRRS bit set to 0, the scan operation is restarted from the start channel after the scan for the priority group is completed.
If a scan operation on the low-priority group has been stopped due to group priority operation with the LGRRS bit set to 1, the scan operation is restarted*1 on the channel on which A/D conversion is not competed after the scan for the priority group is completed.
The LGRRS bit should be set while the ADCSR.ADST bit is 0.

Note 1. If A/D conversion on the addition set channel is not completed for the set number of times when scanning is stopped, A/D conversion on the channel is restarted for the set number of times when scanning is restarted.

GBRP Bit (Single Scan Continuous Start)

The GBRP bit is set when the lowest-priority group is continuously operated in single scan mode while group priority operation is set. The lowest-priority group is group C when groups A, B, and C are used; group B when groups A and B are used.
Setting the GBRP bit to 1 starts a single scan on the lowest-priority group. On completion of the scan, another single scan on the lowest-priority group is automatically started.
If scanning has been stopped due to group priority operation, single scan on the lowest-priority group is automatically restarted on completion of the A/D conversion on the priority group.
Disable the trigger input for the lowest-priority group before setting the GBRP bit to 1. When the GBRP bit is set to 1, only the lowest-priority group is scanned again even if the GBRSCN bit is 0.
The ADCSR.ADST bit must be 0 when the GBRP bit is to be set.
The setting of the GBRP bit is enabled when the PGS bit is 1.

42.2.26 A/D Comparison Function Control Register (ADCMPCR)

Address(es): S12AD.ADCMPCR 0008 9090h, S12AD1.ADCMPCR 0008 9290h, S12AD2.ADCMPCR 0008 9490h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CMPAIE	WCMPE	CMPBIE	—	CMPAE	—	CMPBE	—	—	—	—	—	—	—	CMPAB[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CMPAB[1:0]	Window A/B Complex Conditions Setting	b1 b0 0 0: Window A comparison condition matched OR window B comparison condition matched 0 1: Window A comparison condition matched XOR window B comparison condition matched 1 0: Window A comparison condition matched AND window B comparison condition matched 1 1: Setting is prohibited	R/W
b8 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9	CMPBE	Comparison Window B Enable	0: Comparison window B disabled 1: Comparison window B enabled	R/W
b10	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b11	CMPAE	Comparison Window A Enable	0: Comparison window A disabled 1: Comparison window A enabled	R/W
b12	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b13	CMPBIE	Comparison B Interrupt Enable	0: Comparison interrupt by a match with the comparison condition (window B) is disabled 1: Comparison interrupt by a match with the comparison condition (window B) is enabled	R/W
b14	WCMPE	Window Function Setting	0: The window function is disabled Window A or B operates as a comparator for comparing between one value on the low side and the A/D-converted value 1: The window function is enabled Window A or B operates as a window comparator for comparing between two values on the high and low sides and the A/D-converted value	R/W
b15	CMPAIE	Comparison A Interrupt Enable	0: Comparison interrupt by a match with the comparison condition (window A) is disabled 1: Comparison interrupt by a match with the comparison condition (window A) is enabled	R/W

The ADCMPCR register is used to set the settings for the comparison window function (windows A and B).

CMPAB[1:0] Bits (Window A/B Complex Conditions Setting)

The CMPAB[1:0] bits are enabled when single scan mode and window A/B are both enabled (CMPAE bit = 1 and CMPBE bit = 1). These bits specify the monitor conditions of the ADWINMON.MONCOMB flag. The CMPAB[1:0] bits should be set when the ADCSR.ADST bit is 0.

CMPBE Bit (Comparison Window B Enable)

The CMPBE bit is used to disable or enable comparison window B. The CMPBE bit should be set when the ADCSR.ADST bit is 0.

To set the following registers, set this bit to 0.

- A/D channel select registers A0/A1/B0/B1/C0/C1 (ADANSA0, ADANSA1, ADANSB0, ADANSB1, ADANSC0, and ADANSC1)
- Bits OCSB, TSSB, OCSA, and TSSA in the A/D conversion extended input control register (ADEXICR.OCSB,

TSSB, OCSA, and TSSA)

- Bits OCSC and TSSC in the A/D group C extended input control register (ADGCEXCR.OCSC and TSSC)
- The CMPCHB[5:0] bits in the window B channel select register (ADCMPBNSR.CMPCHB[5:0])

CMPAE Bit (Comparison Window A Enable)

The CMPAE bit is used to disable or enable comparison window A. The CMPAE bit should be set when the ADCSR.ADST bit is 0.

To set the following registers, set this bit to 0.

- A/D channel select registers A0/A1/B0/B1/C0/C1 (ADANSA0, ADANSA1, ADANSB0, ADANSB1, ADANSC0, and ADANSC1)
- Bits OCSB, TSSB, OCSA, and TSSA in the A/D conversion extended input control register (ADEXICR.OCSB, TSSB, OCSA, and TSSA)
- Bits OCSC and TSSC in the A/D group C extended input control register (ADGCEXCR.OCSC and TSSC)
- Window A channel select registers 0 and 1 (ADCMPANSR0 and ADCMPANSR1)
- Window A extended input select register (ADCMPANSER)

CMPBIE Bit (Comparison B Interrupt Enable)

Enables or disables comparison interrupts by a window B comparison condition match. A single comparison interrupt exists for each unit. Table 42.18 lists the relationship between each unit and its comparison interrupt.

WCMPE Bit (Window Function Setting)

The WCMPE bit is used to disable or enable the window function. The WCMPE bit should be set when the ADCSR.ADST bit is 0.

CMPAIE Bit (Comparison A Interrupt Enable)

Enables or disables comparison interrupts by a window A comparison condition match. A single comparison interrupt exists for each unit. Table 42.18 lists the relationship between each unit and its comparison interrupt.

Table 42.18 Relationship Between Each Unit and Its Comparison Interrupt Unit

Unit	Comparison Interrupt	
	When Window A Comparison Condition is Met	When Window B Comparison Condition is Met
S12AD	S12CMPAI	S12CMPBI
S12AD1	S12CMPAI1	S12CMPBI1
S12AD2	S12CMPAI2	S12CMPBI2

42.2.27 A/D Comparison Function Window A Channel Select Register 0 (ADCMPANSR0)

(1) S12AD.ADCMPANSR0

Address(es): S12AD.ADCMPANSR0 0008 9094h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	CMPCHA006	CMPCHA005	CMPCHA004	CMPCHA003	CMPCHA002	CMPCHA001	CMPCHA000
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMPCHA000	Comparison Window A Channel Select	These bits determine whether to include pins AN000 to AN006 as the targets for comparison window A. 0: Exclude from the targets for comparison window A. 1: Include as the targets for comparison window A.	R/W
b1	CMPCHA001			R/W
b2	CMPCHA002			R/W
b3	CMPCHA003			R/W
b4	CMPCHA004			R/W
b5	CMPCHA005			R/W
b6	CMPCHA006			R/W
b15 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The S12AD.ADCMPANSR0 register selects analog inputs AN000 to AN006 of the channels that perform comparison with the conditions of comparison window A.

CMPCHA0n Bit (Comparison Window A Channel Select) (n = 00 to 06)

The comparison function is enabled when the CMPCHA0n bit with the same index number as the A/D conversion channel selected by the ADANSA0.ANSA0n bit, the ADANSB0.ANSB0n bit, and the ADANSC0.ANSC0n bit is set to 1.

The CMPCHA0n bit should be set while the ADCSR.ADST bit is 0.

(2) S12AD1.ADCMPANSR0

Address(es): S12AD1.ADCMPANSR0 0008 9294h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	CMPCHA003	CMPCHA002	CMPCHA001	CMPCHA000
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMPCHA000	Comparison Window A Channel Select	These bits determine whether to include pins AN100 to AN103 as the targets for comparison window A. 0: Exclude from the targets for comparison window A. 1: Include as the targets for comparison window A.	R/W
b1	CMPCHA001			R/W
b2	CMPCHA002			R/W
b3	CMPCHA003			R/W
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The S12AD1.ADCMPANSR0 register selects analog inputs AN100 to AN103 of the channels that perform comparison with the conditions of comparison window A.

CMPCHA0n Bit (Comparison Window A Channel Select) (n = 00 to 03)

The comparison function is enabled when the CMPCHA0n bit with the same index number as the A/D conversion channel selected by the ADANSA0.ANSA0n bit, the ADANSB0.ANSB0n bit, and the ADANSC0.ANSC0n bit is set to 1.

The CMPCHA0n bit should be set while the ADCSR.ADST bit is 0.

(3) S12AD2.ADCMPANSR0

Address(es): S12AD2.ADCMPANSR0 0008 9494h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	CMPCHA011	CMPCHA010	CMPCHA009	CMPCHA008	CMPCHA007	CMPCHA006	CMPCHA005	CMPCHA004	CMPCHA003	CMPCHA002	CMPCHA001	CMPCHA000
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMPCHA000	Comparison Window A Channel Select	These bits determine whether to include pins AN200 to AN211 as the targets for comparison window A. 0: Exclude from the targets for comparison window A. 1: Include as the targets for comparison window A.	R/W
b1	CMPCHA001			R/W
b2	CMPCHA002			R/W
b3	CMPCHA003			R/W
b4	CMPCHA004			R/W
b5	CMPCHA005			R/W
b6	CMPCHA006			R/W
b7	CMPCHA007			R/W
b8	CMPCHA008			R/W
b9	CMPCHA009			R/W
b10	CMPCHA010			R/W
b11	CMPCHA011			R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The S12AD2.ADCMPANSR0 register selects analog inputs AN200 to AN211 of the channels that perform comparison with the conditions of comparison window A.

CMPCHA0n Bit (Comparison Window A Channel Select) (n = 00 to 11)

The comparison function is enabled when the CMPCHA0n bit with the same index number as the A/D conversion channel selected by the ADANSA0.ANSA0n bit, the ADANSB0.ANSB0n bit, and the ADANSC0.ANSC0n bit is set to 1.

The CMPCHA0n bit should be set while the ADCSR.ADST bit is 0.

42.2.28 A/D Comparison Function Window A Channel Select Register 1 (ADCMPANSR1)

Address(es): S12AD2.ADCMPANSR1 0008 9496h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CMPCHA101	CMPCHA100
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMPCHA100	Comparison Window A Channel Select	These bits determine whether to include pins AN216 to AN217 as the targets for comparison window A. 0: Exclude from the targets for comparison window A. 1: Include as the targets for comparison window A.	R/W
b1	CMPCHA101			R/W
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADCMPANSR1 register selects analog inputs AN216 to AN217 of the channels that perform comparison with the conditions of comparison window A.

CMPCHA1n Bit (Comparison Window A Channel Select) (n = 00, 01)

The comparison function is enabled when the CMPCHA1n bit with the same index number as the A/D conversion channel selected by the ADANSA1.ANSA1n bit, the ADANSB1.ANSB1n bit, and the ADANSC1.ANSC1n bit is set to 1.

The CMPCHA1n bit should be set while the ADCSR.ADST bit is 0.

42.2.29 A/D Comparison Function Window A Extended Input Select Register (ADCMPANSER)

Address(es): S12AD2.ADCMPANSER 0008 9492h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	CMPSOC	CMPSTS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMPSTS	Temperature Sensor Output Comparison Select	0: Temperature sensor output is excluded from the targets for comparison window A 1: Temperature sensor output is included as the targets for comparison window A	R/W
b1	CMPSOC	Internal Reference Voltage Compare Select	0: Internal reference voltage is excluded from the targets for comparison window A 1: Internal reference voltage is included as the targets for comparison window A	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADCMPANSER register selects whether the temperature sensor output or internal reference voltage is compared under the conditions of comparison window A.

CMPSTS Bit (Temperature Sensor Output Comparison Select)

The comparison window A is enabled when the CMPSTS bit is set to 1 while the ADEXICR.TSSA, ADEXICR.TSSB, or the ADGCEXCR.TSSC bit is 1. The CMPSTS bit should be set while the ADCSR.ADST bit is 0.

CMPSOC Bit (Internal Reference Voltage Compare Select)

The comparison window A is enabled when the CMPSOC bit is set to 1 while the ADEXICR.OCSA, ADEXICR.OCSB, or ADGCEXCR.OCSC bit is 1. The CMPSOC bit should be set while the ADCSR.ADST bit is 0.

42.2.30 A/D Comparison Function Window A Comparison Condition Setting Register 0 (ADCMPLR0)

(1) S12AD.ADCMPLR0

Address(es): S12AD.ADCMPLR0 0008 9098h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	CMPLCHA006	CMPLCHA005	CMPLCHA004	CMPLCHA003	CMPLCHA002	CMPLCHA001	CMPLCHA000
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMPLCHA000	Comparison Window A Comparison Condition Select	When the window function is disabled (the ADCMPCR.WCMPE bit is 0) 0: ADCMPDR0 register value > A/D-converted value 1: ADCMPDR0 register value < A/D-converted value When the window function is enabled (the ADCMPCR.WCMPE bit is 1) 0: A/D-converted value < ADCMPDR0 register value or ADCMPDR1 register value < A/D-converted value 1: ADCMPDR0 register value < A/D-converted value < ADCMPDR1 register value	R/W
b1	CMPLCHA001			R/W
b2	CMPLCHA002			R/W
b3	CMPLCHA003			R/W
b4	CMPLCHA004			R/W
b5	CMPLCHA005			R/W
b6	CMPLCHA006			R/W
b15 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The S12AD.ADCMPLR0 register specifies the conditions for comparison of the values in the S12AD.ADCMPDR0/ADCMPDR1 register and the value of A/D conversion. The S12AD.ADCMPLR0 register should be set while the S12AD.ADCSR.ADST bit is 0.

CMPLCHA0n Bit (Comparison Window A Comparison Condition Select) (n = 00 to 06)

The CMPLCHA0n bit specifies the comparison conditions of the channels (AN000 to AN006) targeted for the window A comparison conditions. The condition can be specified for each analog input for comparison. The CMPLCHA000 bit corresponds to AN000 and the CMPLCHA006 bit corresponds to AN006.

(2) S12AD1.ADCMPLR0

Address(es): S12AD1.ADCMPLR0 0008 9298h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	CMPLCHA003	CMPLCHA002	CMPLCHA001	CMPLCHA000
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMPLCHA000	Comparison Window A Comparison Condition Select	When the window function is disabled (the ADCMPCR.WCMPE bit is 0) 0: ADCMPDR0 register value > A/D-converted value 1: ADCMPDR0 register value < A/D-converted value When the window function is enabled (the ADCMPCR.WCMPE bit is 1) 0: A/D-converted value < ADCMPDR0 register value or ADCMPDR1 register value < A/D-converted value 1: ADCMPDR0 register value < A/D-converted value < ADCMPDR1 register value	R/W
b1	CMPLCHA001			R/W
b2	CMPLCHA002			R/W
b3	CMPLCHA003			R/W
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The S12AD1.ADCMPLR0 register specifies the conditions for comparison of the values in the S12AD1.ADCMPDR0/ADCMPDR1 register and the value of A/D conversion. The S12AD1.ADCMPLR0 register should be set while the S12AD1.ADCSR.ADST bit is 0.

CMPLCHA0n Bit (Comparison Window A Comparison Condition Select) (n = 00 to 03)

The CMPLCHA0n bit specifies the comparison conditions of the channels (AN100 to AN103) targeted for the window A comparison conditions. The condition can be specified for each analog input for comparison. The CMPLCHA000 bit corresponds to AN100 and the CMPLCHA003 bit corresponds to AN103.

(3) S12AD2.ADCMPLR0

Address(es): S12AD2.ADCMPLR0 0008 9498h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	CMPLCHA011	CMPLCHA010	CMPLCHA009	CMPLCHA008	CMPLCHA007	CMPLCHA006	CMPLCHA005	CMPLCHA004	CMPLCHA003	CMPLCHA002	CMPLCHA001	CMPLCHA000
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMPLCHA000	Comparison Window A Comparison Condition Select	When the window function is disabled (the ADCMPCR.WCMPE bit is 0) 0: ADCMPDR0 register value > A/D-converted value 1: ADCMPDR0 register value < A/D-converted value When the window function is enabled (the ADCMPCR.WCMPE bit is 1) 0: A/D-converted value < ADCMPDR0 register value or ADCMPDR1 register value < A/D-converted value 1: ADCMPDR0 register value < A/D-converted value < ADCMPDR1 register value	R/W
b1	CMPLCHA001			R/W
b2	CMPLCHA002			R/W
b3	CMPLCHA003			R/W
b4	CMPLCHA004			R/W
b5	CMPLCHA005			R/W
b6	CMPLCHA006			R/W
b7	CMPLCHA007			R/W
b8	CMPLCHA008			R/W
b9	CMPLCHA009			R/W
b10	CMPLCHA010			R/W
b11	CMPLCHA011			R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The S12AD2.ADCMPLR0 register specifies the conditions for comparison of the values in the S12AD2.ADCMPDR0/ADCMPDR1 register and the value of A/D conversion. The S12AD2.ADCMPLR0 register should be set while the S12AD2.ADCSR.ADST bit is 0.

CMPLCHA0n Bit (Comparison Window A Comparison Condition Select) (n = 00 to 11)

The CMPLCHA0n bit specifies the comparison conditions of the channels (AN200 to AN211) targeted for the window A comparison conditions. The condition can be specified for each analog input for comparison. The CMPLCHA000 bit corresponds to AN200 and the CMPLCHA011 bit corresponds to AN211.

When the result of comparison of each analog input matches with the pre-set condition, the ADCMPSR0.CMPSTCHA0n flag (n = 00 to 11) becomes 1 and a comparison interrupt (S12CMPAI, S12CMPAI1, S12CMPAI2) is generated. The conditions for comparison are shown in Figure 42.6.

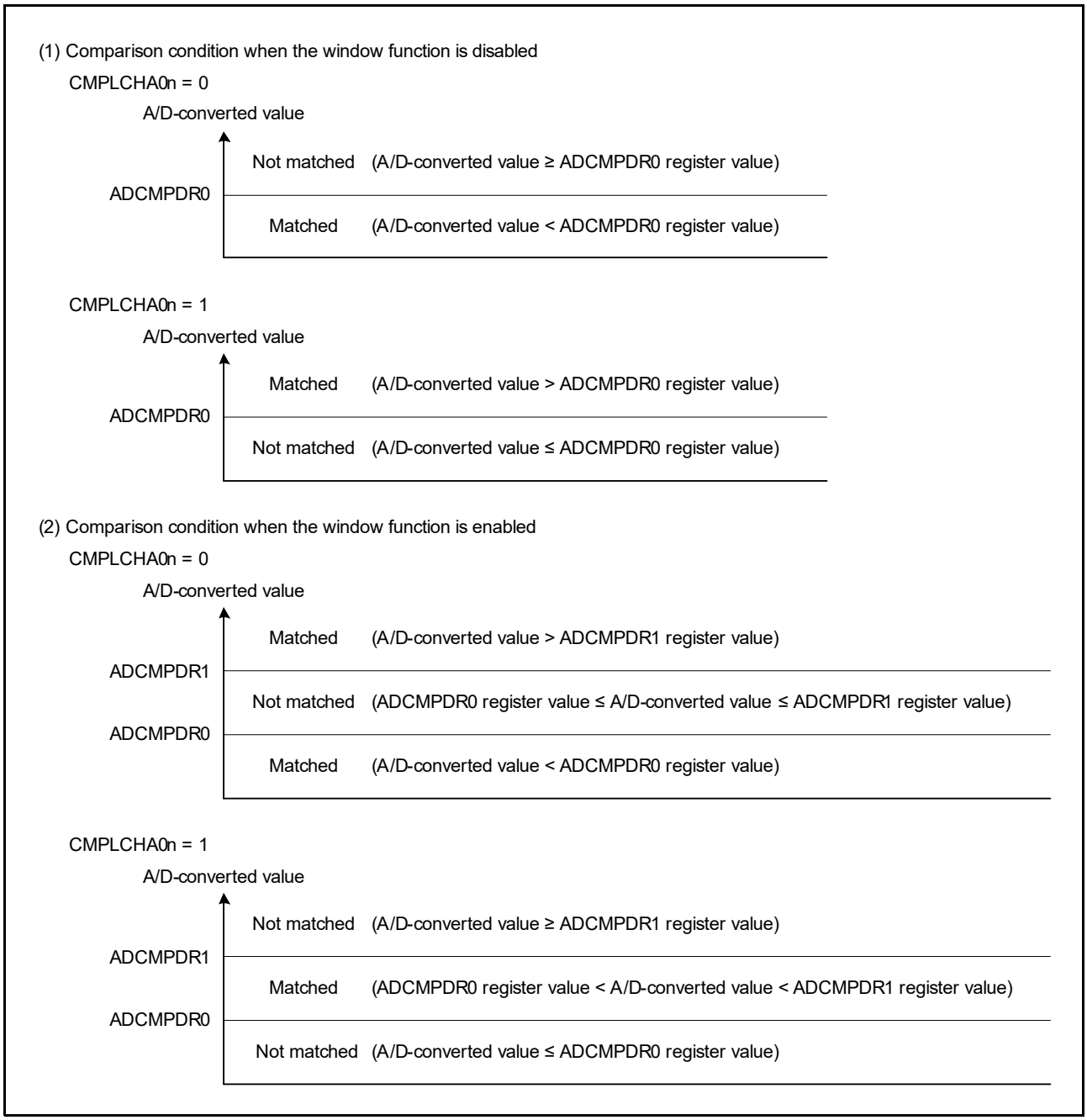


Figure 42.6 Details of Comparison Condition: Comparison Function Window A

42.2.31 A/D Comparison Function Window A Comparison Condition Setting Register 1 (ADCMPLR1)

Address(es): S12AD2.ADCMPLR1 0008 949Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CMPLCHA101	CMPLCHA100
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMPLCHA100	Comparison Window A Comparison Condition Select	When the window function is disabled (the ADCMPCR.WCMPE bit is 0) 0: ADCMPDR0 register value > A/D-converted value 1: ADCMPDR0 register value < A/D-converted value When the window function is enabled (the ADCMPCR.WCMPE bit is 1) 0: A/D-converted value < ADCMPDR0 register value or ADCMPDR1 register value < A/D-converted value 1: ADCMPDR0 register value < A/D-converted value < ADCMPDR1 register value	R/W
b1	CMPLCHA101			R/W
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADCMPLR1 register specifies the conditions for comparison of the values in the ADCMPDR0/ADCMPDR1 register and the value of A/D conversion. The ADCMPLR1 register should be set while the ADCSR.ADST bit is 0.

CMPLCHA1n Bit (Comparison Window A Comparison Condition Select) (n = 00, 01)

The CMPLCHA1n bit specifies the comparison conditions of the channels (AN216 to AN217) targeted for the window A comparison conditions. The condition can be specified for each analog input for comparison. The CMPLCHA100 bit corresponds to AN216 and the CMPLCHA101 bit corresponds to AN217.

When the result of comparison of each analog input matches with the pre-set condition, the S12AD2.ADCMPSR1.CMPSTCHA1n flag becomes 1 and a comparison interrupt (S12CMPAI2) is generated. The conditions for comparison are shown in Figure 42.6.

42.2.32 A/D Comparison Function Window A Extended Input Comparison Condition Setting Register (ADCMPLER)

Address(es): S12AD2.ADCMPLER 0008 9493h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	CMPLOC	CMPLTS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMPLTS	Comparison Window A Temperature Sensor Output Comparison Condition Select	When the window A function is disabled (the ADCMPCR.WCMPE bit is 0) 0: ADCMPDR0 register value > A/D-converted value 1: ADCMPDR0 register value < A/D-converted value When the window A function is enabled (the ADCMPCR.WCMPE bit is 1) 0: A/D-converted value < ADCMPDR0 register value or ADCMPDR1 register value < A/D-converted value 1: ADCMPDR0 register value < A/D-converted value < ADCMPDR1 register value	R/W
b1	CMPLOC	Comparison Window A Internal Reference Voltage Comparison Condition Select	When the window A function is disabled (the ADCMPCR.WCMPE bit is 0) 0: ADCMPDR0 register value > A/D-converted value 1: ADCMPDR0 register value < A/D-converted value When the window A function is enabled (the ADCMPCR.WCMPE bit is 1) 0: A/D-converted value < ADCMPDR0 register value or ADCMPDR1 register value < A/D-converted value 1: ADCMPDR0 register value < A/D-converted value < ADCMPDR1 register value	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADCMPLER register specifies the conditions for comparison of the values in the ADCMPDR0/ADCMPDR1 register and the value of A/D conversion. The ADCMPLER register should be set while the ADCSR.ADST bit is 0.

CMPLTS Bit (Comparison Window A Temperature Sensor Output Comparison Condition Select)

The CMPLTS bit selects the comparison conditions of temperature sensor output targeted for window A.
When the result of comparison of temperature sensor output matches with the pre-set condition, the ADCMPSER.CMPFTS flag becomes 1. A comparison interrupt (S12CMPAI2) is generated.
Figure 42.6 shows the conditions for comparison.

CMPLOC Bit (Comparison Window A Internal Reference Voltage Comparison Condition Select)

The CMPLOC bit selects the comparison conditions of the internal reference voltage targeted for window A.
When the result of comparison of the internal reference voltage matches with the pre-set condition, the ADCMPSER.CMPFOC flag becomes 1. A comparison interrupt (S12CMPAI2) is generated.
Figure 42.6 shows the conditions for comparison.

42.2.33 A/D Comparison Function Window A Lower Level Setting Register (ADCMPDR0)

Address(es): S12AD.ADCMPDR0 0008 909Ch, S12AD1.ADCMPDR0 0008 929Ch, S12AD2.ADCMPDR0 0008 949Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The ADCMPDR0 register is a readable and writable register that specifies the reference data when the comparison window A function is used. The ADCMPDR0 register specifies the lower level of window A.
Writing to the ADCMPDR0 register is enabled even when A/D conversion is in progress. Rewriting the register value during the A/D conversion dynamically changes the reference data.
Satisfy the condition of [Upper limit level ≥ Lower limit level (ADCMPDR1 setting value ≥ ADCMPDR0 setting value)] when setting.

The data formats of the register for each given condition are shown below.

- Setting of the A/D data register format select bit (right-justified or left-justified format)
- Setting of the A/D-converted value addition/averaging function channel select register (A/D-converted value addition/averaging mode is enabled or disabled)
- Setting of the A/D-converted value addition/average count select register (addition/averaging mode is selected and addition count is selected)

Note: When the comparison values are set in a different format from that of the A/D data register y (ADDRy), correct result of comparison cannot be obtained.

(1) When A/D-Converted Value Addition/Average Mode is Disabled

- Right-justified format
  The comparison level (lower) to be compared is set in bits 11 to 0. Write 0 to bits 15 to 12.
- Left-justified format
  The comparison level (lower) to be compared is set in bits 15 to 4. Write 0 to bits 3 to 0.

(2) When A/D-Converted Average Mode is Selected

A/D-converted value average mode can be set only when 2- or 4-time conversion is selected in A/D-converted value addition mode.

- Right-justified format
  The comparison level (lower) to be compared with the value of A/D conversion of the same channel is set in bits 11 to 0. Write 0 to bits 15 to 12.
- Left-justified format
  The comparison level (lower) to be compared with the value of A/D conversion of the same channel is set in bits 15 to 4. Write 0 to bits 3 to 0.

(3) When A/D-Converted Value Addition Mode is Selected

When A/D-converted value addition mode is selected, the value for comparison with the result of cumulative A/D-converted value of the same channel is specified. Conversion 1, 2, 3, 4, or 16 times is selectable. When A/D-converted value addition mode is selected and if conversion 1 to 4 is selected, the resolution of the results of conversion will be extended by 2 bits and so set a value for comparison with the results in the ADCMPDR0 register. When conversion 16 times is selected, the resolution of the results of conversion will be extended by 4 bits and so set a value for

comparison with the results in the ADCMPDR0 register.
Even when A/D-converted value addition mode is selected, set the value for reference based on the setting of the A/D data register format select bit.

- Right-justified format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times)
  The comparison level (lower) to be compared with the value of A/D conversion of the same channel is set in bits 13 to 0. Write 0 to bits 15 and 14.
- Right-justified format (in A/D-converted value addition mode and when number of conversions is selected for 16 times)
  The comparison level (lower) to be compared with the value of A/D conversion of the same channel is set in bits 15 to 0.
- Left-justified format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times)
  The comparison level (lower) to be compared with the A/D-converted value of the same channel is set in bits 15 to 2. Write 0 to bits 1 and 0.
- Left-justified format (in A/D-converted value addition mode and when number of conversions is selected for 16 times)
  The comparison level (lower) to be compared with the value of A/D conversion of the same channel is set in bits 15 to 0.

42.2.34 A/D Comparison Function Window A Upper Level Setting Register (ADCMPDR1)

Address(es): S12AD.ADCMPDR1 0008 909Eh, S12AD1.ADCMPDR1 0008 929Eh, S12AD2.ADCMPDR1 0008 949Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The ADCMPDR1 register is a readable and writable register that specifies the data for reference when the comparison window A function is used. The register specifies the upper level of window A.
A write operation to the ADCMPDR1 register is enabled even during A/D conversion. Rewriting the value of the register during A/D conversion dynamically changes the reference data.
Satisfy the condition of [upper limit level ≥ lower limit level (ADCMPDR1 setting value ≥ ADCMPDR0 setting value)] when setting.
The ADCMPDR1 register is not used when the window function is disabled.

The data formats of the register for each given condition are shown below.

- Setting of the A/D data register format select bit (right-justified or left-justified format)
- Setting of the A/D-converted value addition/averaging function channel select register (A/D-converted value addition/averaging mode is enabled or disabled)
- Setting of the A/D-converted value addition/averaging time select register (addition/averaging mode is selected and addition count is selected)

Note: When the comparison values are set in a different format from that of the A/D data register y (ADDRy), correct result of comparison cannot be obtained.

(1) When A/D-Converted Value Addition/Average Mode is Disabled

- Right-justified format
  The comparison level (upper) is set in bits 11 to 0. Write 0 to bits 15 to 12.
- Left-justified format
  The comparison level (upper) is set in bits 15 to 4. Write 0 to bits 3 to 0.

(2) When A/D-Converted Average Mode is Selected

A/D-converted value average mode can be set only when 2- or 4-time conversion is selected in A/D-converted value addition mode.

- Right-justified format
  The comparison level (upper) to be compared with the A/D-converted results of the same channel is set in bits 11 to 0. Write 0 to bits 15 to 12.
- Left-justified format
  The comparison level (upper) to be compared with the A/D-converted results of the same channel is set in bits 15 to 4. Write 0 to bits 3 to 0.

(3) When A/D-Converted Value Addition Mode is Selected

When A/D-converted value addition mode is selected, the value for comparison with the result of cumulative A/D-converted value of the same channel is specified. Conversion 1, 2, 3, 4, or 16 times is selectable. When A/D-converted value addition mode is selected and if conversion 1 to 4 is selected, the resolution of the results of conversion will be extended by 2 bits and so set a value for comparison with the results in the ADCMPDR1 register. When conversion 16

times is selected, the resolution of the results of conversion will be extended by 4 bits and so set a value for comparison with the results in the ADCMPDR1 register.
Even when A/D-converted value addition mode is selected, set the value for reference based on the setting of the A/D data register format select bit.

- Right-justified format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times)
  The comparison level (upper) to be compared with the A/D-converted value of the same channel is set bits 13 to 0. Write 0 to bits 15 and 14.
- Right-justified format (in A/D-converted value addition mode and when number of conversions is selected for 16 times)
  The comparison level (upper) to be compared with the A/D-converted value of the same channel is set bits 15 to 0.
- Left-justified format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times)
  The comparison level (upper) to be compared with the A/D-converted value of the same channel is set in bits 15 to 2. Write 0 to bits 1 and 0.
- Left-justified format (in A/D-converted value addition mode and when number of conversions is selected for 16 times)
  The comparison level (upper) to be compared with the A/D-converted value of the same channel is set bits 15 to 0.

42.2.35 A/D Comparison Function Window A Channel Status Register 0 (ADCMPSR0)

(1) S12AD.ADCMPSR0

Address(es): S12AD.ADCMPSR0 0008 90A0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	CMPSTCHA006	CMPSTCHA005	CMPSTCHA004	CMPSTCHA003	CMPSTCHA002	CMPSTCHA001	CMPSTCHA000
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMPSTCHA000	Comparison Window A Flag	When window A operation is enabled (ADCMPCR.CMPAE = 1) These bits indicate the comparison result of the channels (AN000 to AN006) for the window A comparison condition 0: The comparison condition is not satisfied. 1: The comparison condition is satisfied.	R/(W) *1
b1	CMPSTCHA001			R/(W) *1
b2	CMPSTCHA002			R/(W) *1
b3	CMPSTCHA003			R/(W) *1
b4	CMPSTCHA004			R/(W) *1
b5	CMPSTCHA005			R/(W) *1
b6	CMPSTCHA006			R/(W) *1
b15 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

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

The S12AD.ADCMPSR0 register is a register that stores the result of comparison by the comparison window A function.

CMPSTCHA0n Flag (Comparison Window A Flag) (n = 00 to 06)

The CMPSTCHA0n flag is a status flag that indicate the result of comparison of the channels (AN000 to AN006) of the comparison condition for window A. When the conversion result matches with the comparison condition specified in the ADCMPLR0.CMPLCHA0n bit on completion of A/D conversion, this flag becomes 1.

When the ADCMPCR.CMPAIE bit is 1, a comparison interrupt request (S12CMPAI) is generated at the setting of the flag. The CMPSTCHA000 flag corresponds to AN000 and the CMPSTCHA006 flag corresponds to AN006.

1 cannot be written to the CMPSTCHA0n flag.

[Setting condition]

- When ADCMPCR.CMPAE bit = 1 and the condition set in the ADCMPLR0.CMPLCHA0n bit is satisfied.

[Clearing condition]

- When 0 is written after reading the state of 1.

(2) S12AD1.ADCMPSR0

Address(es): S12AD1.ADCMPSR0 0008 92A0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	CMPST CHA003	CMPST CHA002	CMPST CHA001	CMPST CHA000
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMPSTCHA000	Comparison Window A Flag	When window A operation is enabled (ADCMPCR.CMPAE = 1) These bits indicate the comparison result of the channels (AN100 to AN103) for the window A comparison condition 0: The comparison condition is not satisfied. 1: The comparison condition is satisfied.	R/(W) *1
b1	CMPSTCHA001			R/(W) *1
b2	CMPSTCHA002			R/(W) *1
b3	CMPSTCHA003			R/(W) *1
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

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

The S12AD1.ADCMPSR0 register is a register that stores the result of comparison by the comparison window A function.

CMPSTCHA0n Flag (Comparison Window A Flag) (n = 00 to 03)

The CMPSTCHA0n flag is a status flag that indicate the result of comparison of the channels (AN100 to AN103) of the comparison condition for window A. When the conversion result matches with the comparison condition specified in the ADCMPLR0.CMPLCHA0n bit on completion of A/D conversion, this flag becomes 1.
When the ADCMPCR.CMPAIE bit is 1, a comparison interrupt request (S12CMPAI1) is generated at the setting of the flag. The CMPSTCHA000 flag corresponds to AN100 and the CMPSTCHA003 flag corresponds to AN103.
1 cannot be written to the CMPSTCHA0n flag.
[Setting condition]

- When ADCMPCR.CMPAE bit = 1 and the condition set in the ADCMPLR0.CMPLCHA0n bit is satisfied.

[Clearing condition]

- When 0 is written after reading the state of 1.

(3) S12AD2.ADCMPSR0

Address(es): S12AD2.ADCMPSR0 0008 94A0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	CMPST CHA011	CMPST CHA010	CMPST CHA009	CMPST CHA008	CMPST CHA007	CMPST CHA006	CMPST CHA005	CMPST CHA004	CMPST CHA003	CMPST CHA002	CMPST CHA001	CMPST CHA000
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMPSTCHA000	Comparison Window A Flag	When window A operation is enabled (ADCMPCR.CMPAE = 1) These bits indicate the comparison result of the channels (AN200 to AN211) for the window A comparison 0: The comparison condition is not satisfied. 1: The comparison condition is satisfied.	R/(W) *1
b1	CMPSTCHA001			R/(W) *1
b2	CMPSTCHA002			R/(W) *1
b3	CMPSTCHA003			R/(W) *1
b4	CMPSTCHA004			R/(W) *1
b5	CMPSTCHA005			R/(W) *1
b6	CMPSTCHA006			R/(W) *1
b7	CMPSTCHA007			R/(W) *1
b8	CMPSTCHA008			R/(W) *1
b9	CMPSTCHA009			R/(W) *1
b10	CMPSTCHA010			R/(W) *1
b11	CMPSTCHA011			R/(W) *1
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

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

The S12AD2.ADCMPSR0 register is a register that stores the result of comparison by the comparison window A function.

CMPSTCHA0n Flag (Comparison Window A Flag) (n = 00 to 11)

The CMPSTCHA0n flag is a status flag that indicate the result of comparison of the channels (AN200 to AN211) of the comparison condition for window A. When the conversion result matches with the comparison condition specified in the ADCMPLR0.CMPLCHA0n bit on completion of A/D conversion, this flag becomes 1.
When the ADCMPCR.CMPAIE bit is 1, a comparison interrupt request (S12CMPAI2) is generated at the setting of the flag. The CMPSTCHA000 flag corresponds to AN200 and the CMPSTCHA011 flag corresponds to AN211.
1 cannot be written to the CMPSTCHA0n flag.
[Setting condition]

- When ADCMPCR.CMPAE bit = 1 and the condition set in the ADCMPLR0.CMPLCHA0n bit is satisfied.

[Clearing condition]

- When 0 is written after reading the state of 1.

42.2.36 A/D Comparison Function Window A Channel Status Register 1 (ADCMPSR1)

Address(es): S12AD2.ADCMPSR1 0008 94A2h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CMPSTCHA101	CMPSTCHA100
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMPSTCHA100	Comparison Window A Flag	When window A operation is enabled (ADCMPCR.CMPAE = 1) These bits indicate the comparison result of the channels (AN216 to AN217) for the window A comparison condition 0: The comparison condition is not satisfied. 1: The comparison condition is satisfied.	R/(W) *1
b1	CMPSTCHA101			R/(W) *1
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

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

The ADCMPSR1 register is a register that stores the result of comparison by the comparison window A function.

CMPSTCHA1n Flag (Comparison Window A Flag) (n = 00, 01)

The CMPSTCHA1n flag is a status flag that indicate the result of comparison of the channels (AN216 to AN217) of the comparison condition for window A. When the conversion result matches with the comparison condition specified in the ADCMPLR1.CMPLCHA1n bit on completion of A/D conversion, this flag becomes 1. When the ADCMPCR.CMPAIE bit is 1, a comparison interrupt request (S12CMPAI2) is generated upon the setting of the flag. The CMPSTCHA100 flag corresponds to AN216 and the CMPSTCHA101 flag corresponds to AN217.

1 cannot be written to the CMPSTCHA1n flag.

[Setting condition]

- When ADCMPCR.CMPAE bit = 1 and the condition set in the ADCMPLR1.CMPLCHA1n bit is satisfied.

[Clearing condition]

- When 0 is written after reading the state of 1.

42.2.37 A/D Comparison Function Window A Extended Input Channel Status Register (ADCMPSER)

Address(es): S12AD2.ADCMPSER 0008 94A4h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	CMPFOC	CMPFTS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMPFTS	Comparison Window A Temperature Sensor Output Comparison Flag	When window A operation is enabled (ADCMPCR.CMPAE = 1) This bit indicates the comparison result of the temperature sensor output 0: The comparison condition is not satisfied. 1: The comparison condition is satisfied.	R/(W) *1
b1	CMPFOC	Comparison Window A Internal Reference Voltage Comparison Flag	When window A operation is enabled (ADCMPCR.CMPAE = 1) This bit indicates the comparison result of the internal reference voltage 0: The comparison condition is not satisfied. 1: The comparison condition is satisfied.	R/(W) *1
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

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

The ADCMPSER register is a register that stores the result of comparison by the comparison window A function.

CMPFTS Flag (Comparison Window A Temperature Sensor Output Comparison Flag)

The CMPFTS flag is a status flag that indicates the result of comparison of the temperature sensor output. When the conversion result matches with the comparison condition specified in the ADCMPLER.CMPLTS bit upon completion of A/D conversion, this flag becomes 1. When the ADCMPCR.CMPAIE bit is 1, a comparison interrupt request (S12CMPAI2) is generated upon the setting of the flag.
1 cannot be written to the CMPFTS flag.
[Setting condition]

- When ADCMPCR.CMPAE bit = 1 and the condition set in the ADCMPLER.CMPLTS bit is satisfied.

[Clearing condition]

- When 0 is written after reading the state of 1.

CMPFOC Flag (Comparison Window A Internal Reference Voltage Comparison Flag)

The CMPFOC flag is a status flag that indicates the result of comparison of the internal reference voltage. When the conversion result matches with the comparison condition specified in the ADCMPLER.CMPLOC bit upon completion of A/D conversion, this flag becomes 1. When the ADCMPCR.CMPAIE bit is 1, a comparison interrupt request (S12CMPAI2) is generated upon the setting of the flag.
1 cannot be written to the CMPFOC flag.
[Setting condition]

- When ADCMPCR.CMPAE bit = 1 and the condition set in the ADCMPLER.CMPLOC bit is satisfied.

[Clearing condition]

- When 0 is written after reading the state of 1.

42.2.38 A/D Comparison Function Window A/B Status Monitoring Register (ADWINMON)

Address(es): S12AD.ADWINMON 0008 908Ch, S12AD1.ADWINMON 0008 928Ch, S12AD2.ADWINMON 0008 948Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	MONCMPB	MONCMPA	—	—	—	MONCOMB
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MONCOMB	Combination Result Monitoring Flag	This flag indicates the combination result. This flag is valid when both window A operation and window B operation are enabled. 0: The complex condition for window A/B is not satisfied. 1: The complex condition for window A/B is satisfied.	R
b3 to b1	—	Reserved	These bits are read as 0.	R
b4	MONCMPA	Comparison Result Monitor A Flag	0: The comparison condition for window A is not satisfied. 1: The comparison condition for window A is satisfied.	R
b5	MONCMPB	Comparison Result Monitor B Flag	0: The comparison condition for window B is not satisfied. 1: The comparison condition for window B is satisfied.	R
b7, b6	—	Reserved	These bits are read as 0.	R

The ADWINMON register can monitor the results of comparison and the combined result.

MONCOMB Flag (Combination Result Monitoring Flag)

The MONCOMB bit is a dedicated bit for reading the combined result of comparison condition A and B that were set in the ADCMPCR.CMPAB[1:0] bits as a combined condition.

[Setting condition]

- When ADCMPCR.CMPAE bit = 1 and ADCMPCR.CMPBE bit = 1, the condition set in the ADCMPCR.CMPAB[1:0] bits is satisfied.

[Clearing condition]

- No match with the combined condition set in the ADCMPCR.CMPAB[1:0] bits
- When ADCMPCR.CMPAE bit = 0 or ADCMPCR.CMPBE bit = 0

MONCMPA Flag (Comparison Result Monitor A Flag)

The MONCMPA bit is a dedicated bit for reading 1 when the condition set in the ADCMPLR0, ADCMPLR1, or ADCMPLER register matches with the A/D-converted value of the channel targeted for window A and for reading 0 when such condition is not satisfied.

[Setting condition]

- When ADCMPCR.CMPAE bit = 1, the condition set in the ADCMPLR0.CMPLCHA0n bit is satisfied.

[Clearing condition]

- When ADCMPCR.CMPAE bit = 1, the condition set in the ADCMPLR0.CMPLCHA0n bit is not satisfied.
- When ADCMPCR.CMPAE bit = 0 (the ADCMPCR.CMPAE bit is automatically cleared to 0).

MONCMPB Flag (Comparison Result Monitor B Flag)

The MONCMPB bit is a dedicated bit for reading 1 when the condition set in the ADCMPBNSR.CMPLB bit matches with the A/D-converted value of the channel targeted for window B and for reading 0 when such condition is not satisfied.

[Setting condition]

- When ADCMPCR.CMPBE bit = 1, the condition set in the ADCMPBNSR.CMPLB bit is satisfied.

[Clearing condition]

- When ADCMPCR.CMPBE bit = 1, the condition set in the ADCMPBNSR.CMPLB bit is not satisfied.
- When ADCMPCR.CMPBE bit = 0 (the ADCMPCR.CMPBE bit is automatically cleared to 0).

42.2.39 A/D Comparison Function Window B Channel Select Register (ADCMPBNSR)

(1) S12AD.ADCMPBNSR

Address(es): S12AD.ADCMPBNSR 0008 90A6h

	b7	b6	b5	b4	b3	b2	b1	b0
	CMPLB	—	CMPCHB[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	CMPCHB[5:0]	Comparison Window B Channel Select	Select the channel that compares with the condition of comparison window B b5 b0 0 0 0 0 0 0: AN000 0 0 0 0 0 1: AN001 0 0 0 0 1 0: AN002 : : 0 0 0 1 0 1: AN005 0 0 0 1 1 0: AN006 Settings other than above are prohibited.	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	CMPLB	Comparison Window B Comparison Condition Setting	When the window function is disabled (the ADCMPCR.WCMPE bit is 0) 0: ADWINLLB register value > A/D-converted value 1: ADWINLLB register value < A/D-converted value When the window function is enabled (the ADCMPCR.WCMPE bit is 1) 0: A/D-converted value < ADWINLLB register value or ADWINULB register value < A/D-converted value 1: ADWINLLB register value < A/D-converted value < ADWINULB register value	R/W

The S12AD.ADCMPBNSR register is used to specify the comparison window B function.

CMPCHB[5:0] Bits (Comparison Window B Channel Select)

The CMPCHB[5:0] bits select the channel to be compared with the comparison window B from AN000 to AN006. When the number of the A/D conversion channel selected in the ADANSAy.ANSAyn bit (y = 0 to 1, n = 00 to 06) and the ADANSBy.ANSByn bit is specified, the comparison window B function is enabled.
The CMPCHB[5:0] bits should be set while the ADCSR.ADST bit is 0.

CMPLB Bit (Comparison Window B Comparison Condition Setting)

The CMPLB bit specifies the condition for comparison of the channels for window B. When the result of comparison of each analog input matches with the pre-set condition, the ADCMPBSR.CMPSTB flag becomes 1 and a comparison interrupt (S12CMPBI) is generated. The conditions for comparison are shown in Figure 42.7.

(2) S12AD1.ADCMPBNSR

Address(es): S12AD1.ADCMPBNSR 0008 92A6h

	b7	b6	b5	b4	b3	b2	b1	b0
	CMPLB	—	CMPCHB[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	CMPCHB[5:0]	Comparison Window B Channel Select	Select the channel that compares with the condition of comparison window B b5 b0 0 0 0 0 0 0: AN100 0 0 0 0 0 1: AN101 0 0 0 0 1 0: AN102 0 0 0 0 1 1: AN103 Settings other than above are prohibited.	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	CMPLB	Comparison Window B Comparison Condition Setting	When the window function is disabled (the ADCMPCR.WCMPE bit is 0) 0: ADWINLLB register value > A/D-converted value 1: ADWINLLB register value < A/D-converted value When the window function is enabled (the ADCMPCR.WCMPE bit is 1) 0: A/D-converted value < ADWINLLB register value or ADWINULB register value < A/D-converted value 1: ADWINLLB register value < A/D-converted value < ADWINULB register value	R/W

The S12AD1.ADCMPBNSR register is used to specify the comparison window B function.

CMPCHB[5:0] Bits (Comparison Window B Channel Select)

The CMPCHB[5:0] bits select the channel to be compared with the comparison window B from AN100 to AN103.
When the number of the A/D conversion channel selected in the ADANSAy.ANSAyn bit (y = 0 to 1, n = 00 to 03) and the ADANSBy.ANSByn bit is specified, the comparison window B function is enabled.
The CMPCHB[5:0] bits should be set while the ADCSR.ADST bit is 0.

CMPLB Bit (Comparison Window B Comparison Condition Setting)

The CMPLB bit specifies the condition for comparison of the channels for window B. When the result of comparison of each analog input matches with the pre-set condition, the ADCMPBSR.CMPSTB flag becomes 1 and a comparison interrupt (S12CMPBI1) is generated. The conditions for comparison are shown in Figure 42.7.

(3) S12AD2.ADCMPBNSR

Address(es): S12AD2.ADCMPBNSR 0008 94A6h

	b7	b6	b5	b4	b3	b2	b1	b0
	CMPLB	—	CMPCHB[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	CMPCHB[5:0]	Comparison Window B Channel Select	Select the channel that compares with the condition of comparison window B b5 b0 0 0 0 0 0 0: AN200 0 0 0 0 0 1: AN201 0 0 0 0 1 0: AN202 : : 0 0 0 1 1 0: AN206 0 0 0 1 1 1: AN207 : : 0 0 1 0 1 0: AN210 0 0 1 0 1 1: AN211 0 1 0 0 0 0: AN216 0 1 0 0 0 1: AN217 1 0 0 0 0 0: Temperature sensor 1 0 0 0 0 1: Internal reference voltage Settings other than above are prohibited.	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	CMPLB	Comparison Window B Comparison Condition Setting	When the window function is disabled (the ADCMPCR.WCMPE bit is 0) 0: ADWINLLB register value > A/D-converted value 1: ADWINLLB register value < A/D-converted value When the window function is enabled (the ADCMPCR.WCMPE bit is 1) 0: A/D-converted value < ADWINLLB register value or ADWINULB register value < A/D-converted value 1: ADWINLLB register value < A/D-converted value < ADWINULB register value	R/W

The S12AD2.ADCMPBNSR register is used to specify the comparison window B function.

CMPCHB[5:0] Bits (Comparison Window B Channel Select)

The CMPCHB[5:0] bits select the channel to be compared under the comparison window B condition from AN200 to AN211, AN216, AN217, temperature sensor, and internal reference voltage.

When the number of the A/D conversion channel selected in the ADANSAy.ANSAyn bit (y = 0, 1, n = 00 to 11) and the ADANSBy.ANSByn bit is specified, the comparison window B function is enabled.

The CMPCHB[5:0] bits should be set while the ADCSR.ADST bit is 0.

CMPLB Bit (Comparison Window B Comparison Condition Setting)

The CMPLB bit specifies the condition for comparison of the channels for window B. When the result of comparison of each analog input matches with the pre-set condition, the ADCMPBSR.CMPSTB flag becomes 1 and a comparison interrupt (S12CMPBI2) is generated. The conditions for comparison are shown in Figure 42.7.

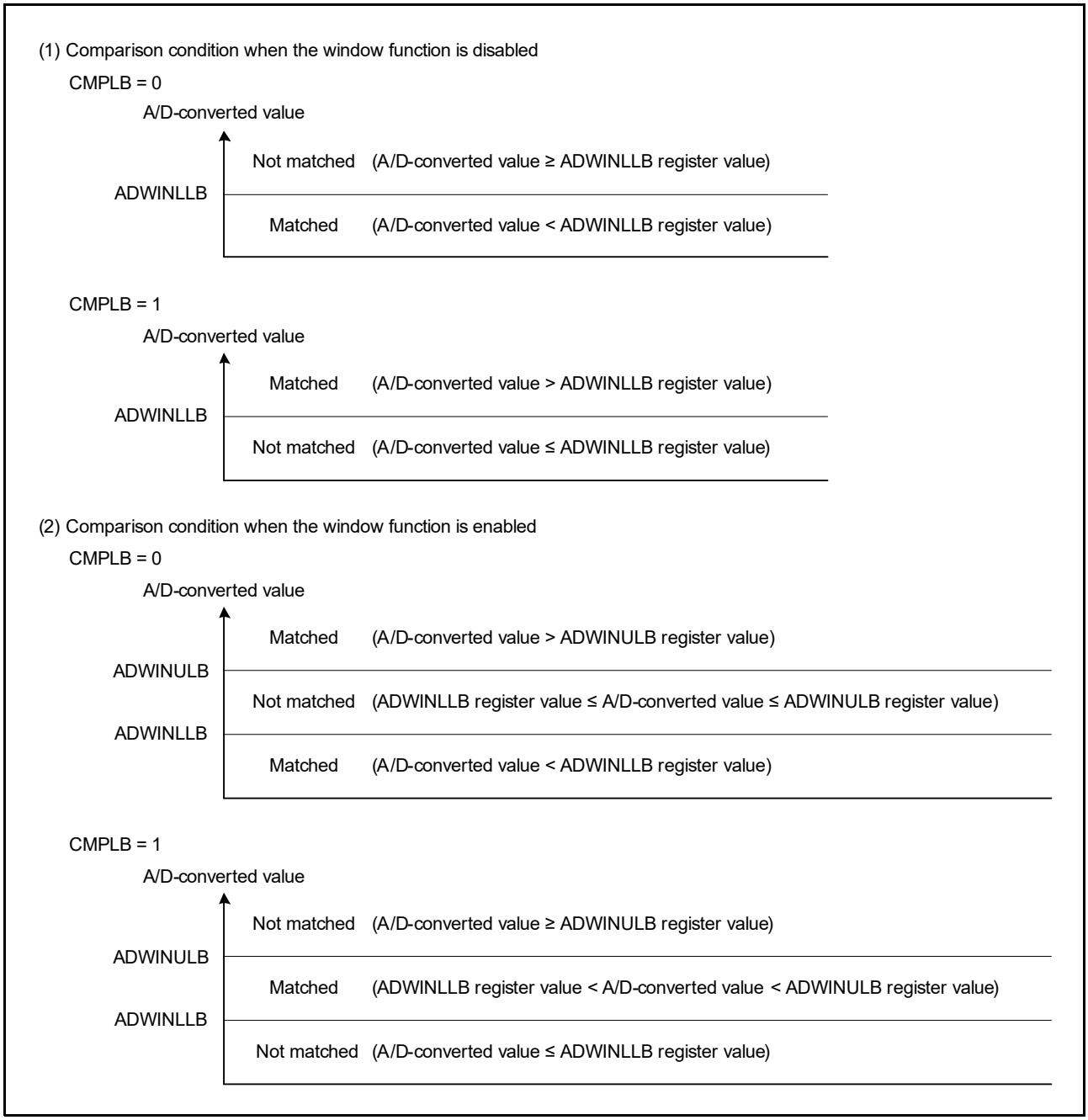


Figure 42.7 Details of Comparison Condition: Comparison Function Window B

42.2.40 A/D Comparison Function Window B Lower Level Setting Register (ADWINLLB)

Address(es): S12AD.ADWINLLB 0008 90A8h, S12AD1.ADWINLLB 0008 92A8h, S12AD2.ADWINLLB 0008 94A8h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The ADWINLLB register is a readable and writable register that specifies reference data when comparison window B is in use. The register specifies the lower level of window B.
A write operation to the ADWINLLB register is enabled even during A/D conversion. Rewriting the value of the register during A/D conversion dynamically changes the reference data.
Satisfy the condition of [upper limit level ≥ lower limit level (ADWINULB register setting value ≥ ADWINLLB register setting value)] when setting.

The data formats of the register for each given condition are shown below.

- Setting of the A/D data register format select bit (right-justified or left-justified format)
- Setting of the A/D-converted value addition/averaging function channel select register (A/D-converted value addition/averaging mode is enabled or disabled)
- Setting of the A/D-converted value addition/averaging count select register (addition/averaging mode is selected and addition count is selected)

Note: When the comparison values are set in a different format from that of the A/D data register y (ADDRy), correct result of comparison cannot be obtained.

(1) When A/D-Converted Value Addition/Average Mode is Disabled

- Right-justified format
  The comparison level (lower) to be compared is set in bits 11 to 0. Write 0 to bits 15 to 12.
- Left-justified format
  The comparison level (lower) to be compared is set in bits 15 to 4. Write 0 to bits 3 to 0.

(2) When A/D-Converted Average Mode is Selected

A/D-converted value average mode can be set only when 2- or 4-time conversion is selected in A/D-converted value addition mode.

- Right-justified format
  The comparison level (lower) to be compared with the value of A/D conversion of the same channel is set in bits 11 to 0. Write 0 to bits 15 to 12.
- Left-justified format
  The comparison level (lower) to be compared with the value of A/D conversion of the same channel is set in bits 15 to 4. Write 0 to bits 3 to 0.

(3) When A/D-Converted Value Addition Mode is Selected

When A/D-converted value addition mode is selected, the value for comparison with the result of cumulative A/D-converted value of the same channel is specified. Conversion 1, 2, 3, 4, or 16 times is selectable. When A/D-converted value addition mode is selected and if conversion 1 to 4 is selected, the resolution of the results of conversion will be extended by 2 bits and so set a value for comparison with the results in the ADWINLLB register. When conversion 16 times is selected, the resolution of the results of conversion will be extended by 4 bits and so set a value for comparison with the results in the ADWINLLB register.

Even when A/D-converted value addition mode is selected, set the value for reference based on the setting of the A/D data register format select bit.

- Right-justified format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times)
  The comparison level (lower) to be compared with the value of A/D conversion of the same channel is set in bits 13 to 0. Write 0 to bits 15 and 14.
- Right-justified format (in A/D-converted value addition mode and when number of conversions is selected for 16 times)
  The comparison level (lower) to be compared with the value of A/D conversion of the same channel is set in bits 15 to 0.
- Left-justified format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times)
  The comparison level (lower) to be compared with the A/D-converted value of the same channel is set in bits 15 to 2. Write 0 to bits 1 and 0.
- Left-justified format (in A/D-converted value addition mode and when number of conversions is selected for 16 times)
  The comparison level (lower) to be compared with the value of A/D conversion of the same channel is set in bits 15 to 0.

42.2.41 A/D Comparison Function Window B Upper Level Setting Register (ADWINULB)

Address(es): S12AD.ADWINULB 0008 90AAh, S12AD1.ADWINULB 0008 92AAh, S12AD2.ADWINULB 0008 94AAh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The ADWINULB register is a readable and writable register that specifies the data for reference when the comparison window B function is used. The register specifies the upper level of window B.
A write operation to the register is enabled even during A/D conversion. Rewriting the value of the register during A/D conversion dynamically changes the reference data.
Satisfy the condition of [upper limit level ≥ lower limit level (ADWINULB register setting value ≥ ADWINLLB register setting value)] when setting.
The register is not used when the window function is disabled.

The data formats of the register for each given condition are shown below.

- Setting of the A/D data register format select bit (right-justified or left-justified format)
- Setting of the A/D-converted value addition/averaging function channel select register (A/D-converted value addition/averaging mode is enabled or disabled)
- Setting of the A/D-converted value addition/averaging count select register (addition/averaging mode is selected and addition count is selected)

Note: When the comparison values are set in a different format from that of the A/D data register y (ADDRy), correct result of comparison cannot be obtained.

(1) When A/D-Converted Value Addition/Average Mode is Disabled

- Right-justified format
  The comparison level (upper) is set in bits 11 to 0. Write 0 to bits 15 to 12.
- Left-justified format
  The comparison level (upper) is set in bits 15 to 4. Write 0 to bits 3 to 0.

(2) When A/D-Converted Average Mode is Selected

A/D-converted value average mode can be set only when 2- or 4-time conversion is selected in A/D-converted value addition mode.

- Right-justified format
  The comparison level (upper) to be compared with the A/D-converted results of the same channel is set in bits 11 to 0. Write 0 to bits 15 to 12.
- Left-justified format
  The comparison level (upper) to be compared with the A/D-converted results of the same channel is set in bits 15 to 4. Write 0 to bits 3 to 0.

(3) When A/D-Converted Value Addition Mode is Selected

When A/D-converted value addition mode is selected, the value for comparison with the result of cumulative A/D-converted value of the same channel is specified. Conversion 1, 2, 3, 4, or 16 times is selectable. When A/D-converted value addition mode is selected and if conversion 1 to 4 is selected, the resolution of the results of conversion will be extended by 2 bits and so set a value for comparison with the results in the ADWINULB register. When conversion 16

times is selected, the resolution of the results of conversion will be extended by 4 bits and so set a value for comparison with the results in the ADWINULB register.
Even when A/D-converted value addition mode is selected, set the value for reference based on the setting of the A/D data register format select bit.

- Right-justified format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times)
  The comparison level (upper) to be compared with the A/D-converted value of the same channel is set bits 13 to 0. Write 0 to bits 15 and 14.
- Right-justified format (in A/D-converted value addition mode and when number of conversions is selected for 16 times)
  The comparison level (upper) to be compared with the A/D-converted value of the same channel is set bits 15 to 0.
- Left-justified format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times)
  The comparison level (upper) to be compared with the A/D-converted value of the same channel is set in bits 15 to 2. Write 0 to bits 1 and 0.
- Left-justified format (in A/D-converted value addition mode and when number of conversions is selected for 16 times)
  The comparison level (upper) to be compared with the A/D-converted value of the same channel is set bits 15 to 0.

42.2.42 A/D Comparison Function Window B Channel Status Register (ADCMPBSR)

(1) S12AD.ADCMPBSR

Address(es): S12AD.ADCMPBSR 0008 90ACh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CMPSTB
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMPSTB	Comparison Window B Flag	0: The comparison condition is not satisfied. 1: The comparison condition is satisfied.	R/(W) *1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

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

The S12AD.ADCMPBSR register is used to store results of comparison of the comparison window B function.

CMPSTB Flag (Comparison Window B Flag)

The CMPSTB flag is a status flag that indicates the result of comparison of the channels for window B comparison conditions (AN000 to AN006).

When the conversion result matches with the comparison condition that is set in the ADCMPBNSR.CMPLB bit upon completion of A/D conversion, this flag becomes 1. When the ADCMPCR.CMPBIE bit is 1, a comparison interrupt request (S12CMPBI) is generated upon the setting of the flag.

1 cannot be written to the CMPSTB flag.

[Setting condition]

- When ADCMPCR.CMPBE bit = 1 and the condition set in the ADCMPBNSR.CMPLB bit is satisfied.

[Clearing condition]

- When 0 is written after reading the state of 1.

(2) S12AD1.ADCMPBSR

Address(es): S12AD1.ADCMPBSR 0008 92ACh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CMPSTB
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMPSTB	Comparison Window B Flag	0: The comparison condition is not satisfied. 1: The comparison condition is satisfied.	R/(W) *1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

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

The S12AD1.ADCMPBSR register is used to store results of comparison of the comparison window B function.

CMPSTB Flag (Comparison Window B Flag)

The CMPSTB flag is a status flag that indicates the result of comparison of the channels for window B comparison conditions (AN100 to AN103).
When the conversion result matches with the comparison condition that is set in the ADCMPBNSR.CMPLB bit upon completion of A/D conversion, this flag becomes 1. When the ADCMPCR.CMPBIE bit is 1, a comparison interrupt request (S12CMPBI1) is generated upon the setting of the flag.
1 cannot be written to the CMPSTB flag.
[Setting condition]

- When ADCMPCR.CMPBE bit = 1 and the condition set in the ADCMPBNSR.CMPLB bit is satisfied.

[Clearing condition]

- When 0 is written after reading the state of 1.

(3) S12AD2.ADCMPBSR

Address(es): S12AD2.ADCMPBSR 0008 94ACh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CMPSTB
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMPSTB	Comparison Window B Flag	0: The comparison condition is not satisfied. 1: The comparison condition is satisfied.	R/(W) *1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

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

The S12AD2.ADCMPBSR register is used to store results of comparison of the comparison window B function.

CMPSTB Flag (Comparison Window B Flag)

The CMPSTB flag is a status flag that indicates the result of comparison of the channels for window B comparison conditions (AN200 to AN211, AN216, AN217, temperature sensor, internal reference voltage).
When the conversion result matches with the comparison condition that is set in the ADCMPBNSR.CMPLB bit upon completion of A/D conversion, this flag becomes 1. When the ADCMPCR.CMPBIE bit is 1, a comparison interrupt request (S12CMPBI2) is generated upon the setting of the flag.
1 cannot be written to the CMPSTB flag.
[Setting condition]

- When ADCMPCR.CMPBE bit = 1 and the condition set in the ADCMPBNSR.CMPLB bit is satisfied.

[Clearing condition]

- When 0 is written after reading the state of 1.

42.2.43 A/D Programmable Gain Amplifier Control Register (ADPGACR)

(1) S12AD.ADPGACR

Address(es): S12AD.ADPGACR 0008 91A0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	P002CR[3:0]				P001CR[3:0]				P000CR[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	P000CR[3:0]	P000 Amplifier Control	These bits control the PGA (P000) that is connected to AN000 and the destinations of the analog signal. For details of the settings, refer to Table 42.19.	R/W
b7 to b4	P001CR[3:0]	P001 Amplifier Control	These bits control the PGA (P001) that is connected to AN001 and the destinations of the analog signal. For details of the settings, refer to Table 42.19.	R/W
b11 to b8	P002CR[3:0]	P002 Amplifier Control	These bits control the PGA (P002) that is connected to AN002 and the destinations of the analog signal. For details of the settings, refer to Table 42.19.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

(2) S12AD1.ADPGACR

Address(es): S12AD1.ADPGACR 0008 93A0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	P102CR[3:0]				P101CR[3:0]				P100CR[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	P100CR[3:0]	P100 Amplifier Control	These bits control the PGA (P100) that is connected to AN100 and the destinations of the analog signal. For details of the settings, refer to Table 42.19.	R/W
b7 to b4	P101CR[3:0]	P101 Amplifier Control	These bits control the PGA (P101) that is connected to AN101 and the destinations of the analog signal. For details of the settings, refer to Table 42.19.	R/W
b11 to b8	P102CR[3:0]	P102 Amplifier Control	These bits control the PGA (P102) that is connected to AN102 and the destinations of the analog signal. For details of the settings, refer to Table 42.19.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

PxCR[3:0] Bits (Px Amplifier Control) (x = 000 to 002, 100 to 102)

These bits set the paths for analog signals which were input to pins AN000 to AN002, and pins AN100 to AN102. Refer to Table 42.19 for the relationships among the settings, input signals to the A/D converter, and input signals to comparator C.

Table 42.19 Settings of the PxCR[3:0] bits and paths for analog signals (x = 000 to 002, 100 to 102; m = 0 to 5)

PxCR[3:0] Bits	PGA	Input to the A/D converter	Input to comparator C	
			CMPCm0	CMPCm1
0000b (initial value)	Disabled	—	—	—
0001b	Disabled	ANx	—	—
1000b	Disabled	—	ANx	—
1001b	Disabled	ANx	ANx	—
1100b	Enabled	—	ANx	Px output
1101b	Enabled	ANx	ANx	Px output
1110b	Enabled	Px output	ANx	Px output

Note: Settings other than those listed above are prohibited.

42.2.44 A/D Programmable Gain Amplifier Gain Setting Register 0 (ADPGAGS0)

(1) S12AD.ADPGAGS0

Address(es): S12AD.ADPGAGS0 0008 91A2h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	P002GAIN[3:0]				P001GAIN[3:0]				P000GAIN[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	P000GAIN[3:0]	P000 Amplifier Gain Setting	0 0 0 0: × 2.000 0 0 0 1: × 2.500 0 0 1 1: × 3.077 0 1 0 1: × 3.636 0 1 1 0: × 4.000 0 1 1 1: × 4.444 1 0 0 0: × 5.000 1 0 1 0: × 6.667 1 0 1 1: × 8.000 1 1 0 0: × 10.000 1 1 0 1: × 13.333 1 1 1 0: × 20.000 Settings other than above are prohibited.	R/W
b7 to b4	P001GAIN[3:0]	P001 Amplifier Gain Setting		R/W
b11 to b8	P002GAIN[3:0]	P002 Amplifier Gain Setting		R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

(2) S12AD1.ADPGAGS0

Address(es): S12AD1.ADPGAGS0 0008 93A2h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	P102GAIN[3:0]				P101GAIN[3:0]				P100GAIN[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	P100GAIN[3:0]	P100 Amplifier Gain Setting	0 0 0 0: × 2.000 0 0 0 1: × 2.500 0 0 1 1: × 3.077 0 1 0 1: × 3.636 0 1 1 0: × 4.000 0 1 1 1: × 4.444 1 0 0 0: × 5.000 1 0 1 0: × 6.667 1 0 1 1: × 8.000 1 1 0 0: × 10.000 1 1 0 1: × 13.333 1 1 1 0: × 20.000 Settings other than above are prohibited.	R/W
b7 to b4	P101GAIN[3:0]	P101 Amplifier Gain Setting		R/W
b11 to b8	P102GAIN[3:0]	P102 Amplifier Gain Setting		R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

PxGAIN[3:0] Bits (Px Amplifier Gain Setting) (x = 000 to 002, 100 to 102)

The PxGAIN[3:0] bits are used to specify the gain of each programmable gain amplifier Px.

42.2.45 A/D Internal Reference Voltage Monitoring Circuit Enable Register (ADVMONCR)

Address(es): S12AD2.ADVMONCR 0008 95E2h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	VDE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	VDE	Voltage Monitoring Circuit Enable	0: Disables internal reference voltage monitoring circuit operation. 1: Enables internal reference voltage monitoring circuit operation.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADVMONCR register is used to enable the voltage monitoring circuit operation for the internal reference voltage.
Set the register while the ADCSR.ADST bit is 0.

42.2.46 A/D Internal Reference Voltage Monitoring Circuit Output Enable Register (ADVMONO)

Address(es): S12AD2.ADVMONO 0008 95E4h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	VDO
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	VDO	Voltage Monitoring Circuit Voltage Output Enable	0: Disables voltage output of internal reference voltage monitoring circuit. 1: Enables voltage output of internal reference voltage monitoring circuit.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADVMONO register is used to enable the voltage output of the voltage monitoring circuit for the internal reference voltage.
Set the register while the ADCSR.ADST bit is 0.

42.3 Operation

42.3.1 A/D Converter Initialization Flowchart

Initialize the A/D converter following the example of flowchart shown in Figure 42.8.

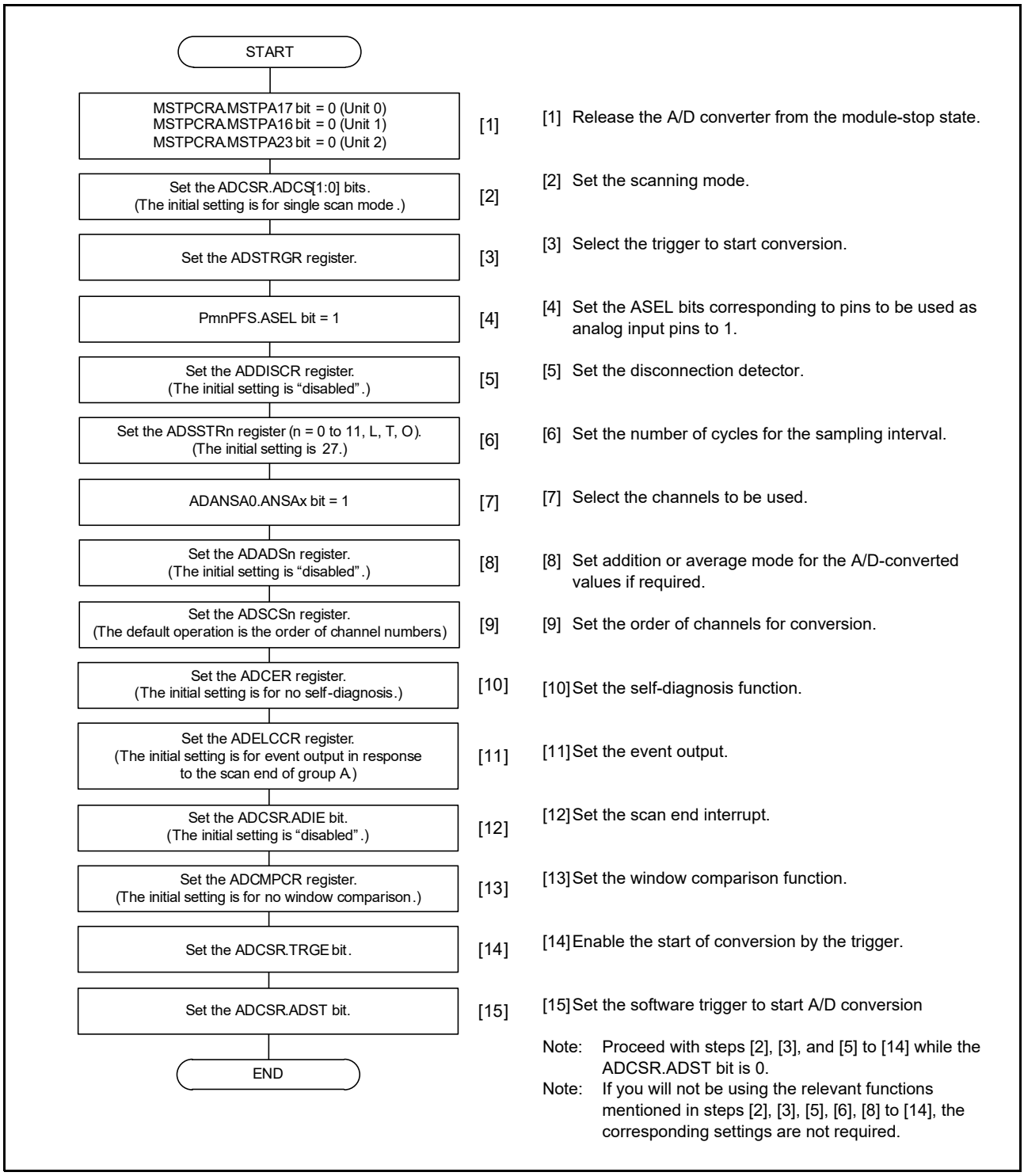


Figure 42.8 A/D Converter Initialization Flowchart

42.3.2 Scanning Operation

In scanning, A/D conversion is performed sequentially on the analog inputs of the specified channels.
A scan conversion is performed in three operating modes: single scan mode, continuous scan mode, and group scan mode. In single scan mode, one or more specified channels are scanned once. In continuous scan mode, one or more specified channels are scanned repeatedly until the ADCSR.ADST bit is cleared to 0 from 1 by software. In group scan mode, the selected channels of groups A, B, and C are scanned once after starting to be scanned according to the respective synchronous trigger.
In single scan mode and continuous scan mode, A/D conversion is performed for the analog channels selected in the ADANSA0 and ADANSA1 registers according to the order of conversion specified in the ADSCSn register.
In group scan mode, A/D conversion is performed for analog channels of groups A, B, and C selected by the ADANSA0 and ADANSA1 registers, the ADANSB0 and ADANSB1 registers, and the ADANSC0 and ADANSC1 registers, respectively, according to the order of conversion specified in the ADSCSn register.
When self-diagnosis is selected, it is executed once at the beginning of each scan and one of the three voltages internally generated in the 12-bit A/D converter is converted.
The temperature sensor output and the internal reference voltage can be selected; however, they cannot be selected in combination with an analog input of the channels in the same group.
Conversion of the temperature sensor output takes priority when the temperature sensor output and internal reference voltage are selected simultaneously.
When converting the temperature sensor output or the internal reference voltage, an auto-discharging is automatically performed.
Double trigger mode is to be used with single scan mode or group scan mode. With double trigger mode being enabled, A/D conversion data of a channel selected by the ADCSR.DBLANS[4:0] bits is duplicated only if the conversion is started by the synchronous trigger selected by the ADSTRGR.TRSA[6:0] bits. In group scan mode, the double trigger function can be used only for group A.
Extended double trigger mode indicates a state when the following synchronous trigger (two synchronous trigger sources enabled) is selected by the TRSA[6:0] bits in the A/D conversion start trigger select register (ADSTRGR) in double trigger mode.

- TRG4AN or TRG4BN (the ADSTRGR.TRSA[6:0] bits are set to 0001011b)
- TRG7AN or TRG7BN (the ADSTRGR.TRSA[6:0] bits are set to 0001111b)
- TRGA0N or TRG0N (the ADSTRGR.TRSA[6:0] bits are set to 0011001b)
- TRGA9N or TRG9N (the ADSTRGR.TRSA[6:0] bits are set to 0011010b)
- TRGA0N or TRGA9N (the ADSTRGR.TRSA[6:0] bits are set to 0011011b)
- TRG0N or TRG9N (the ADSTRGR.TRSA[6:0] bits are set to 0011100b)
- GTADTRA0N or GTADTRB0N (the ADSTRGR.TRSA[6:0] bits are set to 1001000b)
- GTADTRA1N or GTADTRB1N (the ADSTRGR.TRSA[6:0] bits are set to 1001001b)
- GTADTRA2N or GTADTRB2N (the ADSTRGR.TRSA[6:0] bits are set to 1001010b)
- GTADTRA3N or GTADTRB3N (the ADSTRGR.TRSA[6:0] bits are set to 1001011b)
- GTADTRA4N or GTADTRB4N (the ADSTRGR.TRSA[6:0] bits are set to 1010100b)
- GTADTRA5N or GTADTRB5N (the ADSTRGR.TRSA[6:0] bits are set to 1010101b)
- GTADTRA6N or GTADTRB6N (the ADSTRGR.TRSA[6:0] bits are set to 1010110b)
- GTADTRA7N or GTADTRB7N (the ADSTRGR.TRSA[6:0] bits are set to 1010111b)
- ELCTRG00N or ELCTRG01N, ELCTRG10N or ELCTRG11N, or ELCTRG20N or ELCTRG21N (the ADSTRGR.TRSA[6:0] bits are set to 0111010b)

In extended double trigger mode, in addition to normal operations in double trigger mode, A/D conversion data is stored in A/D data duplication register A (ADDBLDRA) or A/D data duplication register B (ADDBLDRB) depending on the trigger type. If two types of triggers have occurred simultaneously in this mode, A/D conversion data is not sorted by the trigger sources and is stored in data duplication register B (ADDBLDRB).

Note that if a new trigger is input during A/D conversion caused by another trigger, the new trigger is ignored.
When any of AN000 to AN002 or AN100 to AN102 channels is set as a channel-dedicated sample-and-hold circuit by the ADSHCR.SHANS[2:0] bits, the target analog input is sampled and held before the first A/D conversion of each scan.

42.3.3 Single Scan Mode

42.3.3.1 Basic Operation (Without Channel-Dedicated Sample-and-Hold Circuits)

In basic operation of single scan mode, A/D conversion is performed once on the analog input of the specified channels as below.

(1) When the ADCSR.ADST bit is set to 1 (starting A/D conversion) by software, or synchronous or asynchronous trigger input, A/D conversion is performed for the analog channels selected in the ADANSA0 and ADANSA1 registers according to the order of conversion specified in the ADSCSn register.
(Specified conversion order in Figure 42.9: AN000 → AN001 → AN002)

(2) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).

(3) When A/D conversion of all the selected channels is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (scan end interrupt is enabled).

(4) The ADCSR.ADST bit remains 1 (starting A/D conversion) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels is completed. Then the 12-bit A/D converter enters a wait state.

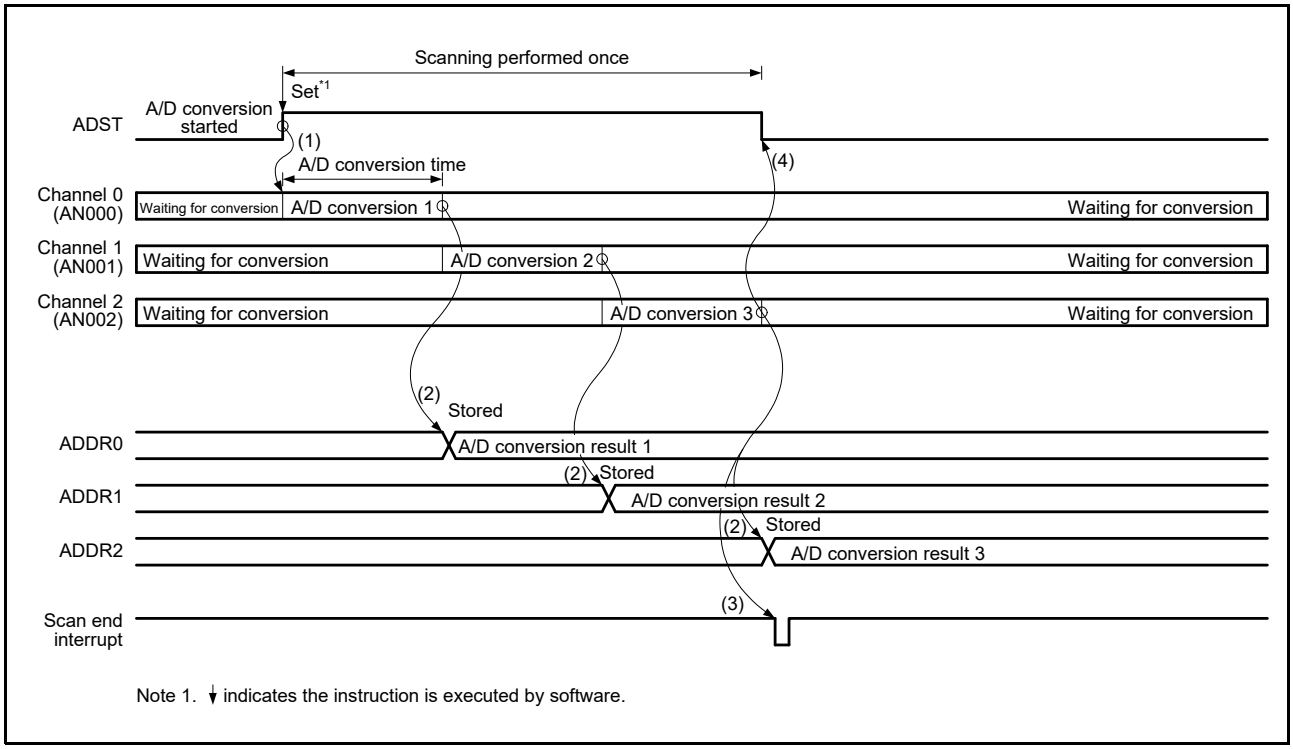


Figure 42.9 Example of Operation in Single Scan Mode (Basic Operation: AN000, AN001, AN002 Selected)

42.3.3.2 Basic Operation (with Channel-Dedicated Sample-and-Hold Circuits, with Constant Sampling Disabled)

When a channel-dedicated sample-and-hold circuit is used, sample-and-hold operations are performed first, and this is followed by A/D conversion once of the analog inputs on all selected channels. The ADSHCR.SHANS[2:0] bits are used to select the channels for which the channel-dedicated sample-and-hold circuits are to be used.

(1) Analog input sampling of all channels for which the channel-dedicated sample-and-hold circuits are to be used is started when the ADCSR.ADST bit is set to 1 (starting A/D conversion) by software, or synchronous or asynchronous trigger input.
(2) After sample-and-hold operation, A/D conversion is performed for the analog channels selected in the ADANSA0 and ADANSA1 registers according to the order of conversion specified in the ADSCSn register. To use the channel-dedicated sample-and-hold circuit, set the corresponding channel in any of the ADSCS0, ADSCS1, or ADSCS2 register.
(3) Each time A/D conversion of a single channel is completed, the result of A/D conversion is stored in the corresponding A/D data register (ADDRy).
(4) When A/D conversion of all the selected channels is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (scan end interrupt is enabled).
(5) The ADCSR.ADST bit remains 1 (starting A/D conversion) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels is completed. Then the 12-bit A/D converter enters a wait state.

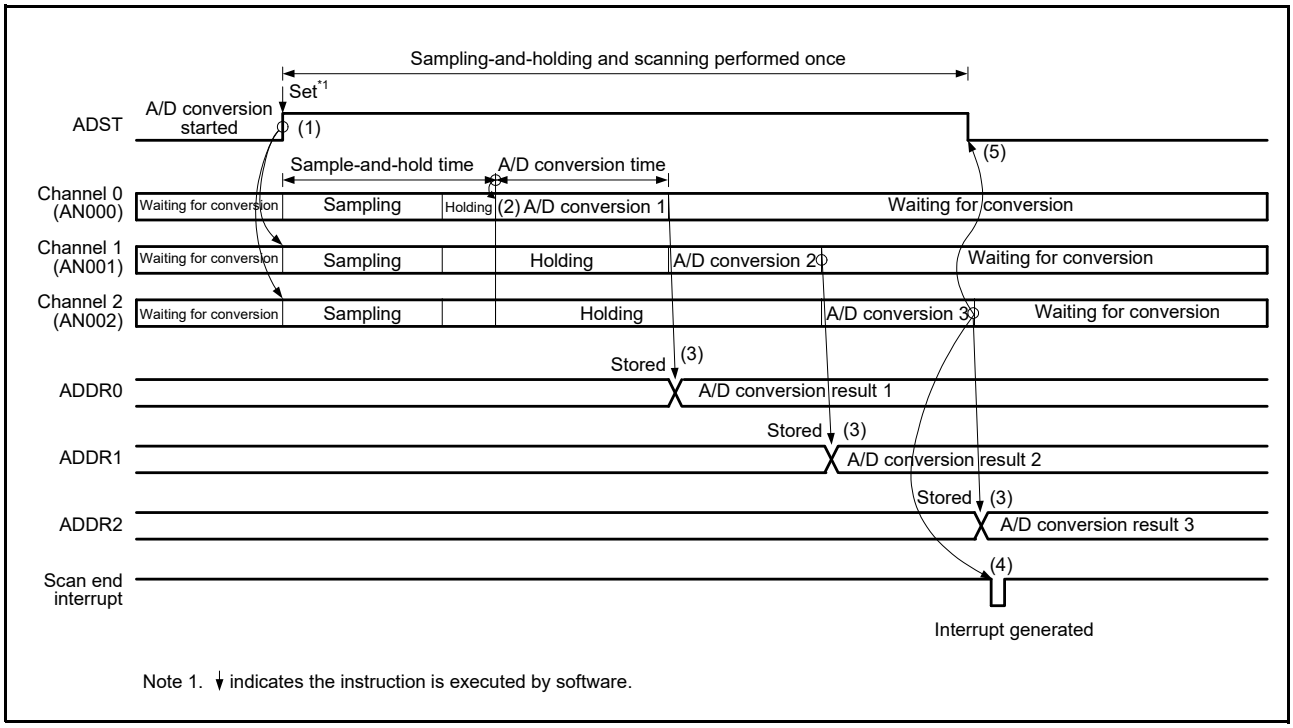


Figure 42.10 Example of Operation in Single Scan Mode (Channel-Dedicated Sample-and-Hold Circuits Used; AN000, AN001, AN002 Selected, with Constant Sampling Disabled)

42.3.3.3 Basic Operation (with Channel-Dedicated Sample-and-Hold Circuits, with Constant Sampling Enabled)

When constant sampling is enabled and the channel sample-and-hold circuit is used, a single A/D conversion is performed for analog input of all channels selected after sampling and holding is executed.
The channels for channel sample-and-hold circuit can be selected in the ADSHCR.SHANS[2:0] bits.

(1) When the ADSHMSR.SHMD bit is set to 1, the sample-and-hold circuit selected in the ADSHCR.SHANS[2:0] bits starts constant sampling.
(2) Analog input sampling of all channels for which the channel-dedicated sample-and-hold circuits are to be used is started when the ADCSR.ADST bit is set to 1 (starting A/D conversion) by software, or synchronous or asynchronous trigger input. After setting the ADSHMSR.SHMD bit to 1 and 750 ns minimum are elapsed, set the ADCSR.ADST bit to 1.
(3) When the stabilizing time for the sample-and-hold circuit is elapsed, A/D conversion is performed for the analog channels selected in the ADANSA0 and ADANSA1 registers according to the order of conversion specified in the ADSCSn register. To use the channel-dedicated sample-and-hold circuit, set the corresponding channel in any of the ADSCS0, ADSCS1, or ADSCS2 register.
(4) Each time A/D conversion of a single channels is completed, the A/D-converted value is stored in the corresponding A/D data register (ADDRy), the sample-and-hold circuit resumes constant sampling.
(5) When A/D conversion of all the selected channels is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (scan end interrupt is enabled).
(6) The ADCSR.ADST bit remains 1 (starting A/D conversion) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels is completed. Then the 12-bit A/D converter enters a wait state. When a single scan is performed in sequence, set the time of constant sampling of the sample-and-hold circuit to 400 ns or longer (when allowable impedance of the source of the enabling signal is 1 kΩ).
(7) When the ADSHMSR.SHMD bit is cleared to 0, the sample-and-hold circuit stops.

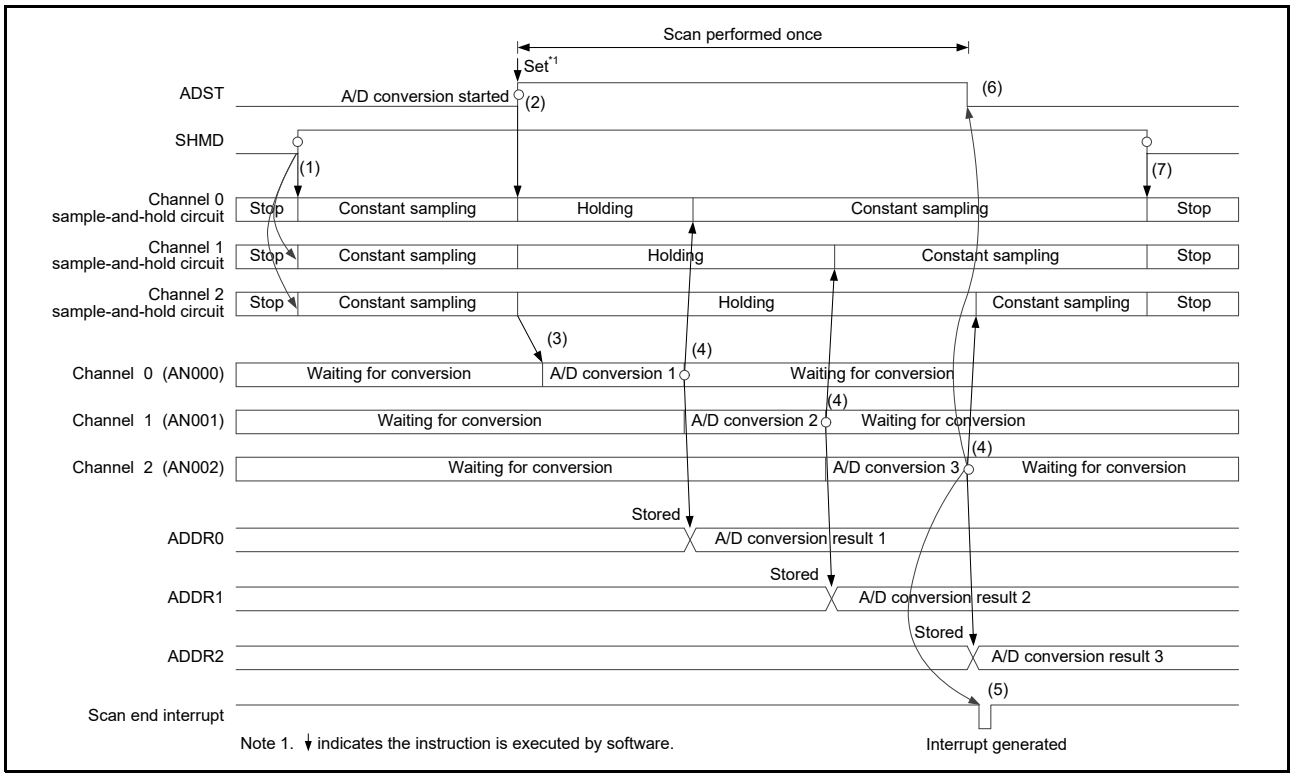


Figure 42.11 Example of Operation in Single Scan Mode (Channel-Dedicated Sample-and-Hold Circuits Used; AN000 to AN002 Selected, with Constant Sampling Enabled)

42.3.3.4 Channel Selection and Self-Diagnosis (Without Channel-Dedicated Sample-and-Hold Circuits)

When channels and self-diagnosis are selected, A/D conversion is performed once for the reference voltage supplied to the 12-bit A/D converter as below. After that, A/D conversion is performed only once on the analog input of the selected channels.

(1) A/D conversion for self-diagnosis is started when the ADCSR.ADST bit is set to 1 (starting A/D conversion) by software, or synchronous or asynchronous trigger input.
(2) When A/D conversion for self-diagnosis is completed, A/D conversion result is stored into the A/D self-diagnosis data register (ADRD), A/D conversion is performed for the analog channels selected in the ADANSA0 and ADANSA1 registers according to the order of conversion specified in the ADSCSn register.
(3) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(4) When A/D conversion of all the selected channels is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (scan end interrupt is enabled).
(5) The ADST bit remains 1 (starting A/D conversion) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels is completed. Then the 12-bit A/D converter enters a wait state.

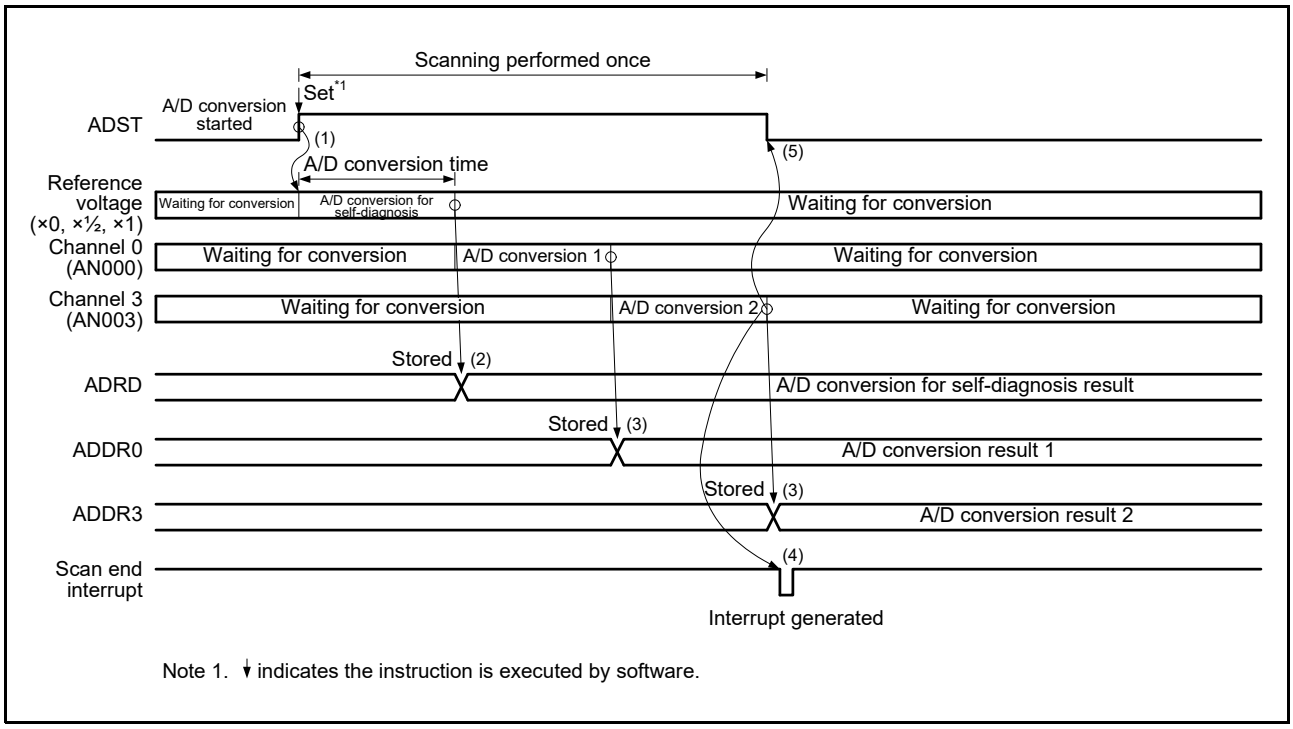


Figure 42.12 Example of Operation in Single Scan Mode (Basic Operation: AN000, AN003 Selected + Self-Diagnosis)

42.3.3.5 Channel Selection and Self-Diagnosis (with Channel-Dedicated Sample-and-Hold Circuits, with Constant Sampling Disabled)

When channels and self-diagnosis are selected and a channel-dedicated sample-and-hold circuit is used, sample-and-hold operations are performed first, and A/D conversion is performed once for the reference voltage supplied to the 12-bit A/D converter as below. After that, A/D conversion is performed only once on the analog input of the selected channels. The self-diagnosis cannot be used with the temperature sensor output or with the internal reference voltage.

(1) Analog input sampling of all channels for which the channel-dedicated sample-and-hold circuits are to be used is started when the ADCSR.ADST bit is set to 1 (starting A/D conversion) by software, or synchronous or asynchronous trigger input.
(2) A/D conversion for self-diagnosis is started after completion of sampling and holding.
(3) When A/D conversion for self-diagnosis is completed, A/D conversion result is stored into the A/D self-diagnosis data register (ADRD), A/D conversion is performed for the analog channels selected in the ADANSA0 and ADANSA1 registers according to the order of conversion specified in the ADSCSn register. To use the channel-dedicated sample-and-hold circuit, set the corresponding channel in any of the ADSCS0, ADSCS1, or ADSCS2 register.
(4) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(5) When A/D conversion of all the selected channels is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (scan end interrupt is enabled).
(6) The ADCSR.ADST bit remains 1 (starting A/D conversion) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels is completed. Then the 12-bit A/D converter enters a wait state.

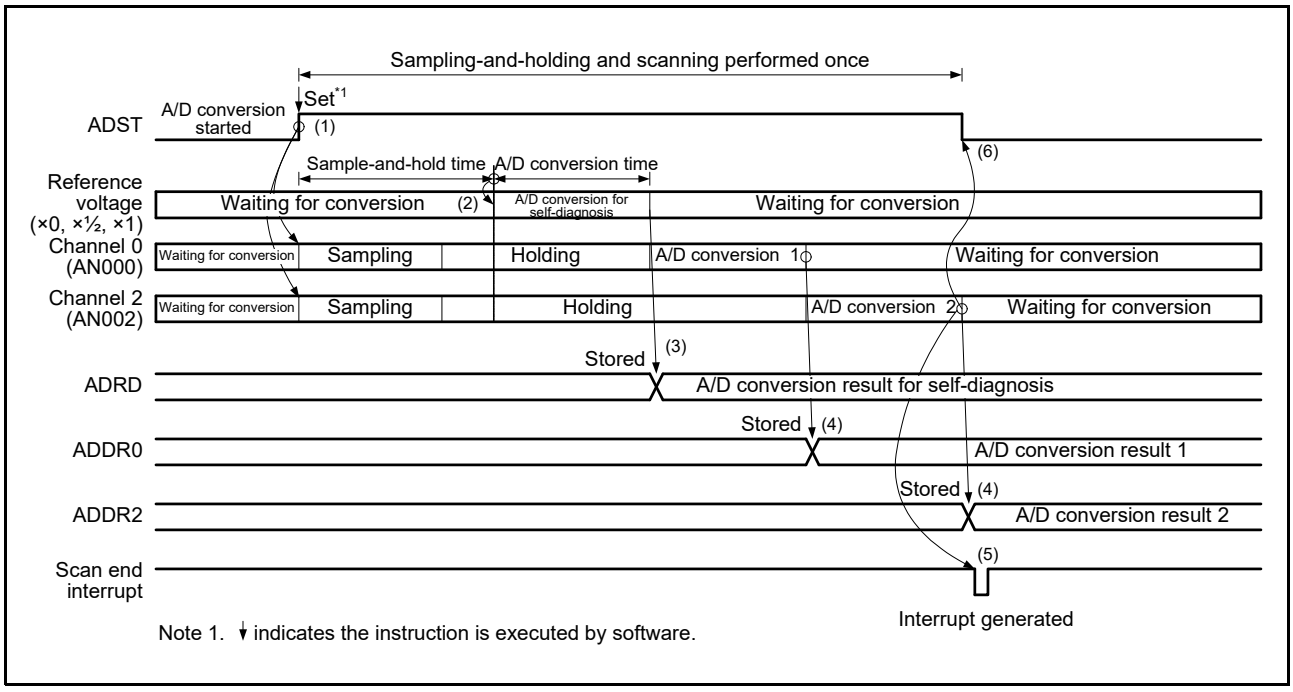


Figure 42.13 Example of Operation in Single Scan Mode (Channel-Dedicated Sample-and-Hold Circuits Used: AN000, AN002 Selected + Self-Diagnosis, with Constant Sampling Disabled)

42.3.3.6 Channel Selection and Self-Diagnosis (with Channel-Dedicated Sample-and-Hold Circuits, with Constant Sampling Enabled)

When self-diagnosis is selected with channels and a channel-dedicated sample-and-hold circuit is used with constant sampling enabled, sample-and-hold operations are performed first, and A/D conversion is performed on the reference voltage supplied to the 12-bit A/D converter as described below. After that, A/D conversion is performed only once on the analog input of the selected channels. The self-diagnosis cannot be used with the temperature sensor output or with the internal reference voltage.

(1) When the ADSHMSR.SHMD bit is set to 1, the sample-and-hold circuit selected in the ADSHCR.SHANS[2:0] bits starts constant sampling.
(2) Analog input sampling of all channels for which the channel-dedicated sample-and-hold circuits are to be used is started when the ADCSR.ADST bit is set to 1 (starting A/D conversion) by software, or synchronous or asynchronous trigger input. After setting the ADSHMSR.SHMD bit to 1 and 750 ns minimum are elapsed, set the ADCSR.ADST bit to 1.
(3) When the stabilizing time for the sample-and-hold circuit is elapsed, A/D conversion by self-diagnosis is started.
(4) When A/D conversion for self-diagnosis is completed, A/D conversion result is stored into the A/D self-diagnosis data register (ADRD), A/D conversion is performed for the analog channels selected in the ADANSA0 and ADANSA1 registers according to the order of conversion specified in the ADSCSn register. To use the channel-dedicated sample-and-hold circuit, set the corresponding channel in any of the ADSCS0, ADSCS1, or ADSCS2 register.
(5) Each time A/D conversion of a single channel is completed, the A/D-converted value is stored in the corresponding A/D data register (ADDRy), the sample-and-hold circuit resumes constant sampling.
(6) When A/D conversion of all the selected channels is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (scan end interrupt is enabled).
(7) The ADCSR.ADST bit remains 1 (starting A/D conversion) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels is completed. Then the 12-bit A/D converter enters a wait state. When a single scan is performed in sequence, set the time of constant sampling of the sample-and-hold circuit to 400 ns or longer (when allowable impedance of the source of the enabling signal is 1 kΩ).
(8) When the ADSHMSR.SHMD bit is cleared to 0, the sample-and-hold circuit stops.

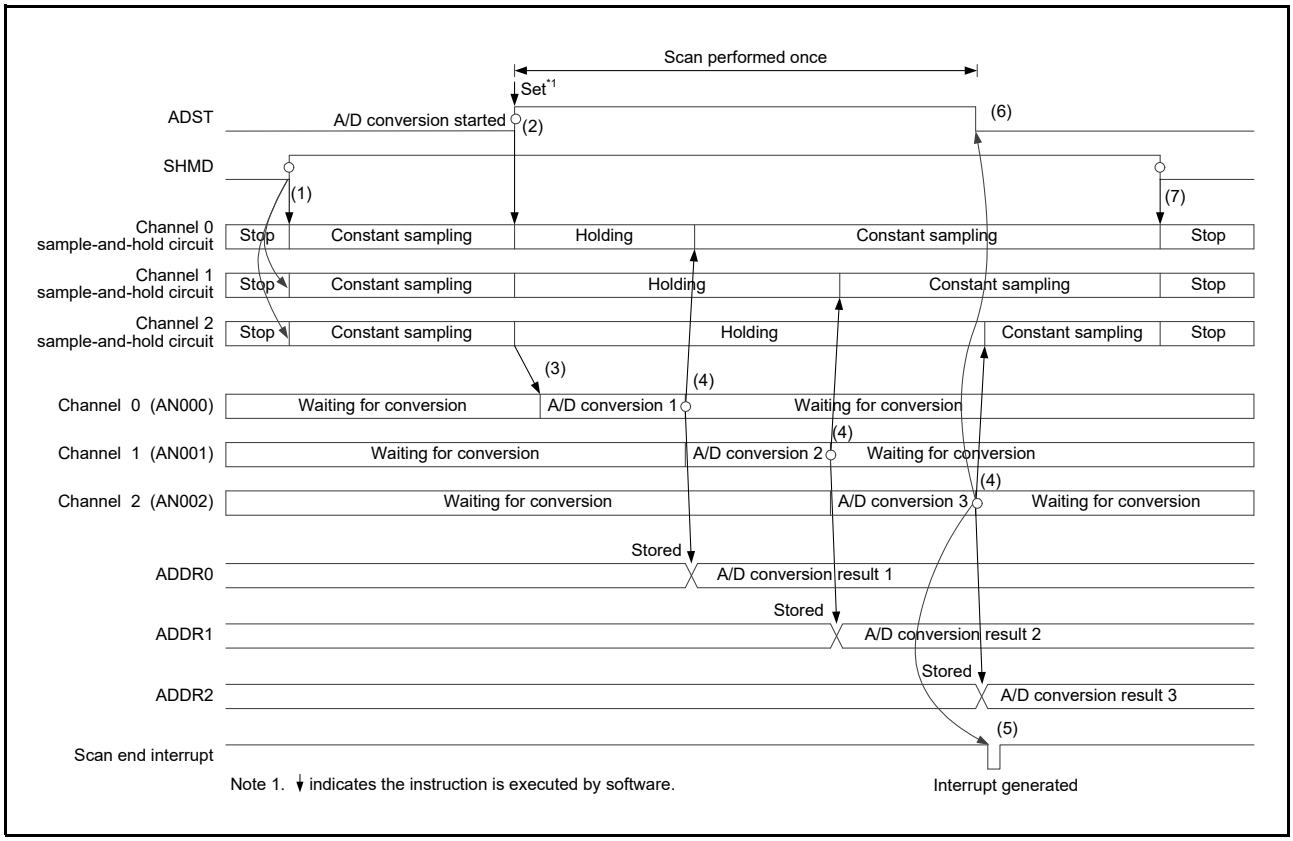


Figure 42.14 Example of Operation in Single Scan Mode (Channel-Dedicated Sample-and-Hold Circuits Used; AN000, AN002 Selected + Self-Diagnosis, with Constant Sampling Enabled)

42.3.3.7 A/D Conversion with Temperature Sensor Output/Internal Reference Voltage Selected

When temperature sensor output or internal reference voltage is selected with channels, A/D conversion is performed only once for the temperature sensor output or for internal reference voltage as described below.
When the temperature sensor output and internal reference voltage are both selected, A/D conversion is performed for the temperature sensor output first and then for internal reference voltage.
Selecting the temperature sensor output and internal reference voltage simultaneously for analog input is prohibited.

(1) When the ADCSR.ADST bit becomes 1 (starting A/D conversion) by software, or synchronous or asynchronous trigger input, auto-discharging and A/D conversion is performed in this order for the temperature sensor output.
(2) When A/D conversion of the temperature sensor output is completed, the A/D-converted value is stored in the corresponding A/D temperature sensor data register (ADTSDR), and then, auto-discharging and A/D conversion is performed in this order for the internal reference voltage.
(3) After completion of A/D conversion of the internal reference voltage, the A/D-converted value is stored in the corresponding A/D internal reference voltage data register (ADOCDR), a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (scan end interrupt is enabled).
(4) The ADCSR.ADST bit remains 1 (starting A/D conversion) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels is completed. Then the 12-bit A/D converter enters a wait state.

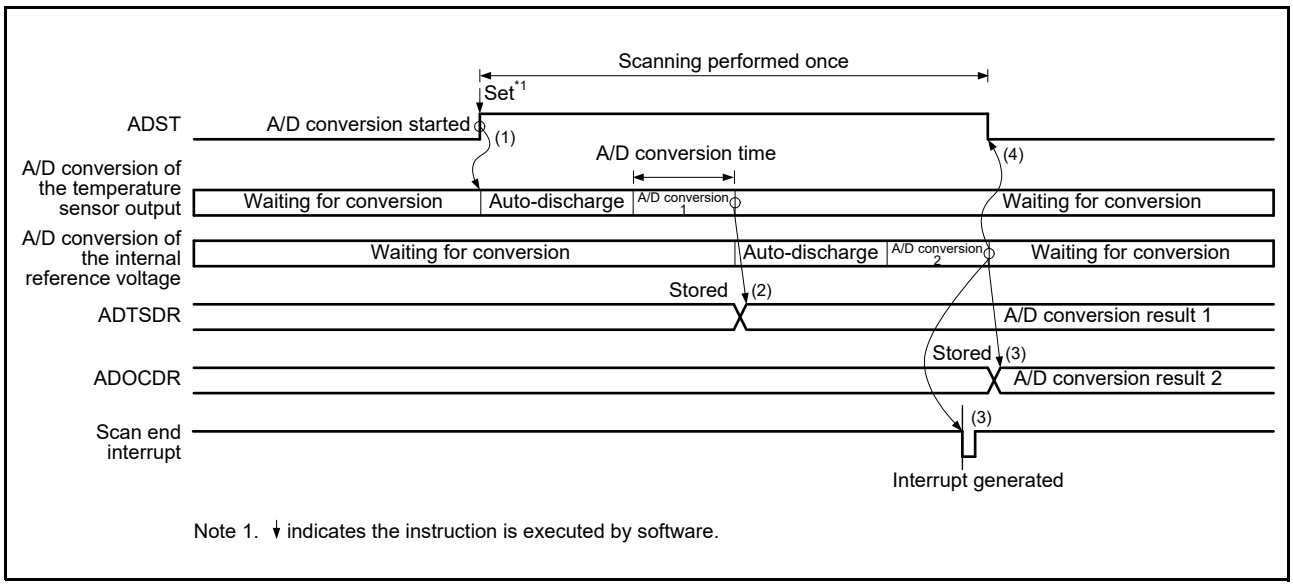


Figure 42.15 Example of Operation in Single Scan Mode (Basic Operation: Temperature Sensor Output and Internal Reference Voltage Selected)

42.3.3.8 A/D Conversion in Double Trigger Mode

In single scan mode with double trigger mode, single scan operation started by synchronous trigger is performed twice as below.

Deselect self-diagnosis and set the temperature sensor A/D conversion select bit (ADEXICR.TSSA, ADEXICR.TSSB, ADGCEXCR.TSSC) and internal reference voltage A/D conversion select bit (ADEXICR.OCSA, ADEXICR.OCSB, ADGCEXCR.OCSC) to 0.

Duplication of A/D conversion data is enabled by setting the channel numbers to be duplicated to the ADCSR.DBLANS[4:0] bits and setting the ADCSR.DBLE bit to 1. When the ADCSR.DBLE bit is set to 1, channel selection using the ADANSA0 and ADANSA1 registers is invalid. In double trigger mode, synchronous triggers should be selected using the ADSTRGR.TRSA[6:0] bits, the ADCSR.EXTRG bit should be set to 0, and the ADCSR.TRGE bit should be set to 1. Software trigger should not be used.

(1) When the ADCSR.ADST bit becomes 1 (starting A/D conversion) by synchronous trigger input, A/D conversion is started on the single channel selected by the ADCSR.DBLANS[4:0] bits.
(2) When A/D conversion is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(3) The ADCSR.ADST bit is automatically cleared to 0 and the 12-bit A/D converter enters a wait state. Here, a scan end interrupt request is not generated irrespective of the ADCSR.ADIE bit setting (scan end interrupt is enabled).
(4) When the ADCSR.ADST bit is set to 1 (starting A/D conversion) by the second trigger input, A/D conversion is started on the single channel selected by the ADCSR.DBLANS[4:0] bits.
(5) When A/D conversion is completed, the A/D conversion result is stored into the A/D data duplication register (ADDBLDR).
(6) A scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (scan end interrupt is enabled).
(7) The ADCSR.ADST bit remains 1 (starting A/D conversion) during A/D conversion, and is automatically cleared to 0 when A/D conversion is completed. Then the 12-bit A/D converter enters a wait state.

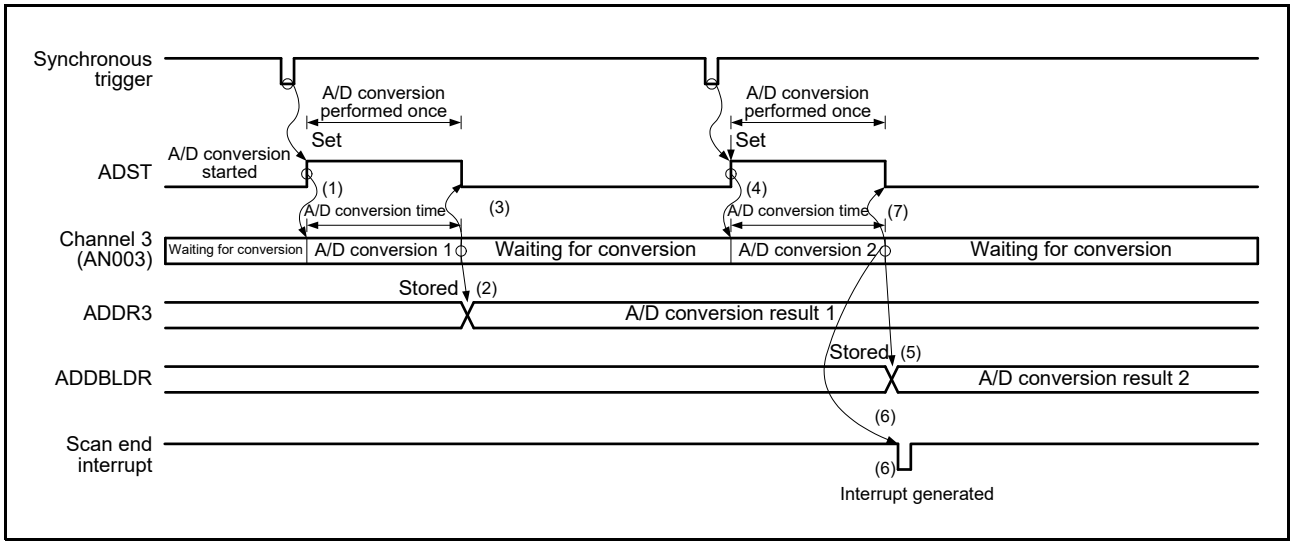


Figure 42.16 Example of Operation in Single Scan Mode (Double Trigger Mode Selected; AN003 Duplicated)

42.3.3.9 A/D Conversion in Extended Double Trigger Mode

When selecting the double-trigger mode in single-scanning mode, with TRG4AN or TRG4BN, TRG7AN or TRG7BN, TRGA0N or TRG0N, TRGA9N or TRG9N, TRGA0N or TRGA9N, TRG0N or TRG9N, GTADTRA0N or GTADTRB0N, GTADTRA1N or GTADTRB1N, GTADTRA2N or GTADTRB2N, GTADTRA3N or GTADTRB3N, GTADTRA4N or GTADTRB4N, GTADTRA5N or GTADTRB5N, GTADTRA6N or GTADTRB6N, GTADTRA7N or GTADTRB7N, or ELCTRG00N or ELCTRG01N/ELCTRG10N or ELCTRG11N/ELCTRG20N or ELCTRG21N selected in the TRSA[6:0] bits of the A/D conversion start trigger select register (ADSTRGR), proceed with single scanning mode twice as follows.
Deselect self-diagnosis and set the temperature sensor A/D conversion select bit (ADEXICR.TSSA, ADEXICR.TSSB, ADGCEXCR.TSSC) and internal reference voltage A/D conversion select bit (ADEXICR.OCSA, ADEXICR.OCSB, ADGCEXCR.OCSC) to 0.
Duplication of A/D conversion data is enabled by setting the channel numbers to be duplicated to the ADCSR.DBLANS[4:0] bits and setting the ADCSR.DBLE bit to 1. When the ADCSR.DBLE bit is set to 1, channel selection using the ADANSA0 and ADANSA1 registers is invalid. In extended double trigger mode, the ADCSR.EXTRG bit should be set to 0, and the ADCSR.TRGE bit should be set to 1. Software trigger should not be used.

(1) When the ADCSR.ADST bit is set to 1 (starting A/D conversion) by TRG4AN input, A/D conversion is started on the single channel selected by the ADCSR.DBLANS[4:0] bits.
(2) When A/D conversion is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy) and A/D data duplication register A (ADDBLDRA).
(3) The ADCSR.ADST bit is automatically cleared to 0 and the 12-bit A/D converter enters a wait state. Here, a scan end interrupt request is not generated irrespective of the ADCSR.ADIE bit setting (scan end interrupt is enabled).
(4) When the ADCSR.ADST bit is set to 1 (starting A/D conversion) by TRG4BN input, A/D conversion is started on the single channel selected by the ADCSR.DBLANS[4:0] bits.
(5) When A/D conversion is completed, the A/D conversion result is stored into A/D data duplication register (ADDBLDR) and A/D data duplication register B (ADDBLDRB).
(6) A scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (scan end interrupt is enabled).
(7) The ADCSR.ADST bit remains 1 (starting A/D conversion) during A/D conversion, and is automatically cleared to 0 when A/D conversion is completed. Then the 12-bit A/D converter enters a wait state.

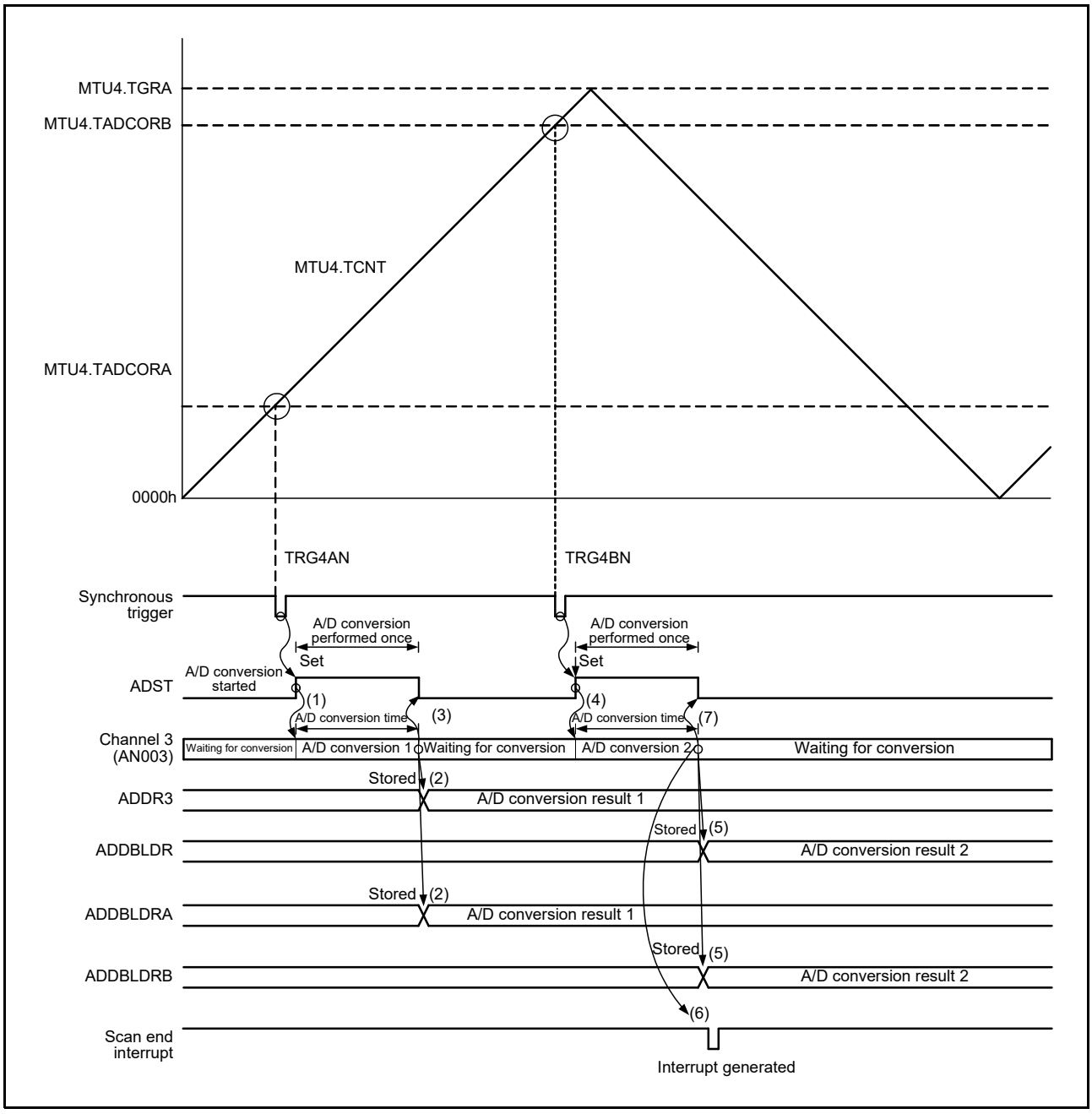


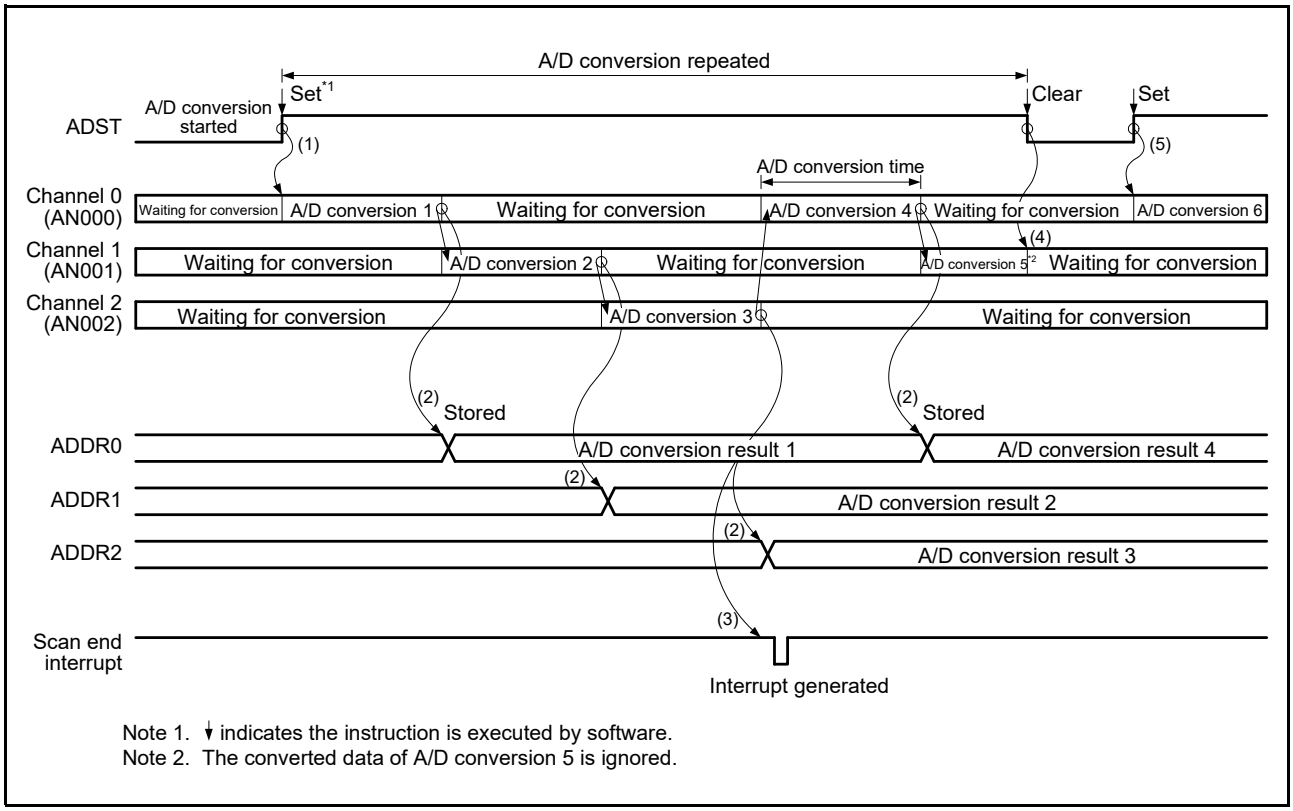
Figure 42.17 Example of Extended Operation in Double Trigger Mode (1)
(Duplication Selected for AN003, TRG4AN or TRG4BN Selected, First Trigger is TRG4AN)

42.3.4 Continuous Scan Mode

42.3.4.1 Basic Operation (Without Channel-Dedicated Sample and-Hold Circuits)

In basic operation of continuous scan mode, A/D conversion is performed repeatedly on the analog input of the specified channels as below.
In continuous scan mode, specify A/D conversion select bits for the temperature sensor and for the internal reference voltage to deselect respectively.

(1) When the ADCSR.ADST bit is set to 1 (starting A/D conversion) by software, or synchronous or asynchronous trigger input, A/D conversion is performed for the analog channels selected in the ADANSA0 and ADANSA1 registers according to the order of conversion specified in the ADSCSn register. (Specified conversion order in Figure 42.18: AN000 → AN001 → AN002)
(2) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(3) When A/D conversion of all the selected channels is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (scan end interrupt is enabled).
The 12-bit A/D converter sequentially starts A/D conversion for the analog channels selected in the ADANSA0 and ADANSA1 registers according to the order of conversion specified in the ADSCSn register.
(4) The ADCSR.ADST bit is not automatically cleared and steps (2) and (3) are repeated as long as the bit remains 1 (starting A/D conversion). When the ADCSR.ADST bit is set to 0 (stopping A/D conversion), A/D conversion stops and the 12-bit A/D converter enters a wait state.
(5) When the ADCSR.ADST bit is later set to 1 (starting A/D conversion), A/D conversion starts again for the analog channels selected in the ADANSA0 and ADANSA1 registers according to the order of conversion specified in the ADSCSn register.



Note 1. ↓ indicates the instruction is executed by software.
Note 2. The converted data of A/D conversion 5 is ignored.

Figure 42.18 Example of Operation in Continuous Scan Mode (Basic Operation: AN000 to AN002 Selected)

42.3.4.2 Basic Operation (with Channel-Dedicated Sample-and-Hold Circuits, with Constant Sampling Disabled)

When a channel-dedicated sample-and-hold circuit is used, sample-and-hold operations are performed first and A/D conversion is repeated for analog input of all selected channels as described below. The ADSHCR.SHANS[2:0] bits are used to select the channels for which the channel-dedicated sample-and-hold circuits are to be used.
In continuous scan mode, specify A/D conversion select bits for the temperature sensor and for the internal reference voltage to deselect respectively.

(1) Analog input sampling of all channels for which the channel-dedicated sample-and-hold circuits are to be used is started when the ADCSR.ADST bit is set to 1 (starting A/D conversion) by software, synchronous trigger input, or asynchronous trigger input.
(2) After sample-and-hold operation, A/D conversion is performed for the analog channels selected in the ADANSA0 and ADANSA1 registers according to the order of conversion specified in the ADSCSn register. To use the channel-dedicated sample-and-hold circuit, set the corresponding channel in any of the ADSCS0, ADSCS1, or ADSCS2 register.
(3) Each time A/D conversion of a single channel is completed, the result of A/D conversion is stored in the corresponding A/D data register (ADDRy).
(4) When A/D conversion of all the selected channels is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (scan end interrupt is enabled). At the same time, analog input sampling is started for all the channels for which the channel-dedicated sample-and-hold circuits are to be used.
(5) The ADCSR.ADST bit is not automatically cleared and steps (2) to (4) are repeated as long as the bit remains 1. When the ADCSR.ADST bit is set to 0 (stopping A/D conversion), A/D conversion stops and the 12-bit A/D converter enters a wait state.
(6) When the ADCSR.ADST bit is then set to 1 (starting A/D conversion), analog input sampling is started again for all the channels for which the channel-dedicated sample-and-hold circuits are to be used.

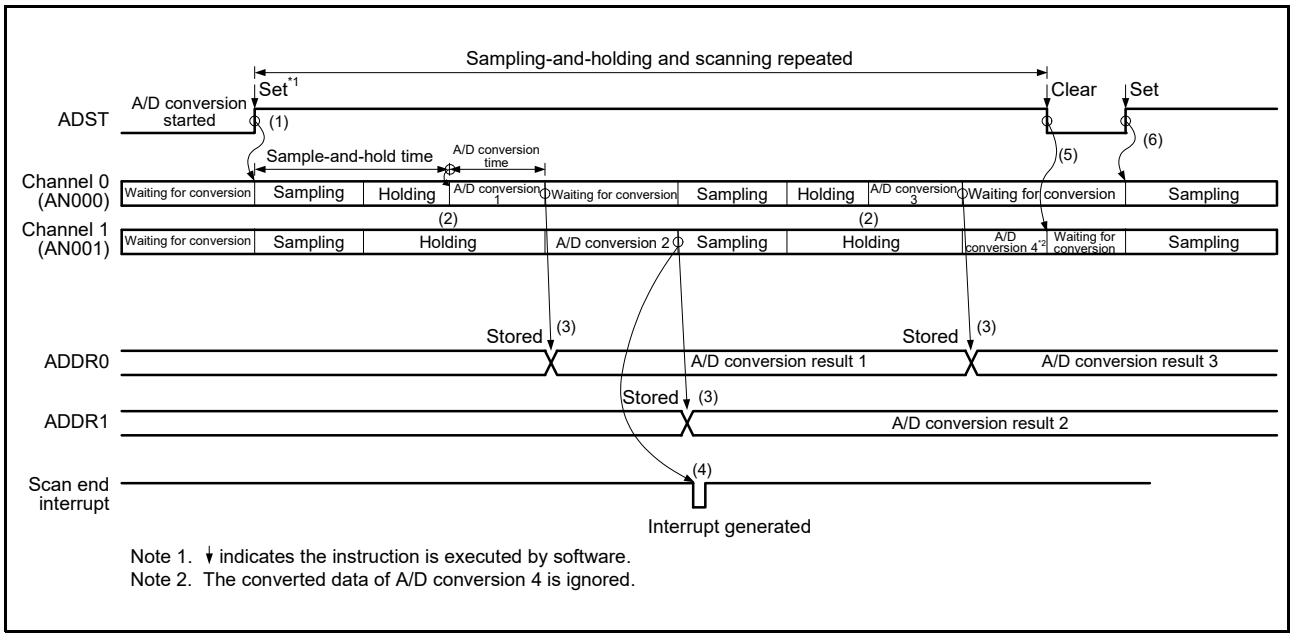


Figure 42.19 Example of Operation in Continuous Scan Mode (Channel-Dedicated Sample-and-Hold Circuits Used, AN000/AN001 Selected, with Constant Sampling Disabled)

42.3.4.3 Basic Operation (with Channel Sample-and-Hold Circuits, with Constant Sampling Enabled)

When constant sampling is enabled and a channel sample-and-hold circuit is used, sample-and-hold operations are performed first and A/D conversion is repeated for analog input of all selected channels as described below.
The ADSHCR.SHANS[2:0] bits are used to select the channels for which the channel-dedicated sample-and-hold circuits are to be used.
In continuous scan mode, specify A/D conversion select bits for the temperature sensor and for the internal reference voltage to deselect respectively.

(1) When the ADSHMSR.SHMD bit is set to 1, the sample-and-hold circuit selected in the ADSHCR.SHANS[2:0] bits starts constant sampling.
(2) Analog input sampling of all channels for which the channel-dedicated sample-and-hold circuits are to be used is started when the ADCSR.ADST bit is set to 1 (starting A/D conversion) by software, or synchronous or asynchronous trigger input. After setting the ADSHMSR.SHMD bit to 1 and 750 ns minimum are elapsed, set the ADCSR.ADST bit to 1.
(3) When the stabilizing time for the sample-and-hold circuit is elapsed, A/D conversion is performed for the analog channels selected in the ADANSA0 and ADANSA1 registers according to the order of conversion specified in the ADSCSn register. To use the channel-dedicated sample-and-hold circuit, set the corresponding channel in any of the ADSCS0, ADSCS1, or ADSCS2 register.
(4) Each time A/D conversion of a single channels is completed, the A/D-converted value is stored in the corresponding A/D data register (ADDRy), the sample-and-hold circuit resumes constant sampling.
(5) After completion of A/D conversion of all selected channels, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (scan end interrupt is enabled). In addition, all channels that use channel sample-and-hold circuit start holding of analog inputs.
(6) The ADCSR.ADST bit is not automatically cleared and steps (3) to (5) are repeated as long as the bit remains 1. When the ADCSR.ADST bit is set to 0 (stopping A/D conversion), A/D conversion is stopped and the 12-bit A/D converter enters into a wait state.
(7) When the ADSHMSR.SHMD bit is cleared to 0, the sample-and-hold circuit stops.
(8) After that, when the ADSHMSR.SHMD bit is set to 1, the sample-and-hold circuit selected in the ADSHCR.SHANS[2:0] bits starts constant sampling.
(9) When the ADCSR.ADST bit is set to 1 (starting A/D conversion), all channels that uses sample-and-hold circuit starts holding analog inputs.

Note: If only a channel with a channel-dedicated sample-and-hold circuit is selected and continuous scanning is performed, securing a period for normal sampling will not be possible in the second and subsequent rounds of continuous scanning. To proceed with scanning continuously while the sample-and-hold circuit is enabled, select AN003 and set the period for constant sampling of the sample-and-hold circuit to at least 400 ns.

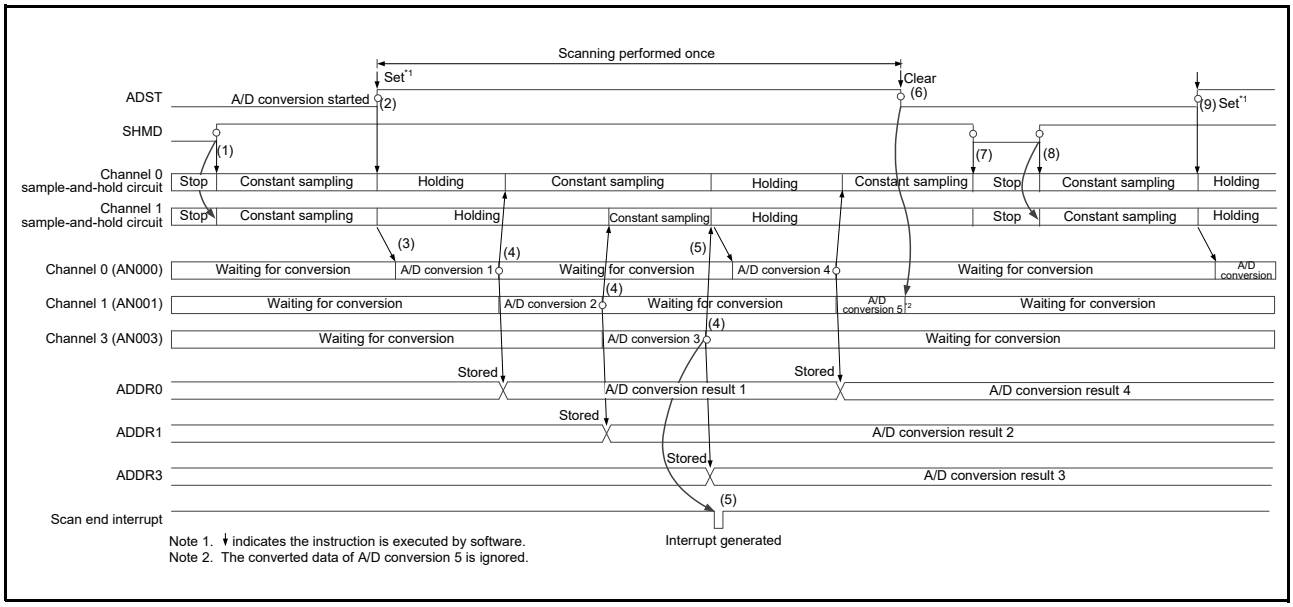
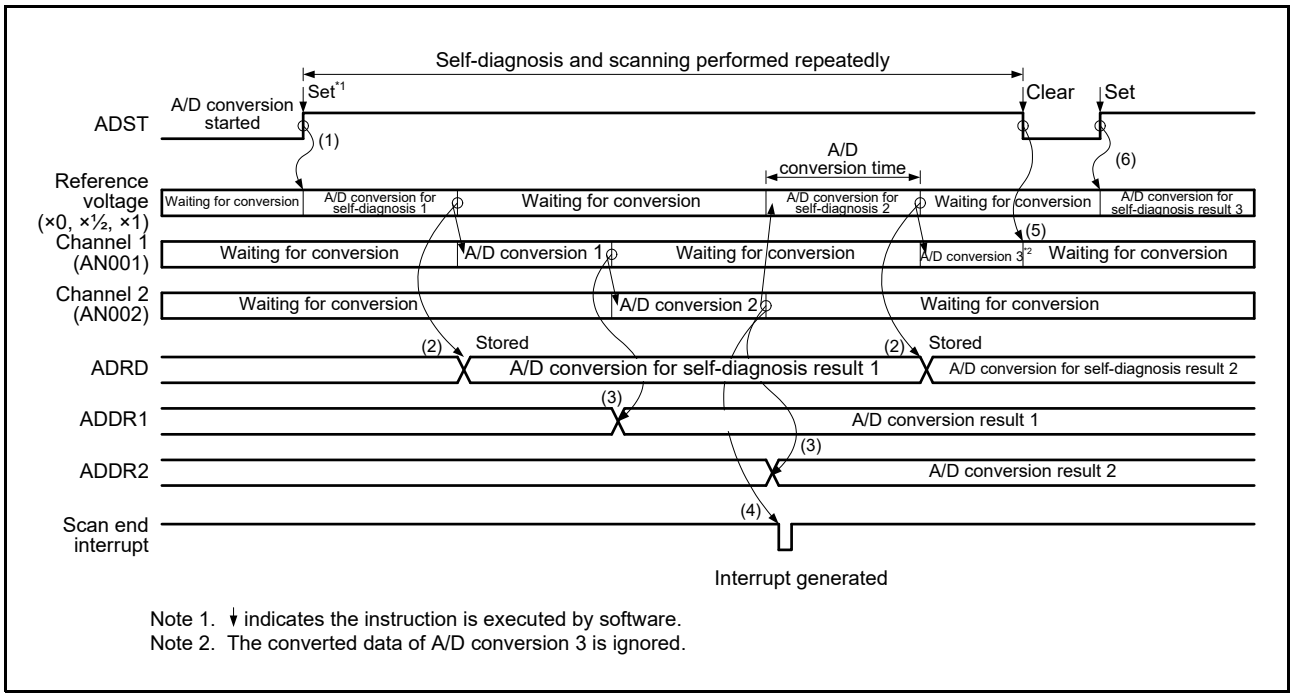


Figure 42.20 Example of Operation in Continuous Scan Mode (Channel-Dedicated Sample-and-Hold Circuits Used: AN000, AN001, and AN003 Selected, with Constant Sampling Enabled)

42.3.4.4 Channel Selection and Self-Diagnosis (Without Channel-Dedicated Sample-and-Hold Circuits)

When channels and self-diagnosis are selected at the same time, A/D conversion is first performed for the reference voltage supplied to the 12-bit A/D converter, and then A/D conversion is performed on the analog input of the selected channels, which sequence is repeated as below.

(1) When the ADCSR.ADST bit is set to 1 (starting A/D conversion) by software, or synchronous or asynchronous trigger input, A/D conversion for self-diagnosis is started first.
(2) When A/D conversion for self-diagnosis is completed, A/D conversion result is stored into the A/D self-diagnosis data register (ADRD), A/D conversion is performed for the analog channels selected in the ADANSA0 and ADANSA1 registers according to the order of conversion specified in the ADSCSn register.
(3) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(4) When A/D conversion of all the selected channels is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (scan end interrupt is enabled). At the same time, the 12-bit A/D converter starts A/D conversion for self-diagnosis, and, upon its completion, starts A/D conversion for the analog channels selected in the ADANSA0 and ADANSA1 registers according to the order of conversion specified in the ADSCSn register.
(5) The ADCSR.ADST bit is not automatically cleared and steps (2) to (4) are repeated as long as the bit remains 1. When the ADST bit is set to 0 (stopping A/D conversion), A/D conversion stops and the 12-bit A/D converter enters a wait state.
(6) When the ADCSR.ADST bit is later becomes 1 (starting A/D conversion), the A/D conversion for self-diagnosis is started again.



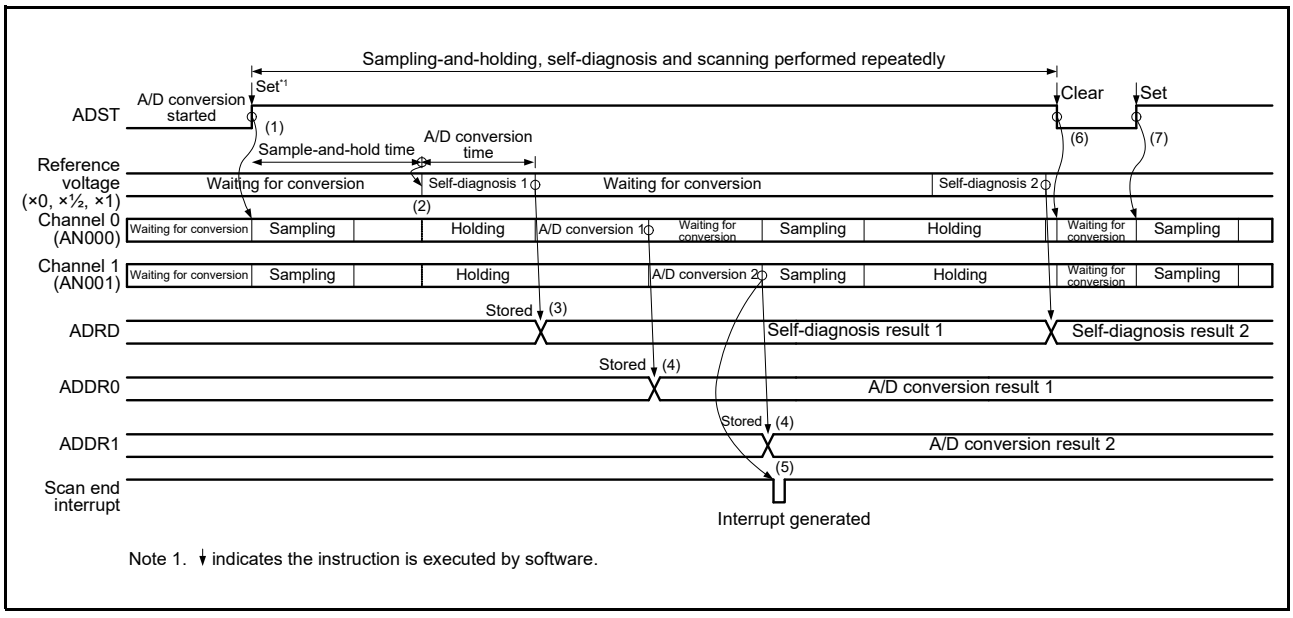
Note 1. ↓ indicates the instruction is executed by software.
Note 2. The converted data of A/D conversion 3 is ignored.

Figure 42.21 Example of Operation in Continuous Scan Mode (Basic Operation; AN001 and AN002 Selected + Self-Diagnosis)

42.3.4.5 Channel Selection and Self-Diagnosis (with Channel-Dedicated Sample-and-Hold Circuits, with Constant Sampling Disabled)

When channels and self-diagnosis are selected and a channel-dedicated sample-and-hold circuit is used, sample-and-hold operations are performed first, and A/D conversion is performed for the reference voltage supplied to the 12-bit A/D converter as below. After that, A/D conversion is performed repeatedly on the analog input of the selected channels. In continuous scan mode, specify A/D conversion select bits for the temperature sensor and for the internal reference voltage to deselect respectively.

(1) Analog input sampling of all channels for which the channel-dedicated sample-and-hold circuits are to be used is started when the ADCSR.ADST bit is set to 1 (starting A/D conversion) by software, or synchronous or asynchronous trigger input.
(2) A/D conversion for self-diagnosis is started after completion of sampling and holding.
(3) When A/D conversion for self-diagnosis is completed, A/D conversion result is stored into the A/D self-diagnosis data register (ADRD), A/D conversion is performed for the analog channels selected in the ADANSA0 and ADANSA1 registers according to the order of conversion specified in the ADSCSn register. To use the channel-dedicated sample-and-hold circuit, set the corresponding channel in any of the ADSCS0, ADSCS1, or ADSCS2 register.
(4) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(5) When A/D conversion of all the selected channels is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (scan end interrupt is enabled). At the same time, analog input sampling is started for all the channels for which the channel-dedicated sample-and-hold circuits are to be used.
(6) The ADCSR.ADST bit is not automatically cleared and steps (2) to (5) are repeated as long as the ADCSR.ADST bit as long as the bit remains 1. When the ADCSR.ADST bit is set to 0 (stopping A/D conversion), A/D conversion stops and the 12-bit A/D converter enters a wait state.
(7) When the ADCSR.ADST bit is later becomes 1 (starting A/D conversion), the A/D conversion for self-diagnosis is started again.



Note 1. ↓ indicates the instruction is executed by software.

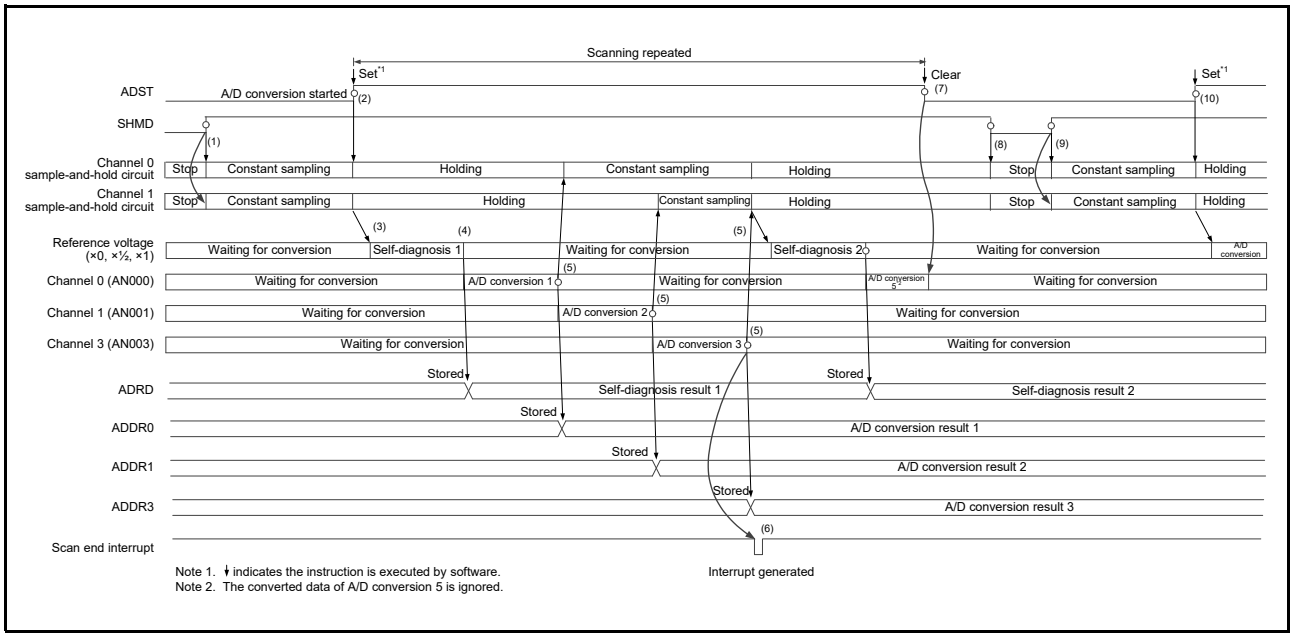
Figure 42.22 Example of Operation in Continuous Scan Mode (Channel-Dedicated Sample-and-Hold Circuits Used: AN000 and AN001 Selected + Self-Diagnosis)

42.3.4.6 Channel Selection and Self-Diagnosis (with Channel-Dedicated Sample-and-Hold Circuits, with Constant Sampling Enabled)

When self-diagnosis is selected with channels and a channel-dedicated sample-and-hold circuit is used with constant sampling enabled, sample-and-hold operations are performed first and A/D conversion is performed for the reference voltage to be supplied to the 12-bit A/D converter as described below. After that, A/D conversion is performed repeatedly on the analog input of the selected channels.

(1) When the ADSHMSR.SHMD bit is set to 1, the sample-and-hold circuit selected in the ADSHCR.SHANS[2:0] bits starts constant sampling.
(2) Analog input sampling of all channels for which the channel-dedicated sample-and-hold circuits are to be used is started when the ADCSR.ADST bit is set to 1 (starting A/D conversion) by software, or synchronous or asynchronous trigger input. After setting the ADSHMSR.SHMD bit to 1 and 750 ns minimum are elapsed, set the ADCSR.ADST bit to 1.
(3) When the stabilizing time for the sample-and-hold circuit is elapsed, A/D conversion by self-diagnosis is started.
(4) When A/D conversion for self-diagnosis is completed, A/D conversion result is stored into the A/D self-diagnosis data register (ADRD), A/D conversion is performed for the analog channels selected in the ADANSA0 and ADANSA1 registers according to the order of conversion specified in the ADSCSn register. To use the channel-dedicated sample-and-hold circuit, set the corresponding channel in any of the ADSCS0, ADSCS1, or ADSCS2 register.
(5) Each time A/D conversion of a single channels is completed, the A/D-converted value is stored in the corresponding A/D data register (ADDRy), the sample-and-hold circuit resumes constant sampling.
(6) After completion of A/D conversion of all selected channels, if the ADCSR.ADIE bit is set to 1 (scan end interrupt is enabled), a scan end interrupt request is generated. In addition, all channels that use channel sample-and-hold circuit start holding of analog inputs.
(7) The ADCSR.ADST bit is not automatically cleared and steps (3) to (6) are repeated as long as the bit remains 1. When the ADCSR.ADST bit is set to 0 (stopping A/D conversion), A/D conversion is stopped and the 12-bit A/D converter enters into a wait state.
(8) When the ADSHMSR.SHMD bit is cleared to 0, the sample-and-hold circuit stops.
(9) After that, when the ADSHMSR.SHMD bit is set to 1, the sample-and-hold circuit selected in the ADSHCR.SHANS[2:0] bits starts constant sampling.
(10) When the ADCSR.ADST bit is set to 1 (starting A/D conversion), all channels that uses sample-and-hold circuit starts holding analog inputs.

Note: If only a channel with a channel-dedicated sample-and-hold circuit is selected and continuous scanning is performed, securing a period for normal sampling will not be possible in the second and subsequent rounds of continuous scanning. To proceed with scanning continuously while the sample-and-hold circuit is enabled, select AN003 and set the period for constant sampling of the sample-and-hold circuit to at least 400 ns.



Note 1. ↓ indicates the instruction is executed by software.
Note 2. The converted data of A/D conversion 5 is ignored.

Figure 42.23 Example of Operation in Continuous Scan Mode (Channel-Dedicated Sample-and-Hold Circuits Used: AN000, AN001, and AN003 Selected + Self-Diagnosis, with Constant Sampling Enabled)

42.3.5 Group Scan Mode

42.3.5.1 Basic Operation

Either two (groups A and B) or three (groups A, B, and C) can be selected as the number of the groups to be used in group scan mode.

In basic operation of group scan mode, A/D conversion is performed once on the analog inputs of all the specified channels in groups A and B, or groups A, B, and C after scan is started by a synchronous trigger as below. Scan operation of each group is similar to the scan operation in single scan mode.

The synchronous triggers of groups A, B, and C can be selected using the TRSA[6:0] and TRSB[6:0] bits in the ADSTRGR register, and the ADGCTRGR2.TRSC6 and ADGCTRGR.TRSC[5:0] bits, respectively. Different triggers should be used for each group A, B, and C so that scanning of groups A, B, and C does not occur simultaneously. Software trigger should not be used.

The channels to be scanned are selected using registers ADANSA0 and ADANSA1 and bits TSSA and OCSA in register ADEXICR for group A, registers ADANSB0 and ADANSB1 and bits TSSB and OCSB in register ADEXICR for group B, and registers ADANSC0 and ADANSC1 and bits TSSC and OCSC in register ADGCEXCR for group C.

When the temperature sensor output is to be scanned, set only one from among the TSSA, TSSB, and TSSC bits to 1. In the same way, when the internal reference voltage is to be scanned, set only one from among the OCSA, OCSB, and OCSC bits to 1. Discharging for a period of 15 ADCLK is performed for all the conversion channels in a scan group where the temperature sensor output or the internal reference voltage is set.

When self-diagnosis is selected in group scan mode, self-diagnosis is separately executed for groups A and B, or groups A, B, and C.

The following describes operation in group scan mode using a trigger from the MTU. The TRG4AN, TRG4BN, and TRG4ABN triggers from the MTU are assumed to be used to start conversion of groups A, B and C, respectively.

(1) Scanning of group A is started by the TRG4AN trigger from the MTU.
(2) When group A scanning is completed, a scan end interrupt is generated if the ADCSR.ADIE bit is 1 (scan end interrupt is enabled).
(3) Scanning of group B is started by the TRG4BN trigger from the MTU.
(4) When group B scanning is completed, a group B scan end interrupt is generated if the ADCSR.GBADIE bit is 1 (group B scan end interrupt is enabled).
(5) Scanning of group C is started by the TRG4ABN trigger from the MTU.
(6) When group C scanning is completed, a group C scan end interrupt is generated if the ADGCTRGR.GCADIE bit is 1 (group C scan end interrupt is enabled).

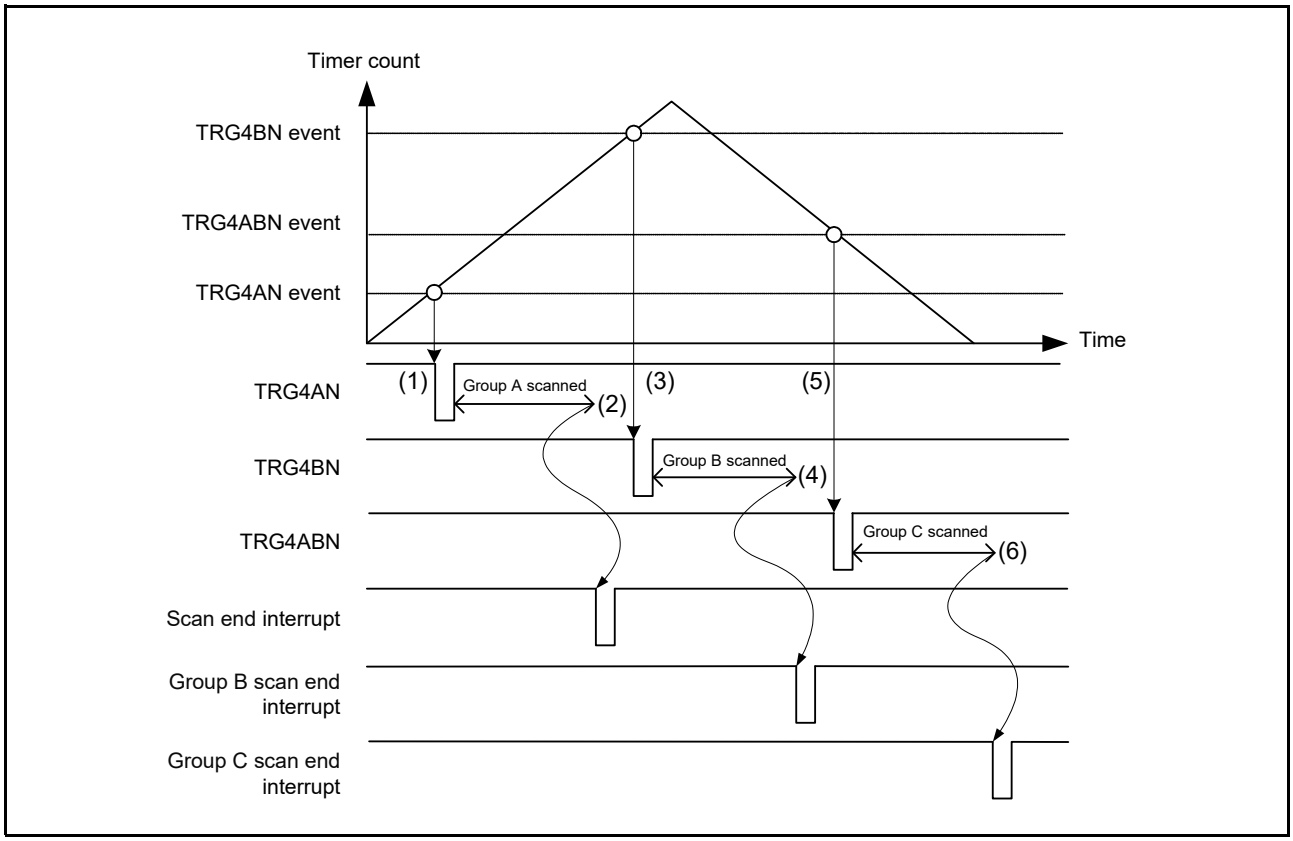


Figure 42.24 Example of Operation in Group Scan Mode (Basic Operation: Synchronous Triggers from MTU Used)

The following describes operation using the temperature sensor output or the internal reference voltage (in group scan mode and not in group priority operation).

The operation is based on a setting: channel 0 to be A/D-converted for group A, the temperature sensor output and internal reference voltage to be A/D-converted for group B, and channel 1 to be A/D-converted for group C.

(1) When input of the group A trigger signal sets the ADCSR.ADST bit to 1 (starting A/D conversion), scan for group A starts.

(2) Start A/D conversion for channel 0, and on its completion, the A/D-converted value is stored in the corresponding A/D data register (ADDR0).

(3) A scan end interrupt is generated if the ADCSR.ADIE bit is set to 1 (scan end interrupt is enabled).

(4) The ADCSR.ADST bit is automatically cleared to 0 when scan for group A is completed, and the 12-bit A/D converter enters a wait state.

(5) When input of the group B trigger signal sets the ADCSR.ADST bit to 1 (starting A/D conversion), scan for group B starts.

(6) Before A/D-converting the temperature sensor output, 15 ADCLK is inserted as a discharging period. After a completion of the discharging period, A/D conversion for the temperature sensor output starts, and the A/D-converted value is stored in the corresponding A/D data register (ADTSDR) upon completion of A/D conversion. Next, before A/D-converting the internal reference voltage, 15 ADCLK is inserted as a discharging period. After a completion of the discharging period, A/D conversion for the internal reference voltage starts, and the A/D-converted value is stored in the corresponding A/D data register (ADOCDR) upon completion of A/D conversion.

(7) Group B scan end interrupt is output if the ADCSR.GBADIE bit is set to1 (group B scan end interrupt is enabled).

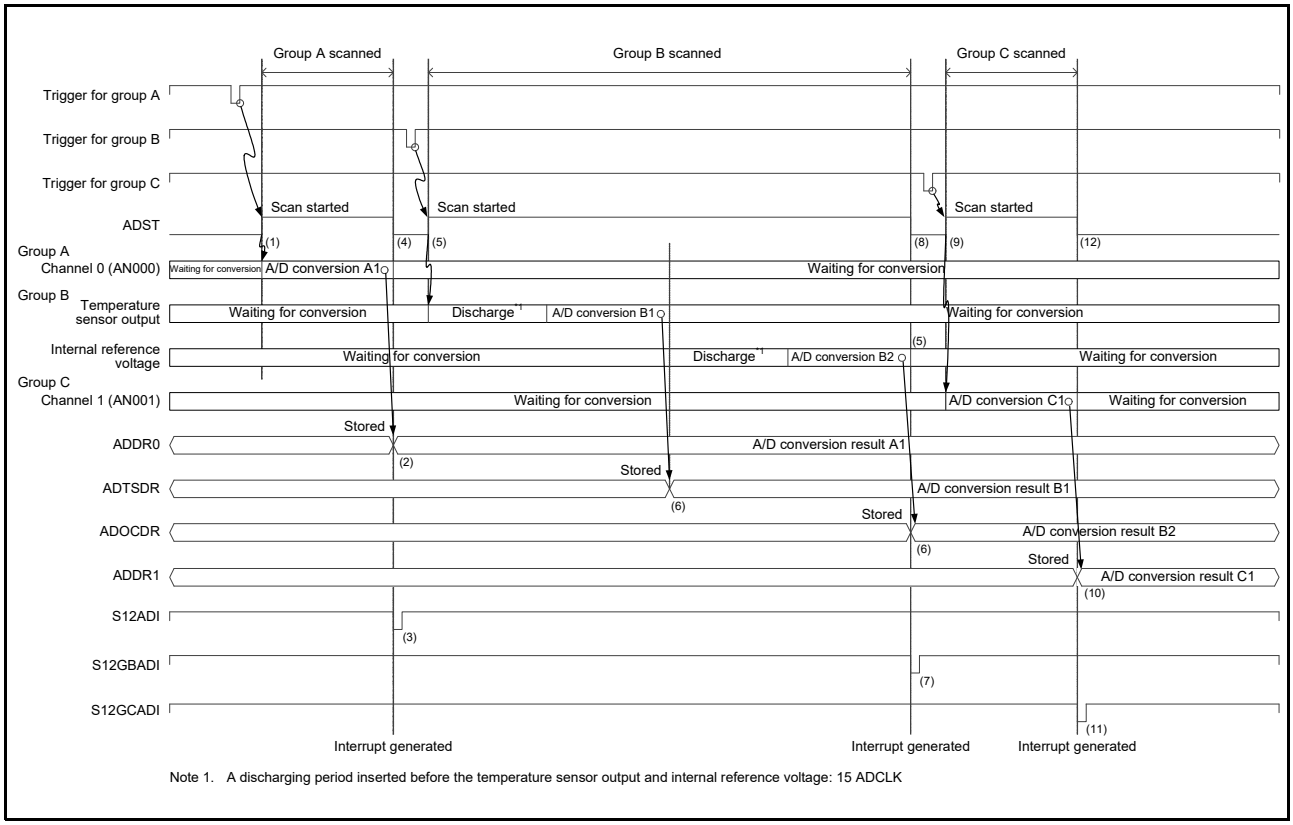
(8) The ADCSR.ADST bit is automatically cleared to 0 when scan for group B is completed, and the 12-bit A/D converter enters a wait state.

(9) When input of the group C trigger signal sets the ADCSR.ADST bit to 1 (starting A/D conversion), scan for group C starts.

(10) Start A/D conversion for channel 1, and on its completion, the A/D-converted value is stored in the corresponding A/D data register (ADDR1).

(11) Group C scan end interrupt is output if the ADGCTRGR.GCADIE bit is set to1 (group C scan end interrupt is enabled).

(12) The ADCSR.ADST bit is automatically cleared to 0 when scan for group C is completed, and the 12-bit A/D converter enters a wait state.



Note 1. A discharging period inserted before the temperature sensor output and internal reference voltage: 15 ADCLK

Figure 42.25 Operation When Using the Temperature Sensor Output and the Internal Reference Voltage (in Group Scan Mode and Not in Group Priority Operation)

42.3.5.2 A/D Conversion in Double Trigger Mode

When double trigger mode is selected in group scan mode, two rounds of single scan operation started by a synchronous trigger are performed as a sequence for group A. For groups B and C, single scan operation started by a synchronous trigger is performed once.
In group scan mode, the synchronous triggers of groups A, B, and C can be selected using the TRSA[6:0] and TRSB[6:0] bits in the ADSTRGR register, and the ADGCTRGR2.TRSC6 and ADGCTRGR.TRSC[5:0] bits, respectively. Different triggers should be used for each group A, B, and C so that scanning of groups A, B, and C does not occur simultaneously. Software trigger and asynchronous trigger should not be used. When TRG4AN or TRG4BN, TRG7AN or TRG7BN, TRGA0N or TRG0N, TRGA9N or TRG9N, TRGA0N or TRGA9N, TRG0N or TRG9N, GTADTRA0N or GTADTRB0N, GTADTRA1N or GTADTRB1N, GTADTRA2N or GTADTRB2N, GTADTRA3N or GTADTRB3N, GTADTRA4N or GTADTRB4N, GTADTRA5N or GTADTRB5N, GTADTRA6N or GTADTRB6N, GTADTRA7N or GTADTRB7N, or ELCTRG00N or ELCTRG01N/ELCTRG10N or ELCTRG11N/ELCTRG20N or ELCTRG21N is selected as the synchronous trigger of group A by the ADSTRGR.TRSA[6:0] bits, operation is performed in extended double trigger mode.
The channels to be scanned are selected using bits ADCSR.DBLANS[4:0] for group A, registers ADANSB0 and ADANSB1 for group B, and registers ADANSC0 and ADANSC1 for group C.
When double trigger mode is selected in group scan mode, the temperature sensor A/D conversion select bit (ADEXICR.TSSA) and internal reference voltage A/D conversion select bit (ADEXICR.OCSA) are set to 0 (not selected).
When double trigger mode is selected in group scan mode, self-diagnosis cannot be selected.
Duplication of A/D conversion data is enabled by setting the channel numbers to be duplicated to the ADCSR.DBLANS[4:0] bits and setting the ADCSR.DBLE bit to 1.
The following describes operation in group scan mode with double trigger mode using a synchronous trigger from the MTU. The TRG4ABN, TRGA0N, and TRGA1N triggers from the MTU are assumed to be used to start conversion of groups A, B, and C, respectively.

(1) Scanning of group C is started by the TRGA1N trigger from the MTU.
(2) When group C scanning is completed, a group C scan interrupt is generated if the ADGCTRGR.GCADIE bit is 1 (group C scan end interrupt is enabled).
(3) Scanning of group B is started by the TRGA0N trigger from the MTU.
(4) When group B scanning is completed, a group B scan end interrupt is generated if the ADCSR.GBADIE bit is 1 (group B scan end interrupt is enabled).
(5) The first scanning of group A is started by the first TRG4ABN trigger from the MTU.
(6) When the first scanning of group A is completed, the conversion result is stored into the corresponding A/D data register (ADDRy); a scan end interrupt request is not generated irrespective of the ADCSR.ADIE bit setting.
(7) The second scanning of group A is started by the second TRG4ABN trigger from the MTU.
(8) When the second scanning of group A is completed, the conversion result is stored into the ADDBLDR register a scan end interrupt is generated if the ADCSR.ADIE bit is 1 (scan end interrupt is enabled).
(9) The second scanning of group B is started by the second TRGA0N trigger from the MTU.
(10) When the second scanning of group B is completed, a group B scan end interrupt is generated if the ADCSR.GBADIE bit is 1 (group B scan end interrupt is enabled).
(11) The second scanning of group C is started by the second TRGA1N trigger from the MTU.
(12) When the second scanning of group C is completed, a group C scan end interrupt is generated if the ADGCTRGR.GCADIE bit is 1 (group C scan end interrupt is enabled).

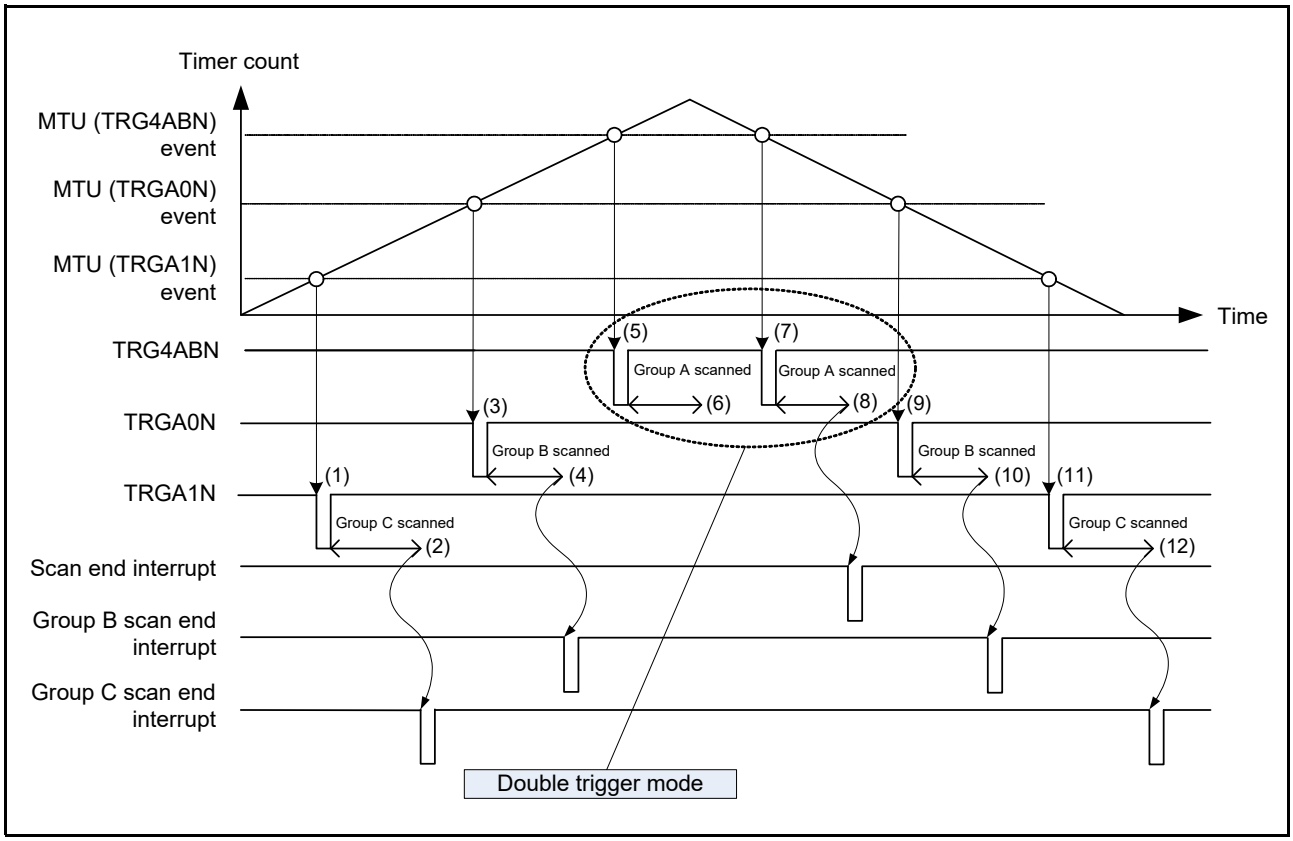


Figure 42.26 Example of Operation in Group Scan Mode with Double Trigger Mode (Basic Operation: Synchronous Triggers from MTU Used)

42.3.5.3 Operation under Group Priority Control

Setting the ADGSPCR.PGS bit to 1 in group scan mode makes operation proceed under group priority control. The group priority order is group A > group B > group C. Either two (groups A and B) or three (groups A, B, and C) can be selected as the number of the groups to be used in group scan mode. When setting the ADGSPCR.PGS bit to 1, follow the procedure described in Figure 42.27. If the procedure is not followed, proper scanning operation and data storage are not guaranteed.

In basic operation of group scan mode, if group A, B, or C is scanning, all other trigger inputs are ignored. Under group priority control, if a priority group trigger is input during A/D conversion for the low-priority group, A/D conversion for the low-priority group is interrupted and A/D conversion for the priority group proceeds.

If the ADGSPCR.GBRSCN bit is 0, the converter enters a wait state for the low-priority group on completion of the A/D conversion for the priority group.

The trigger input for the low-priority group during scanning is ignored.

If the ADGSPCR.GBRSCN bit is 1, the converter automatically restarts scanning for the low-priority group from the head of the group after A/D conversion for the priority group.

Also, the trigger input for the low-priority group generated during scanning for the priority group is enabled, and the converter automatically restarts scanning for the low-priority group after scanning for the priority group.

When the ADGSPCR.LGRRS bit is 0 while the ADGSPCR.GBRSCN bit is 1, the converter restarts scanning for the low-priority group from the head of the group. When the ADGSPCR.LGRRS bit is 1, the converter restarts scanning for the low-priority group from the channel on which scanning is interrupted.

However, when self-diagnosis is used, the converter restarts scanning from the channel on which scanning is interrupted after self-diagnosis is completed.

Table 42.20 summarizes operations in response to the input of a trigger during scanning with the settings of the ADGSPCR.GBRSCN bit.

When the ADGSPCR.GBRP bit is set to 1, scanning operations for the group of the lowest priority are continuously performed in single scan mode.

For the trigger settings in group scan mode, select a synchronous trigger for group A using the ADSTRGR.TRSA[6:0] bits, select a synchronous trigger different from that of group A for group B using the ADSTRGR.TRSB[6:0] bits, and select a synchronous trigger different from those of groups A and B for group C using the ADGCTRGR2.TRSC6 and ADGCTRGR.TRSC[5:0] bits.

Set the ADSTRGR.TRSB[6:0] bits to 3Fh or 7Fh when setting group scan mode for two groups (the ADGCTRGR.GRCE bit to 0) and setting the ADGSPCR.GBRP bit to 1.

Set the ADGCTRGR.TRSC[5:0] bits to 3Fh when setting group scan mode for three groups (the ADGCTRGR.GRCE bit to 1) and setting the ADGSPCR.GBRP bit to 1.

Furthermore, as targets for scanning, select channels for group A by using the ADANSA0 and ADANSA1 registers, and the ADEXICR.TSSA and ADEXICR.OCSA bits; select the group B channels by using the ADANSB0 and ADANSB1 registers, and the ADEXICR.TSSB and ADEXICR.OCSB bits; and select the group C by using the ADANSC0 and ADANSC1 registers, and the ADGCEXCR.TSSC and ADGCEXCR.OCSC bits respectively.

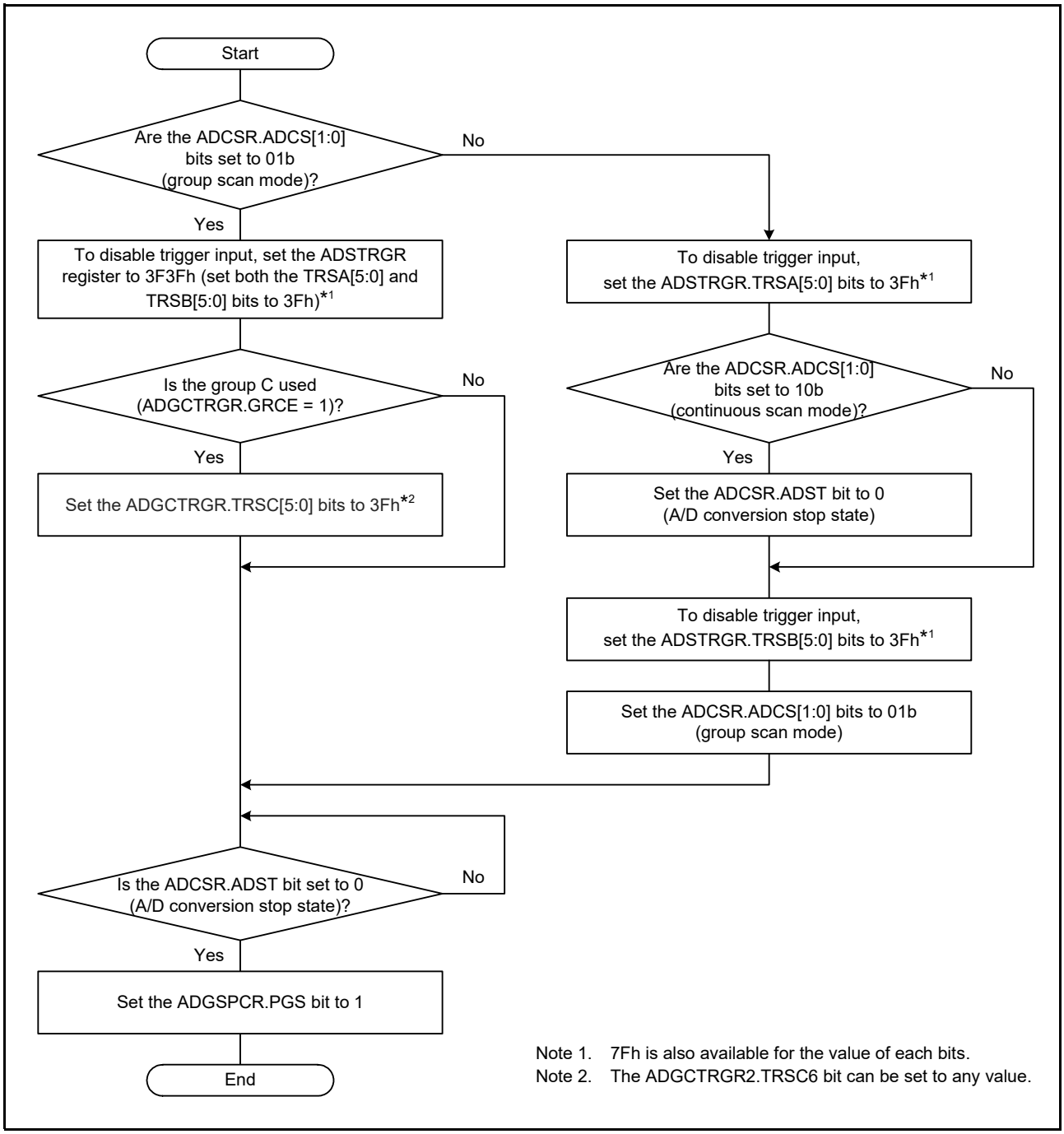


Figure 42.27 Flow of Setting the ADGSPCR.PGS Bit

Table 42.20 Control of Scanning Operations According to the Settings of the ADGSPCR.GBRSCN Bit

Scanning Operation	Trigger Input	ADGSPCR.GBRSCN = 0	ADGSPCR.GBRSCN = 1
When A/D conversion for group A is in progress	Input of trigger for group A	Trigger input is ineffective.	Trigger input is ineffective.
	Input of trigger for group B	Trigger input is ineffective.	A/D conversion is performed on group B after A/D conversion on group A is completed.
	Input of trigger for group C	Trigger input is ineffective.	A/D conversion is performed on group C after A/D conversion on group A is completed.
When A/D conversion for group B is in progress	Input of trigger for group A	A/D conversion in progress for group B interrupted and conversion for group A starts.	• A/D conversion in progress for group B is interrupted and conversion for group A starts. • A/D conversion for group B starts after conversion for group A is completed.
	Input of trigger for group B	Trigger input is ineffective.	Trigger input is ineffective.
	Input of trigger for group C	Trigger input is ineffective.	A/D conversion is performed on group C after A/D conversion on group B is completed.
When A/D conversion for group C is in progress	Input of trigger for group A	A/D conversion in progress for group C interrupted and conversion for group A starts.	• A/D conversion in progress for group C is interrupted and conversion for group A starts. • A/D conversion for group C starts after conversion for group A is completed.
	Input of trigger for group B	A/D conversion in progress for group C is interrupted and conversion for group B starts.	• A/D conversion in progress for group C is interrupted and conversion for group B starts. • A/D conversion for group C starts after conversion for group B is completed.
	Input of trigger for group C	Trigger input is ineffective.	Trigger input is ineffective.

When using group priority operation mode, refer to the following tables to select the desirable operating mode and set the registers.

Table 42.21 Group Priority Operation Setting and Operating Mode for Two Groups (ADGSPCR.PGS = 1, ADGCTRGR.GRCE = 0)

ADGSPCR			
GBRSCN	LGRRS	GBRP	Operation
0	x	0	Group priority operation for two groups (groups A and B) • When a group A trigger is input, group B scan is completed (not restarted)
1	0	0	Group priority operation for two groups (groups A and B) • After a scan for group B is interrupted, upon completion of a scan for group A, a scan for group B starts again from the beginning according to the order specified in the ADANSB0 register.
1	1	0	Group priority operation for two groups (groups A and B) • After group B scan is interrupted, group B scan is restarted from the channel on which scan was interrupted*1, among the channels selected with the ADANSB0 register after group A scan is completed.
x	0	1	Group priority operation for two groups (groups A and B) • Single scan for group B is started continuously without start trigger input. After group B scan is interrupted, single scan is restarted from the head of the channel selected with the ADANSB0 register after scan for group A is completed.
1	1	1	Group priority operation for two groups (groups A and B) • Single scan for group B is started continuously without start trigger input. After group B scan was interrupted, single scan is restarted from the channel on which scan was interrupted*1, among the channels selected with the ADANSB0 register after group A scan is completed.

x = Don't care

Note 1. When self-diagnosis is used (ADCER.DIAGM = 1), A/D conversion of the interrupted channel is started after self-diagnosis.

Table 42.22 Group Priority Operation Setting and Operating Mode for Three Groups (ADGSPCR.PGS = 1, ADGCTRGR.GRCE = 1)

ADGSPCR			
GBRSCN	**LGRRS**	**GBRP**	**Operation**
0	x	0	Group priority operation for three groups (groups A, B, and C) • When a group A trigger is input, group B scan is completed (not restarted) • When a trigger for group A or B is input, group C scan is completed (not restarted).
0	x	1	Group priority operation for three groups (groups A, B, and C) • When a group A trigger is input, group B scan is completed (not restarted) • Single scan for group C is started continuously without start trigger input. After group C scan is interrupted, scan is restarted from the head of the channel selected with the ADANSC0 and ADANSC1 registers after scan for groups A and B is completed.
1	0	0	Group priority operation for three groups (groups A, B, and C) • After group B scan is interrupted, scan is restarted from the head of the channel selected with the ADANSB0 and ADANSB1 registers after group A scan is completed. • After group C scan is interrupted, scan is restarted from the head of the channel selected with the ADANSC0 and ADANSC1 registers after scan for groups A and B is completed.
1	1	0	Group priority operation for three groups (groups A, B, and C) • After group B scan is interrupted, scan is restarted from the channel on which scan was interrupted*1, among the channels selected with the ADANSB0 and ADANSB1 registers after scan for group A is completed. • After group C scan is interrupted, scan is restarted from the channel on which scan was interrupted*1, among the channels selected with the ADANSC0 and ADANSC1 registers after scan for groups A and B is completed.
1	0	1	Group priority operation for three groups (groups A, B, and C) • After group B scan is interrupted, scan is restarted from the head of the channel selected with the ADANSB0 and ADANSB1 registers after group A scan is completed. • Single scan for group C is started continuously without start trigger input. After group C scan is interrupted, single scan is restarted from the head of the channel selected with the ADANSC0 and ADANSC1 registers after scan for groups A and B is completed.
1	1	1	Group priority operation for three groups (groups A, B, and C) • After scanning for group B is interrupted, once scanning for group A completes, scanning is resumed starting from the channel on which scanning was interrupted*1, among those channels selected with the ADANSB0 and ADANSB1 registers. • Single scan for group C is started continuously without start trigger input. After group C scan is interrupted, single scan is restarted from the channel on which scan was interrupted*1, among the channels selected with the ADANSC0 and ADANSC1 registers after scan for groups A and B is completed.

x = Don’t care

Note 1. When self-diagnosis is used (ADCER.DIAGM = 1), A/D conversion of the interrupted channel is started after self-diagnosis.

(1) Group Priority Operation for Two Groups (ADGSPCR.PGS = 1, ADGCTRGR.GRCE = 0)

The following examples 1 to 5 show group priority operation in group scan mode (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, ADGSPCR.LGRRS = 0) when channel 0 is selected for group A and channels 1 to 3 are selected for group B.

Operation example 1: Input of the group A trigger signal during a group B scan, with the rescan setting

(1) When input of the group B trigger signal sets the ADCSR.ADST bit to 1 (starting A/D conversion), A/D conversion starts for the analog channels selected in the ADANSB0 and ADANSB1 registers according to the order of conversion specified in the ADSCSn register. (Specified conversion order in Figure 42.28: AN000 → AN001 → AN002 → AN003)

(2) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).

(3) When a group A trigger is input during the group B scanning, the ADCSR.ADST bit retains the value as 1, and the group B scan is interrupted, and the group A scan starts for the analog channels selected in the ADANSA0 and ADANSA1 registers according to the order of conversion specified in the ADSCSn register.

(4) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).

(5) A scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (scan end interrupt is enabled).

(6) When the ADGSPCR.GBRSCN bit is set to 1 (restarting a group scan which was interrupted due to group priority in scanning), the ADCSR.ADST bit retains the value as 1, and the group B scan for the analog channels selected in the ADANSB0 and ADANSB1 registers according to the order of conversion specified in the ADSCSn register is restarted.

(7) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).

(8) A group B scan end interrupt request is generated if the ADCSR.GBADIE bit is 1 (group B scan end interrupt is enabled).

(9) The ADCSR.ADST bit is automatically cleared to 0 when scan of all selected channels is completed, and the 12-bit A/D converter enters a wait state.

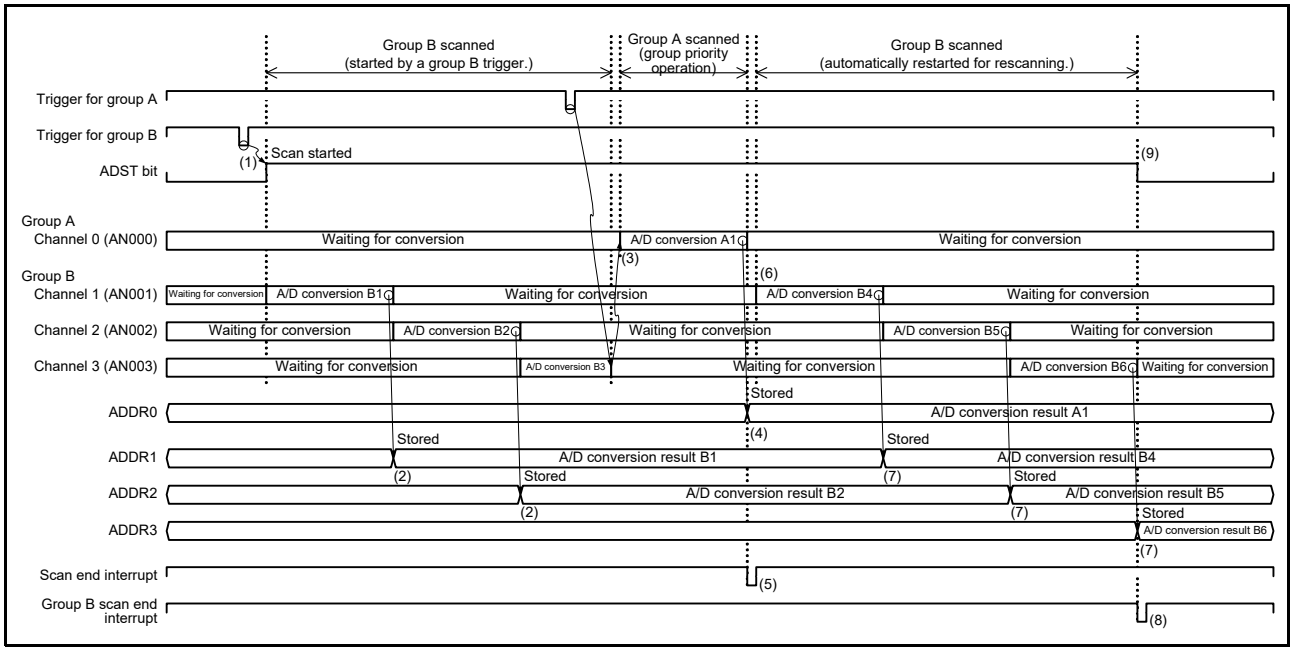


Figure 42.28 Example 1 of Group Priority Operation: Input of Group A Trigger Signal During Group B Scan, with the Rescan Setting (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, ADGSPCR.LGRRS = 0)

Operation example 2: Input of the group A trigger signal during a group B rescan, with the rescan setting

Figure 42.29 shows an example when a group A trigger is input during rescan operation on group B.

If a group A trigger is input, scan for group A starts even while rescan operation is in progress. Scan for group B starts after scan for group A is completed.

Operations of the ADCSR.ADST bit, storing to the A/D conversion result in the A/D data register (ADDRy), and interrupt requests are the same as example 1.

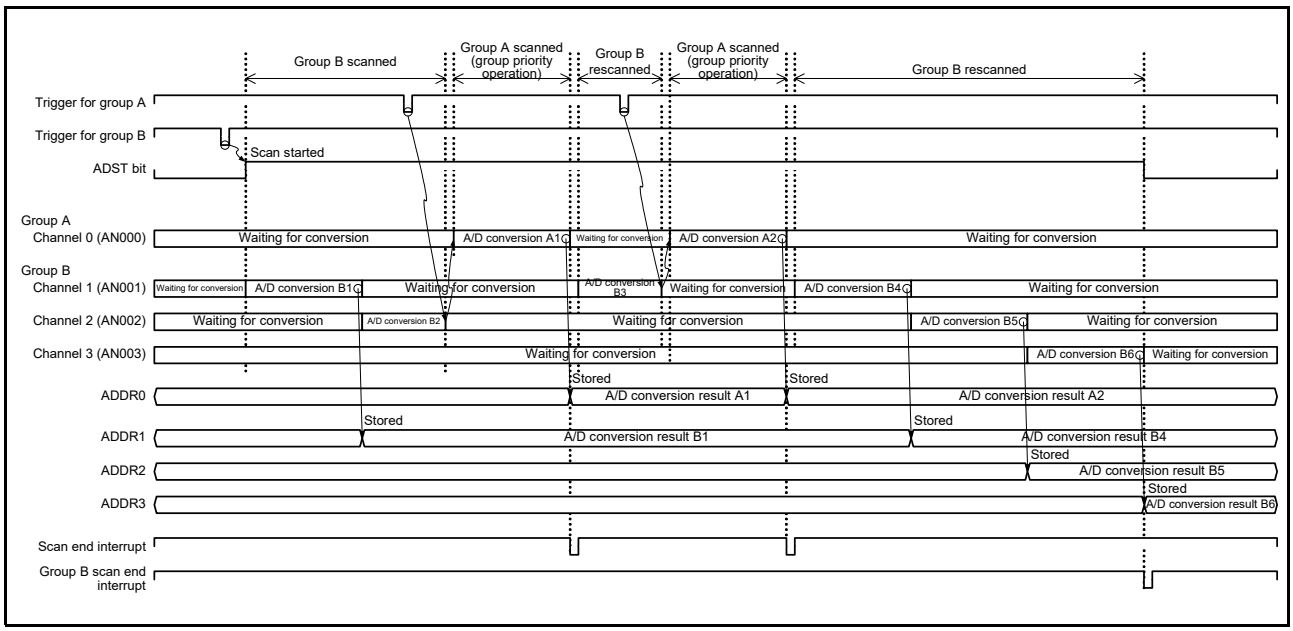


Figure 42.29 Example 2 of Group Priority Operation: Input of Group A Trigger Signal During Group B Rescan, with the Rescan Setting (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, ADGSPCR.LGRRS = 0)

Operation example 3: Input of the group B trigger signal during a group A scan, with the rescan setting

The following describes an example when a group B trigger is input during scan operation on group A when the ADGPSCR.GBRSCN bit is 1 (scan for the group is restarted after having been interrupted due to group priority operation).
If the ADGPSCR.GBRSCN bit is 0, all group B triggers that are input during scan operation on group A are disabled.

(1) When input of the group A trigger signal sets the ADCSR.ADST bit to 1 (starting A/D conversion), a group A scan starts for the analog channels selected in the ADANSA0 and ADANSA1 registers according to the order of conversion specified in the ADSCSn register.
(2) When a group B trigger is input during a group A scanning, a group B becomes ready for a scan.
(3) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(4) After scan for group A is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (scan end interrupt is enabled).
(5) After the group A scan ends, the ADCSR.ADST bit retains the value as 1, and the group B scan is started for the analog channels selected in the ADANSB0 and ADANSB1 registers according to the order of conversion specified in the ADSCSn register.
(When a group A trigger is input during the group B scanning, the group A scan is started in the same way as in the example 1, and the group B scan is restarted after the group A scan ends.)
(6) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(7) After scan for group B is completed, a group B scan end interrupt request is generated if the ADCSR.GBADIE bit is 1 (group B scan end interrupt is enabled).
(8) The ADCSR.ADST bit is automatically cleared to 0 when scan of all selected channels is completed, and the 12-bit A/D converter enters a wait state.

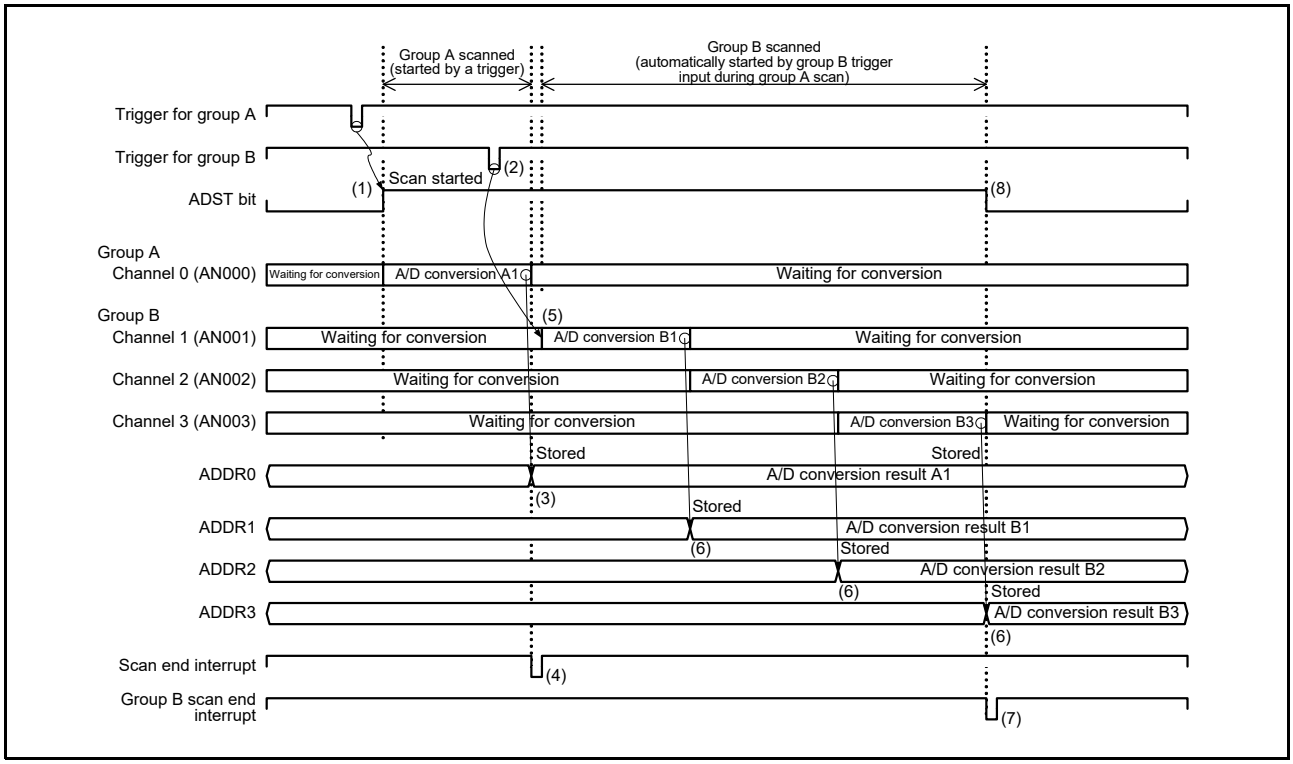


Figure 42.30 Example 3 of Group Priority Operation: Input of Group B Trigger Signal During Group A Scan, with the Rescan Setting (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, ADGSPCR.LGRRS = 0)

Operation example 4 shows group priority operation in group scan mode (ADGSPCR.GBRSCN = 0, ADGSPCR.GBRP = 0, ADGSPCR.LGRRS = 0) when channel 0 is selected for group A and channels 1 to 3 are selected for group B.

Operation Example 4 Input of the group A trigger signal during a group B scan, without the rescan setting

(1) When input of the group B trigger signal sets the ADCSR.ADST bit to 1 (starting A/D conversion), a group B scan starts for the analog channels selected in the ADANSB0 and ADANSB1 registers according to the order of conversion specified in the ADSCSn register.

(2) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).

(3) When a group A trigger is input during the group B scanning, the ADCSR.ADST bit retains the value as 1, and the group B scan is interrupted, and the group A scan starts for the analog channels selected in the ADANSA0 and ADANSA1 registers according to the order of conversion specified in the ADSCSn register.

(4) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).

(5) After scan for group A is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (scan end interrupt is enabled).

(6) The ADCSR.ADST bit is automatically cleared to 0 when scan for group A is completed, and the 12-bit A/D converter enters a wait state. Scan for group B is not started until the next group B trigger is input.

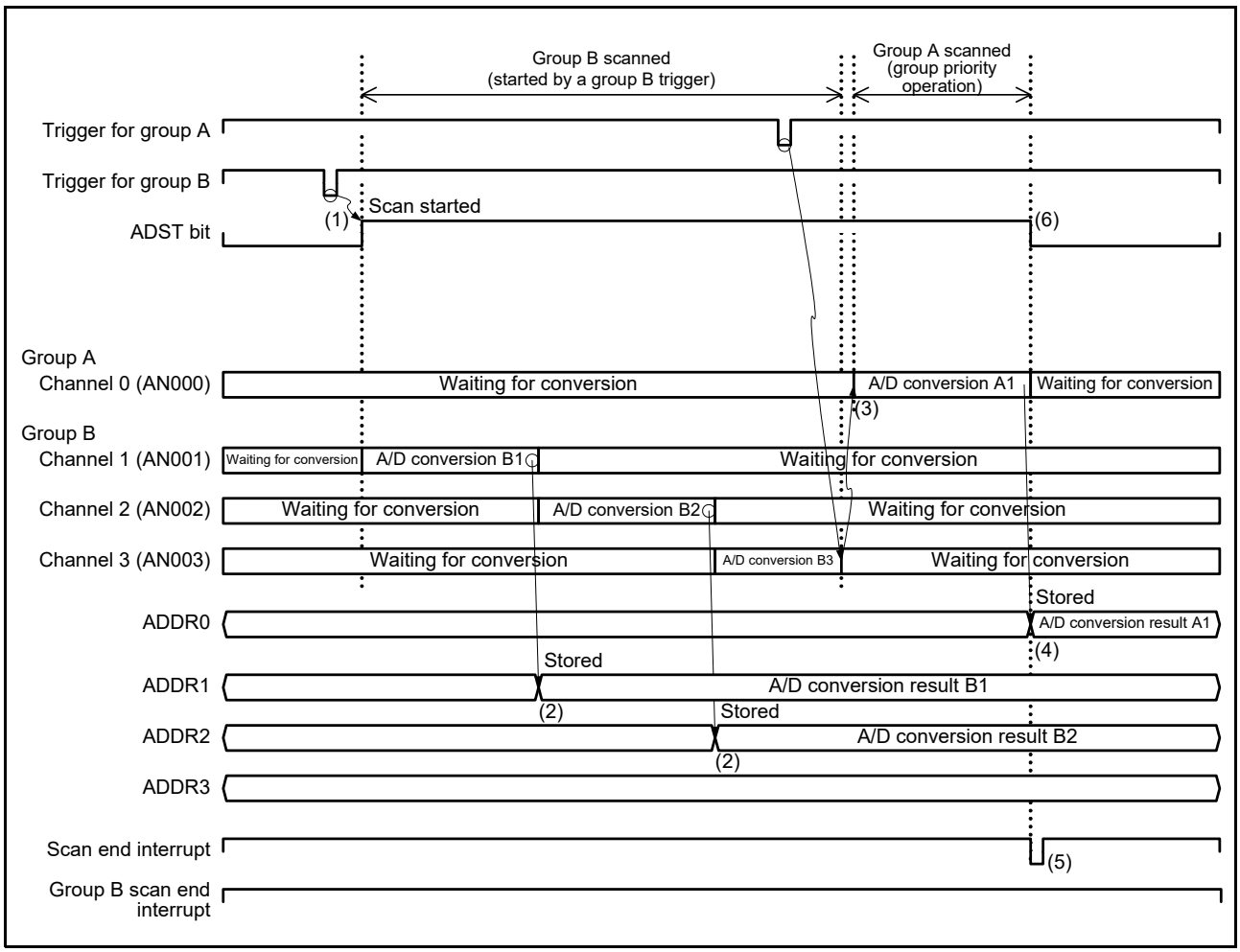


Figure 42.31 Example 4 of Group Priority Operation: Input of Group A Trigger Signal During Group B Scan, Without the Rescan Setting (ADGSPCR.GBRSCN = 0, ADGSPCR.GBRP = 0, ADGSPCR.LGRRS = 0)

Operation example 5 shows group priority operation in group scan mode (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 1, ADGSPCR.LGRRS = 0) when channel 0 is selected for group A and channels 1 and 2 are selected for group B. When the ADGCTRGR.GRCE bit is set to 1, single scan mode is continuously operated on group C and scan for group B is started by trigger input.

Operation example 5: Continuous single scan operation on group B

(1) When the ADGSPCR.GBRP bit is set to 1, the ADCSR.ADST bit is set to 1 (starting A/D conversion), and a group B scan is started for the analog channels selected in the ADANSB0 and ADANSB1 registers according to the order of conversion specified in the ADSCSn register.
(2) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(3) When a group A trigger is input during the group B scanning, the ADCSR.ADST bit retains the value as 1, and the group B scan is interrupted, and the group A scan starts for the analog channels selected in the ADANSA0 and ADANSA1 registers according to the order of conversion specified in the ADSCSn register.
(4) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(5) After scan for group A is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (scan end interrupt is enabled).
(6) When the ADGSPCR.GBRP bit is set to 1 (selecting continuous single scanning), the ADCSR.ADST bit retains the value as 1, and the group B scan for the analog channels selected in the ADANSB0 and ADANSB1 registers according to the order of conversion specified in the ADSCSn register is restarted.
(7) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(8) A group B scan end interrupt request is generated if the ADCSR.GBADIE bit is 1 (group B scan end interrupt is enabled).
(9) When the ADGSPCR.GBRP bit is set to 1 (selecting continuous single scanning), the ADCSR.ADST bit retains the value as 1, and the group B scan for the analog channels selected in the ADANSB0 and ADANSB1 registers according to the order of conversion specified in the ADSCSn register is restarted.

To continuously operate single scan for group B, disable trigger input for group B.
Steps (6) to (9) are repeated as long as the ADGSPCR.GBRP bit remains 1. Clearing of the ADCSR.ADST bit to 0 is prohibited while the ADGSPCR.GBRP bit is set to 1. To forcibly stop scanning when ADGSPCR.GBRP = 1, follow the procedures for clear operation by software through the ADCSR.ADST bit shown in section 42.6.2, Notes on Stopping A/D Conversion.

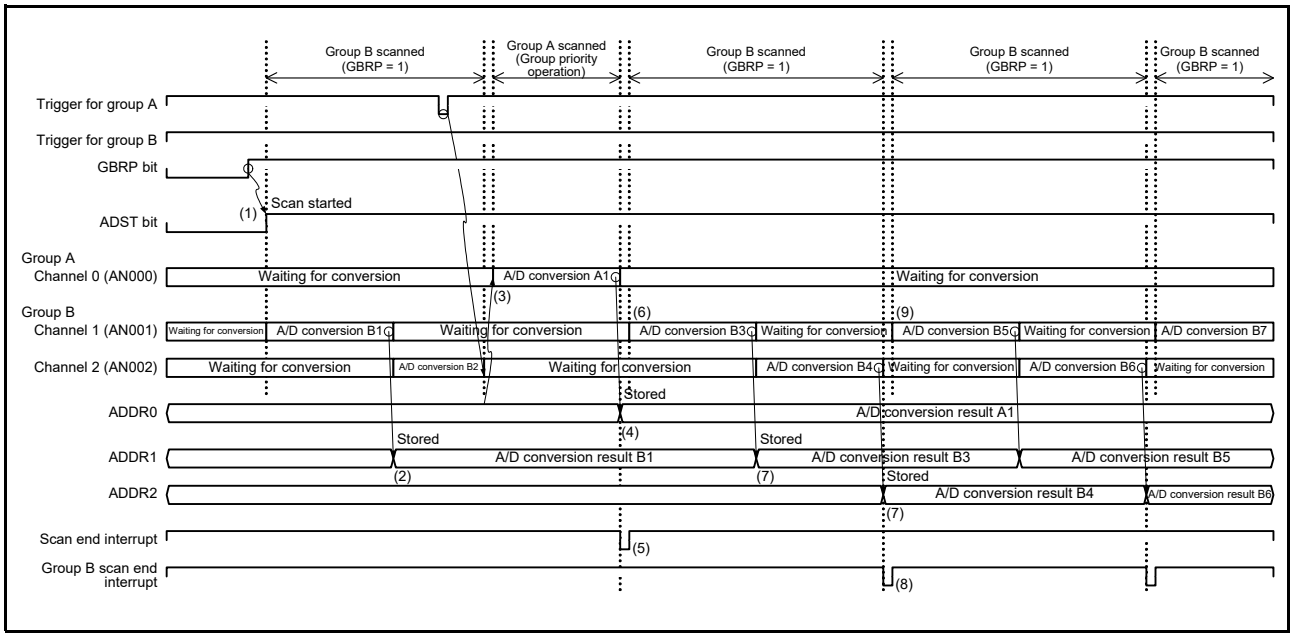


Figure 42.32 Example 5 for Group Priority Operation: Continuous Single Scan Operation on Group B (ADGSPCR.GBRP = 1, ADGSPCR.LGRRS = 1, ADGCTRGR.GRCE = 0)

(2) Group Priority Operation for Three Groups (ADGSPCR.PGS = 1, ADGCTRGR.GRCE = 1)

The following examples 1 to 5 show group priority operation in group scan mode (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, ADGSPCR.LGRRS = 1) when channel 0 is selected for group A, channels 1 and 2 are selected for group B, and channels 3 and 4 are selected for group C.
The priority groups mean groups A and B for group C and group A for group B.

Operation example 1: Priority group trigger input during low-priority group scan, with the rescan setting

(1) When input of the group C trigger signal sets the ADCSR.ADST bit to 1 (starting A/D conversion), the group C scan starts for the analog channels selected in the ADANSC0 and ADANSC1 registers according to the order of conversion specified in the ADSCSn register.
(2) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(3) When a group B trigger is input during the group C scanning, the ADCSR.ADST bit retains the value as 1, the group C scan is interrupted, and the group B scan starts for the analog channels selected in the ADANSB0 and ADANSB1 registers according to the order of conversion specified in the ADSCSn register. The A/D conversion result is not stored into the corresponding data register (ADDRy) if A/D conversion is not completed when the scan is interrupted.
(4) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(5) When a group A trigger is input during the group B scanning, the ADCSR.ADST bit retains the value as 1, and the group B scan is interrupted, and the group A scan starts for the analog channels selected in the ADANSA0 and ADANSA1 registers according to the order of conversion specified in the ADSCSn register. The A/D conversion result is not stored into the corresponding data register (ADDRy) if A/D conversion is not completed when the scan is interrupted.
(6) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(7) A scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (scan end interrupt is enabled).
(8) When the ADGSPCR.GBRSCN bit is set to 1 (restarting a group scan which was interrupted due to group priority in scanning), the ADCSR.ADST bit retains the value as 1, and the group B scan for the analog channels selected in the ADANSB0 and ADANSB1 registers according to the order of conversion specified in the ADSCSn register is restarted. At this time, the group B scan is restarted from the channel that the A/D conversion was interrupted if the ADGSPCR.LGRRS bit is set to 1.
(9) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(10) A group B scan end interrupt request is generated if the ADCSR.GBADIE bit is 1 (group B scan end interrupt is enabled).
(11) When the ADGSPCR.GBRSCN bit is set to 1, the ADCSR.ADST bit retains the value as 1, the group C scan for the analog channels selected in the ADANSC0 and ADANSC1 registers according to the order of conversion specified in the ADSCSn register is restarted. At this time, the group C scan is restarted from the channel that the A/D conversion was interrupted if the ADGSPCR.LGRRS bit is set to 1.
(12) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(13) A group C scan end interrupt is generated if the ADGCTRGR.GCADIE bit is 1 (group C scan end interrupt is enabled).
(14) The ADCSR.ADST bit is automatically cleared to 0 when scan of all selected channels is completed, and the 12-bit A/D converter enters a wait state.

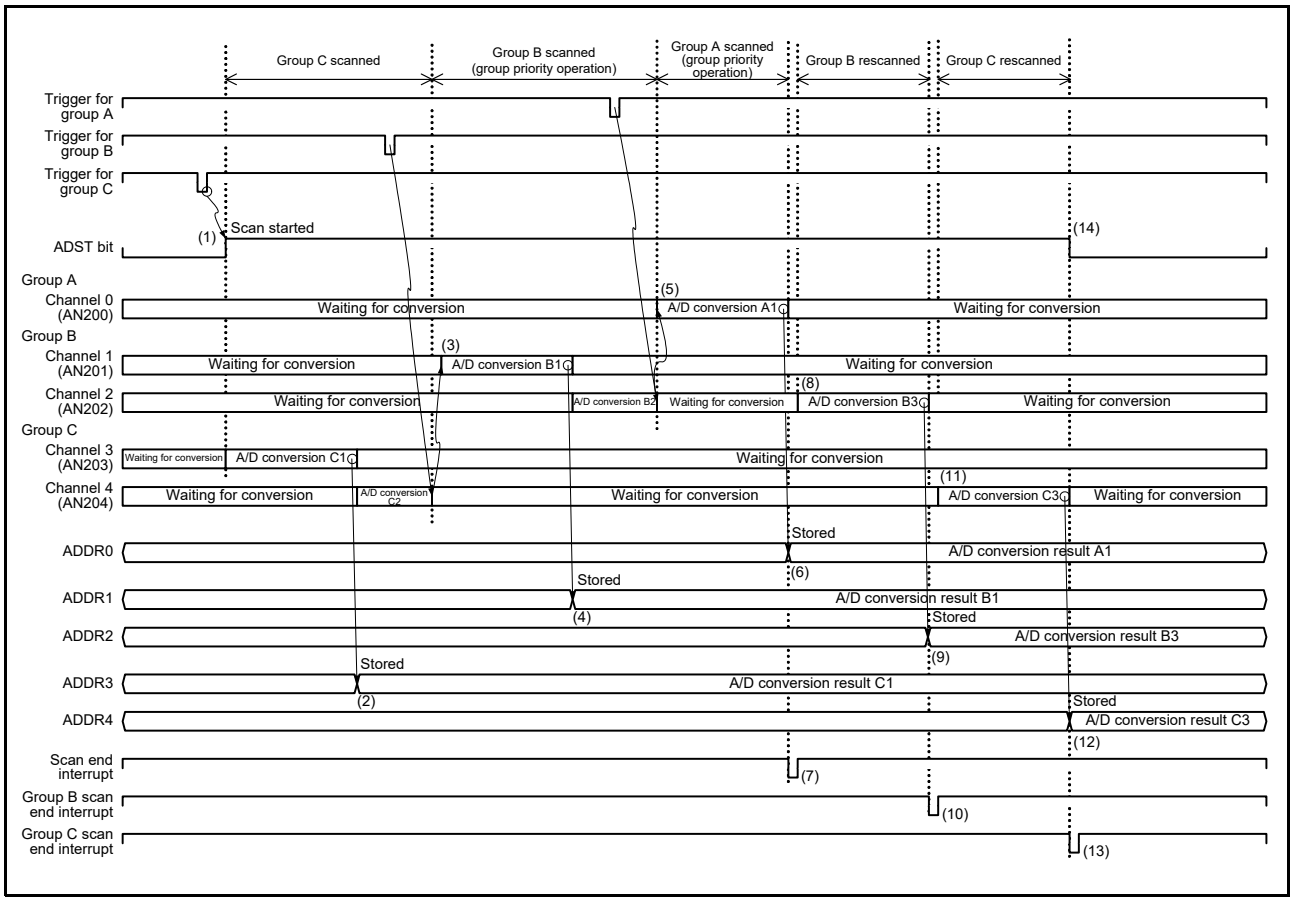


Figure 42.33 Example 1 of Group Priority Operation: Priority Group Trigger Input During Low-Priority Group Scan, with the Rescan Setting (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, ADGSPCR.LGRRS = 1)

Operation example 2: Priority group trigger input during low-priority group rescan, with the rescan setting

Figure 42.34 shows an example when a group A trigger is input during rescan operation on group B.
If a trigger for the priority groups (groups A and B for group C and group A for group B) is input, scan for the priority group starts even while rescan operation on the low-priority group is in progress. After scan for the priority group is completed, scan for the low-priority group is restarted after having been interrupted.
Operations of the ADCSR.ADST bit, storing to the A/D conversion result in the A/D data register (ADDRy), and interrupt requests are the same as example 1.

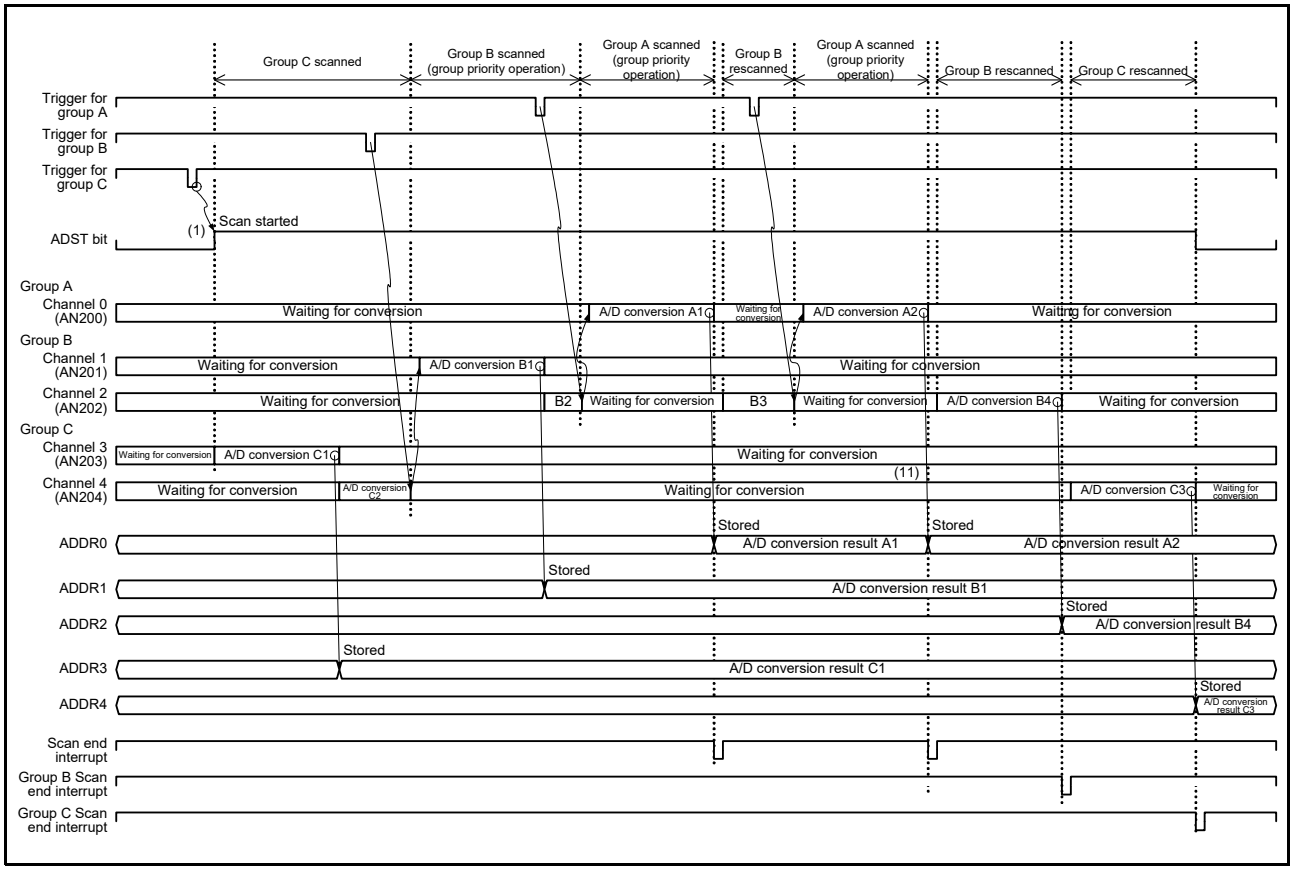


Figure 42.34 Example 2 of Group Priority Operation: Priority Group Trigger Input During Low-Priority Group Rescan, with the Rescan Setting (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, ADGSPCR.LGRRS = 1)

Operation example 3: Low-priority group trigger input during priority group scan, with the rescan setting

The following describes an example when a trigger for the low-priority group is input during scan operation on the priority group when the ADGPSCR.GBRSCN bit is 1 (scan for the group is restarted after having been interrupted due to group priority operation). If the ADGPSCR.GBRSCN bit is 0, all triggers for the low-priority group that are input during scan operation on the priority group are disabled.

(1) When input of the group A trigger signal sets the ADCSR.ADST bit to 1 (starting A/D conversion), a group A scan starts for the analog channels selected in the ADANSA0 and ADANSA1 registers according to the order of conversion specified in the ADSCSn register.
(2) If a group B trigger is input during scan for group A, scan for group B can be started.
(3) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(4) After scan for group A is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (scan end interrupt is enabled).
(5) After the group A scan ends, the ADCSR.ADST bit retains the value as 1, and the group B scan is started for the analog channels selected in the ADANSB0 and ADANSB1 registers according to the order of conversion specified in the ADSCSn register. At this time, the group B scan is restarted from the channel that the A/D conversion was suspended if the ADGSPCR.LGRRS bit is set to 1.
(When a group A trigger is input during the group B scanning, the group A scan is started in the same way as in the example 1, and the group B scan is restarted after the group A scan ends.)
(6) If a group C trigger is input during scan for group B, scan for group C can be started.
(7) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(8) After scan for group B is completed, a group B scan end interrupt request is generated if the ADCSR.GBADIE bit is 1 (group B scan end interrupt is enabled).
(9) After scan for group B is completed, scan for the ANx channels of group C selected in the ADANSC0 and ADANSC1 registers, starts from the channel with the smallest number x while the ADCSR.ADST bit remains 1. If the ADGSPCR.LGRRS bit is set to 1 at this time, scan for group C starts from the channel on which A/D conversion was interrupted.
(When a group A or B trigger is input during scan for group C, group A or B scan starts as in example 1, and group C scan starts after group A or B scan is completed.)
(10) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(11) After scan for group C is completed, a group C scan end interrupt request is generated if the ADGCTRGR.GCADIE bit is 1 (group C scan end interrupt is enabled).
(12) The ADCSR.ADST bit is automatically cleared to 0 when scan of all selected channels is completed, and the 12-bit A/D converter enters a wait state.

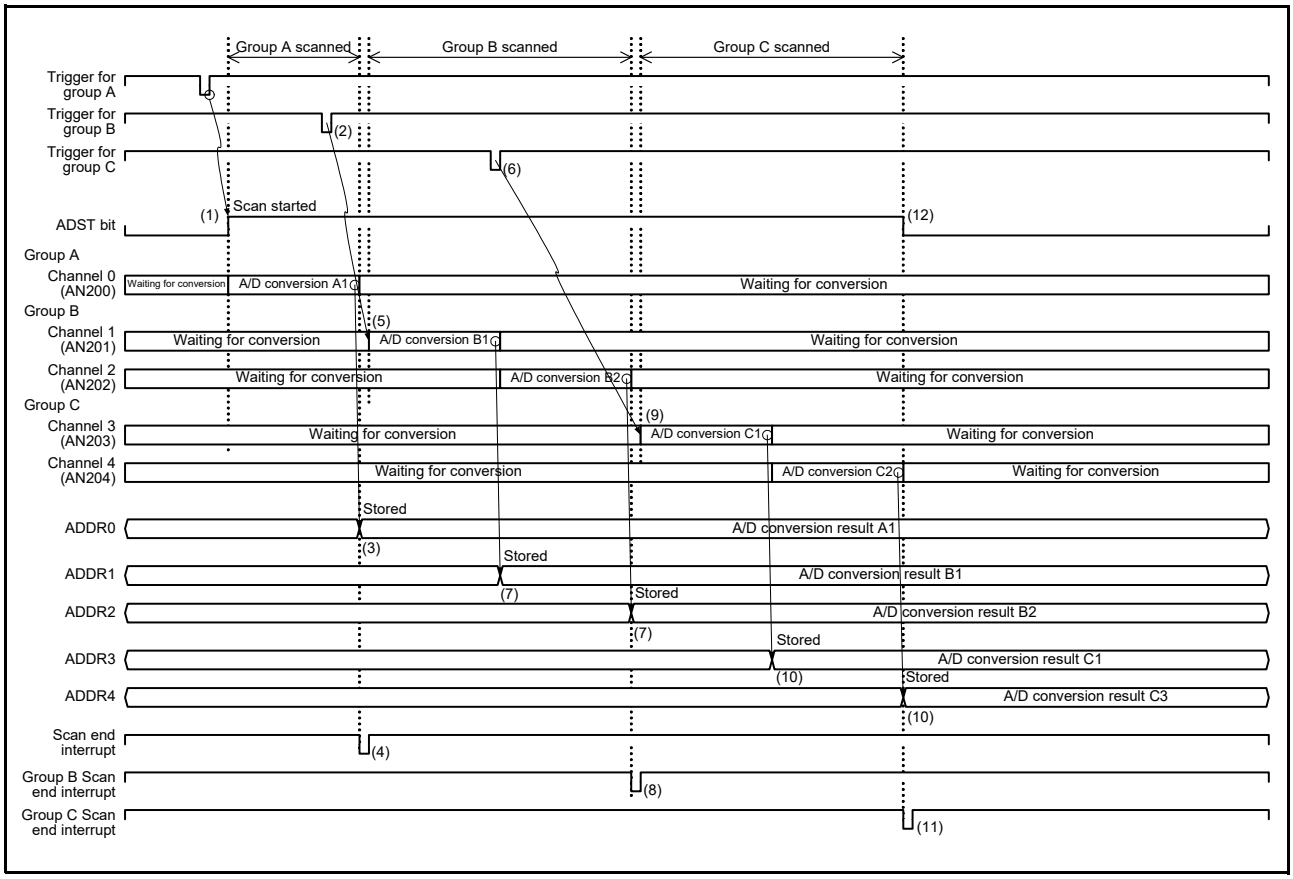


Figure 42.35 Example 3 of Group Priority Operation: Low-Priority Group Trigger Input During Priority Group Scan, with the Rescan Setting (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, ADGSPCR.LGRRS = 1, ADGCTRGR.GRCE = 1)

Operation example 4 shows group priority operation in group scan mode (ADGSPCR.GBRSCN = 0, ADGSPCR.GBRP = 0, ADGSPCR.LGRRS = 0) when channel 0 is selected for group A, channels 1 and 2 are selected for group B, and channels 3 and 4 are selected for group C.

Operation example 4: Priority group trigger input during low-priority group scan, without the rescan setting

(1) When input of the group C trigger signal sets the ADCSR.ADST bit to 1 (starting A/D conversion), the group C scan starts for the analog channels selected in the ADANSC0 and ADANSC1 registers according to the order of conversion specified in the ADSCSn register.

(2) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).

(3) When a group B trigger is input during the group C scanning, the ADCSR.ADST bit retains the value as 1, the group C scan is interrupted, and the group B scan starts for the analog channels selected in the ADANSB0 and ADANSB1 registers according to the order of conversion specified in the ADSCSn register. The A/D conversion result is not stored into the corresponding data register (ADDRy) if A/D conversion is not completed when the scan is interrupted.

(4) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).

(5) When a group A trigger is input during the group B scanning, the ADCSR.ADST bit retains the value as 1, and the group B scan is interrupted, and the group A scan starts for the analog channels selected in the ADANSA0 and ADANSA1 registers according to the order of conversion specified in the ADSCSn register. The A/D conversion

result is not stored into the corresponding data register (ADDRy) if A/D conversion is not completed when the scan is interrupted.

(6) After scan for group A is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (scan end interrupt is enabled).

(7) The ADCSR.ADST bit is automatically cleared to 0 when scan for group A is completed, and the 12-bit A/D converter enters a wait state. Scan for groups C and B is not started until the next trigger corresponding to the group is input.

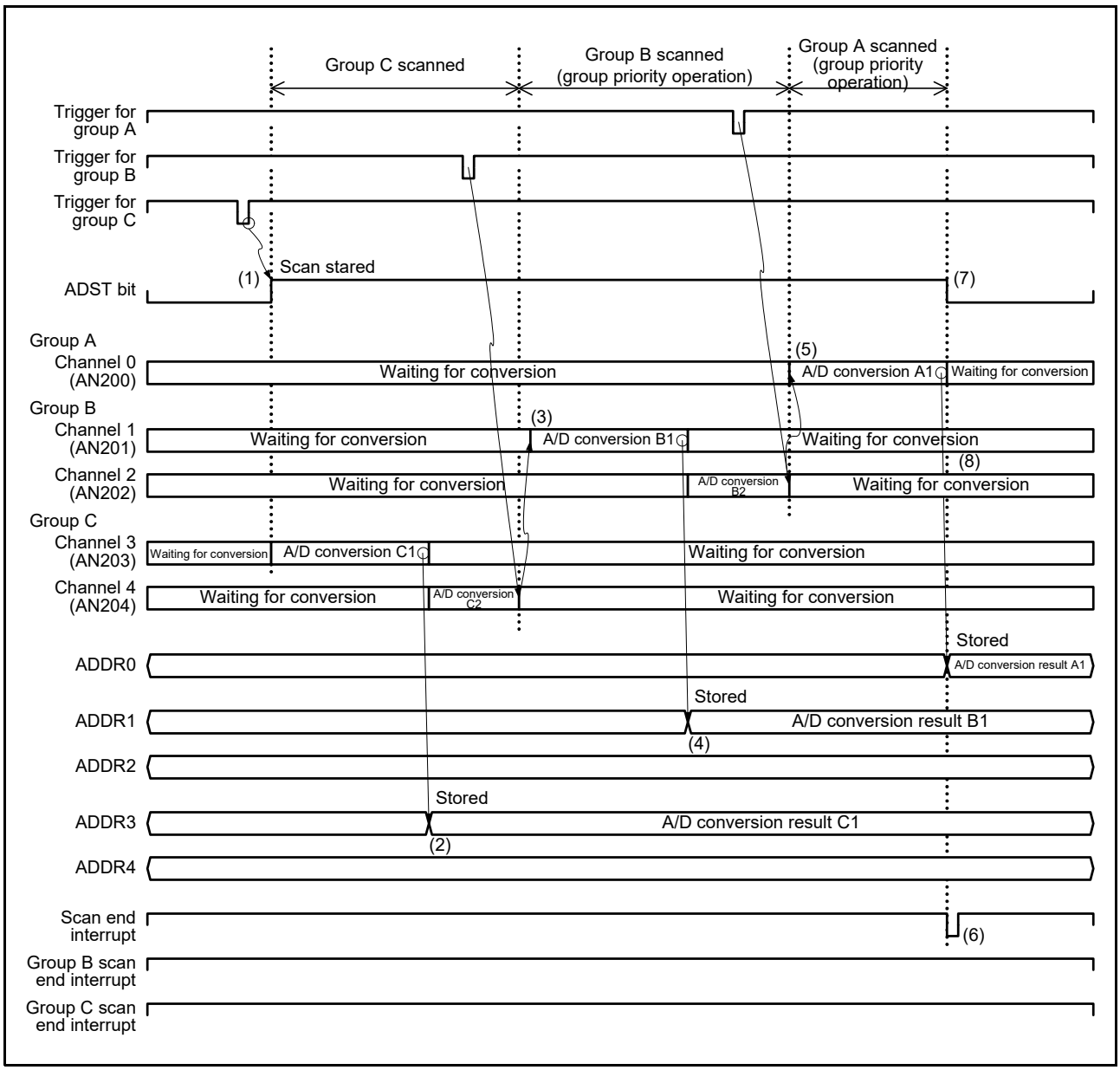


Figure 42.36 Example 4 of Group Priority Operation: Priority Group Trigger Input During Low-Priority Group Scan, Without the Rescan Setting (ADGSPCR.GBRSCN = 0, ADGSPCR.GBRP = 0, ADGSPCR.LGRRS = 1)

Operation example 5 shows group priority operation in group scan mode (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 1, ADGSPCR.LGRRS = 1) when channel 0 is selected for group A, channel 1 is selected for group B, and channels 2 and 3 are selected for group C.

When the ADGCTRGR.GRCE bit is set to 0, single scan mode is continuously operated on group B and trigger input for group C is disabled.

Operation example 5: Continuous single scan operation on group C

(1) When the ADGSPCR.GBRP bit is set to 1, the ADCSR.ADST bit is set to 1 (starting A/D conversion), and a group C scan is started for the analog channels selected in the ADANSC0 and ADANSC1 registers according to the order of conversion specified in the ADSCSn register.
(2) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(3) When a group B trigger is input during the group C scanning, the ADCSR.ADST bit retains the value as 1, the group C scan is interrupted, and the group B scan starts for the analog channels selected in the ADANSB0 and ADANSB1 registers according to the order of conversion specified in the ADSCSn register. The A/D conversion result is not stored into the corresponding data register (ADDRy) if A/D conversion is not completed when the scan is interrupted.
(4) When a group A trigger is input during the group B scanning, the ADCSR.ADST bit retains the value as 1, and the group B scan is interrupted, and the group A scan starts for the analog channels selected in the ADANSA0 and ADANSA1 registers according to the order of conversion specified in the ADSCSn register. The A/D conversion result is not stored into the corresponding data register (ADDRy) if A/D conversion is not completed when the scan is interrupted.
(5) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(6) A scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (scan end interrupt is enabled).
(7) When the ADGSPCR.GBRSCN bit is set to 1 (restarting a group scan which was interrupted due to group priority in scanning), the ADCSR.ADST bit retains the value as 1, and the group B scan for the analog channels selected in the ADANSB0 and ADANSB1 registers according to the order of conversion specified in the ADSCSn register is restarted. At this time, the group B scan is restarted from the channel that the A/D conversion was interrupted if the ADGSPCR.LGRRS bit is set to 1.
(8) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(9) A group B scan end interrupt request is generated if the ADCSR.GBADIE bit is 1 (group B scan end interrupt is enabled).
(10) When the ADGSPCR.GBRSCN bit is set to 1 (restarting a group scan which was interrupted due to group priority in scanning), the ADCSR.ADST bit retains the value as 1, and the group C scan for the analog channels selected in the ADANSC0 and ADANSC1 registers according to the order of conversion specified in the ADSCSn register is restarted. At this time, the group C scan is restarted from the channel that the A/D conversion was interrupted if the ADGSPCR.LGRRS bit is set to 1.
(11) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(12) A group C scan end interrupt is generated if the ADGCTRGR.GCADIE bit is 1 (group C scan end interrupt is enabled).
(13) When the ADGSPCR.GBRP bit is set to 1 (selecting continuous single scanning), the ADCSR.ADST bit retains the value as 1, and the group C scan for the analog channels selected in the ADANSC0 and ADANSC1 registers according to the order of conversion specified in the ADSCSn register is restarted.

To continuously operate single scan for group C, disable trigger input for group B.
Steps (13), (11), (12), and then (13) are repeated as long as the ADGSPCR.GBRP bit remains 1.
Clearing of the ADCSR.ADST bit to 0 is prohibited while the ADGSPCR.GBRP bit is set to 1.
To forcibly stop scanning when ADGSPCR.GBRP = 1, follow the procedures for clear operation by software through the ADCSR.ADST bit shown in section 42.6.2, Notes on Stopping A/D Conversion.

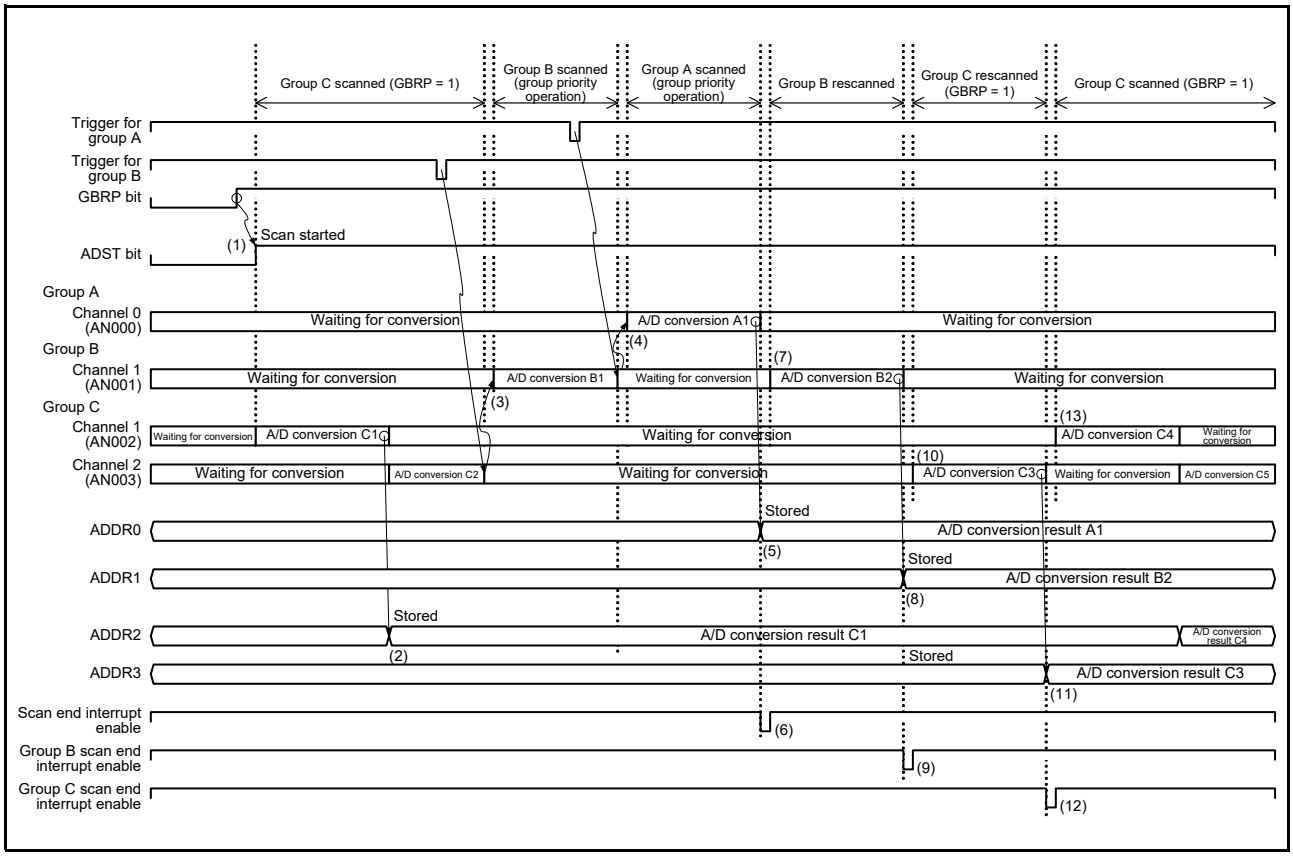


Figure 42.37 Example 5 for Group Priority Operation: Continuous Single Scan Operation on Group C (ADGSPCR.GBRP = 1, ADGSPCR.LGRRS = 1)

42.3.6 Comparison Function (Window A and Window B)

42.3.6.1 Comparison Function Windows A/B

The comparison function compares the reference values set in registers ADCMPDR0, ADCMPDR1, ADWINLLB, and ADWINULB with the A/D conversion results. When this function is used, self-diagnosis or double trigger mode cannot be used. A window comparison function can also be used (when ADCMPCR.WCMPE = 1), enabling comparison of two values. Two sets of voltage level ranges can be set in the window comparison function, one for window A and one for window B.

Operation when window comparison is enabled (ADCMPCR.WCMPE = 1) in combination with continuous scan mode is described below.

(1) When the ADCSR.ADST bit is set to 1 (starting A/D conversion) by software, or synchronous or asynchronous trigger input, A/D conversion is performed for selected channels, temperature sensor output, and internal reference voltage in this order.

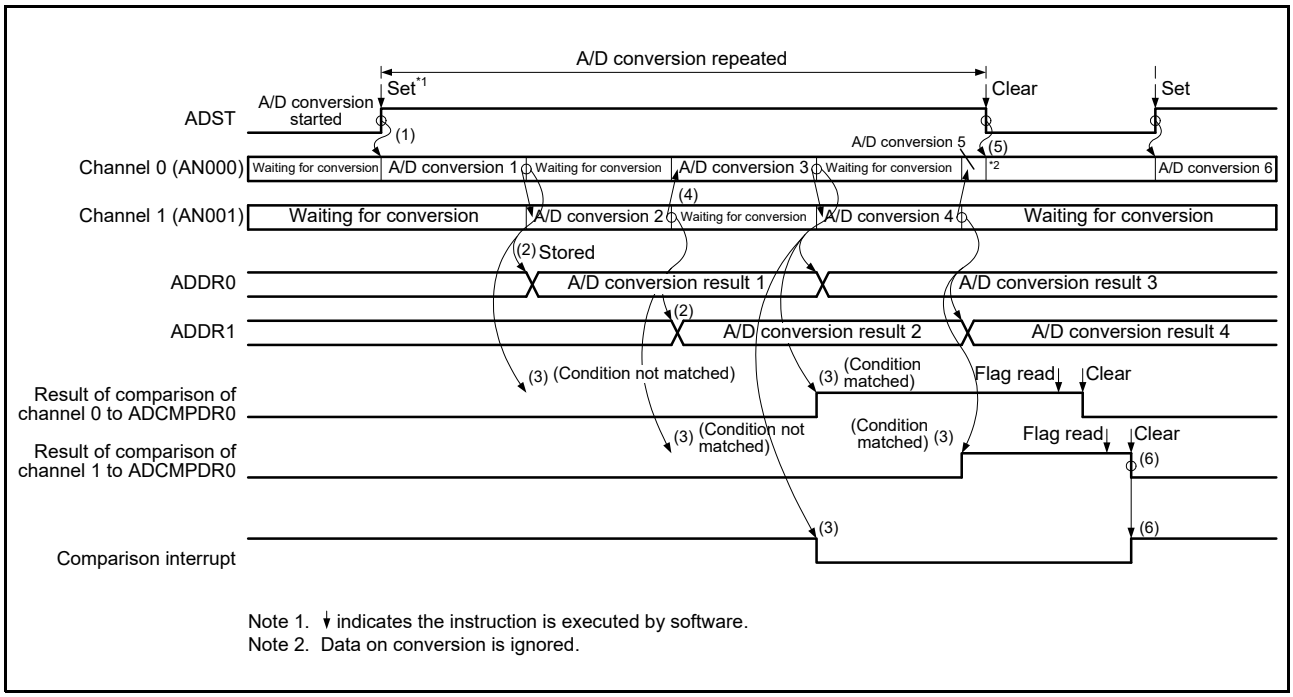
(2) Each time A/D conversion of a single channels is completed, the A/D-converted value is stored in the corresponding A/D data register (ADDRy, ADTSDR, or ADOCDR). When the ADCMPCR.CMPAE bit = 1 and window A is selected in the ADCMPANSRy and ADCMPANSER registers, the A/D-converted value is compared with the setting values of the ADCMPDR0 and ADCMPDR1 registers.
When the ADCMPCR.CMPBE bit = 1 and window B is selected in the ADCMPBNSR register, the A/D-converted value is compare with the setting values of the ADWINULB and ADWINLLB registers.

(3) For window A, upon a comparison match with the conditions specified in the ADCMPLR0, ADCMPLR1, and ADCMPLER registers, the flags of comparison window A (ADCMPSR0.CMPSTCHA0n, ADCMPSR1.CMPSTCHA1n, ADCMPSER.CMPFTS, and ADCMPSER.CMPFOC) become 1.
In this case, if the ADCMPCR.CMPAIE bit is set to 1, an interrupt request S12CMPAI is generated.
Similarly, for window B, upon a match with the conditions set in the ADCMPBNSR.CMPLB register, the comparison window B flag (ADCMPBSR.CMPSTB) becomes 1. In this case, if the ADCMPCR.CMPBIE bit is set to 1, an interrupt request S12CMPBI is generated.

(4) When all selected A/D conversion and comparison are completed, scanning is performed again.

(5) Set the ADCSR.ADST bit to 0 (stopping A/D conversion) and execute the processing for the channels in which the comparison flag is 1.

(6) After the processing above, clear all comparison flags. When comparison is re-executed, start A/D conversion again.



Note 1. ↓ indicates the instruction is executed by software.
Note 2. Data on conversion is ignored.

Figure 42.38 Example of Operation of Comparison (AN000 and AN001 for Targets for Comparison)

42.3.6.2 Restrictions on Comparison Function

The Comparison function has the following restrictions.

1. Use of self-diagnosis and double trigger mode is prohibited.
   (ADRD, ADDBLDR, ADDBLDRA, and ADDBLDRB are not targeted for the comparison function)
2. When using a matching or unmatching event, set to single scan mode.
3. When temperature sensor or internal reference voltage is selected for window A, operation of window B is prohibited.
4. When temperature sensor or internal reference voltage is selected for window B, operation of window A is prohibited.
5. The same channel cannot be set for window A and window B.
6. Satisfy the condition of [reference value (high side) ≥ reference value (low side)].

42.3.7 Analog Input Sampling Time and Scan Conversion Time

Figure 42.39 shows the scan conversion timing in single scan mode, in which scan conversion is started by software or a synchronous trigger. Figure 42.40 shows the scan conversion timing in single scan mode, in which scan conversion is started by an asynchronous trigger. The scan conversion time (t_{SCAN}) includes the start-of-scanning-delay time (t_D), channel-dedicated sample-and-hold circuit processing time (t_{SPLSH})*1, disconnection detection assistance processing time (t_{DIS})*2, auto-discharging processing time when A/D converting the temperature sensor output and the internal reference voltage (t_{ADIS}), self-diagnosis A/D conversion processing time (t_{DIAG})*3, A/D conversion processing time (t_{CONV}), channel-dedicated sample-and-hold circuit end time (t_{SHED})*4, and end-of-scanning-delay time (t_{ED}).
The A/D conversion processing time (t_{CONV}) consists of sampling time (t_{SPL}) and time for conversion by successive approximation (t_{SAM}). The sampling time (t_{SPL}) is used to charge sample-and-hold circuits in the A/D converter. If there is not sufficient sampling time due to the high impedance of an analog input signal source, or if the A/D conversion clock (ADCLK) is slow, sampling time can be adjusted using the ADSSTR register.
The time for conversion by successive approximation (t_{SAM}) is at 24 clock cycles of ADCLK. Table 42.23 shows the scan conversion time.

The scan conversion time (t_{SCAN}) in single scan mode for which the number of selected channels is n can be determined as follows:

$t_{SCAN} = t_D + t_{SPLSH} + (t_{DIS} \times n) + t_{DIAG} + (t_{CONV} \times n)$*5 $+ t_{ED}$

t_{SCAN} (when converting temperature sensor output and internal reference voltage) $= t_D + (t_{ADIS} \times m) + (t_{CONV} \times m) + t_{ED}$*6, *7, *8

The scan conversion time for the first cycle in continuous scan mode is t_{SCAN} for single scan minus t_{ED} plus t_{SHED}.
The scan conversion time for the second and subsequent cycles in continuous scan mode is fixed to $t_{SPLSH} + (t_{DIS} \times n) + t_{DIAG} + t_{DSD} + (t_{CONV} \times n)$*5 $+ t_{SHED}$.

Note 1. When no channel-dedicated sample-and-hold circuits are used, $t_{SPLSH} = 0$.
Note 2. When disconnection detection assistance is not selected, $t_{DIS} = 0$. When A/D-converting the temperature sensor output and the internal reference voltage, the value is fixed to 0Fh indicating 15 ADCLK.
Note 3. When the self-diagnosis function is not used, $t_{DIAG} = 0$, $t_{DSD} = 0$.
Note 4. When no channel-dedicated sample-and-hold circuits are used, $t_{SHED} = 0$. Here, continuous scan mode is assumed. In single scan mode and group scan mode, t_{SHED} is included in the end-of-scanning-delay time (t_{ED}).
Note 5. Although the total t_{CONV} can be expressed as $t_{CONV} \times n$ when the sampling times (t_{SPL}) of the selected channels are the same, it is basically expressed as the sum of the sampling times (t_{SPL}) for each of the channels and times for conversion by successive approximation (t_{SAM}) for each of the channels.
Note 6. Set m to 2 to perform A/D conversion both the temperature sensor output and internal reference voltage, or set m to 1 to perform A/D conversion either of them.
Note 7. When performing A/D conversion of the temperature sensor output or internal reference voltage, both the t_{SPLSH} and t_{DIAG} are 0, since A/D conversion of the self-diagnosis and analog channels cannot be selected in these cases.
Note 8. If the self-diagnosis and analog channels are in one group and the temperature sensor output and internal reference are in another, A/D conversion for all of them can be performed for a group scan.

Table 42.23 Times for Conversion During Scanning (in Numbers of Cycles of ADCLK and PCLKB)

Item			Symbol		Type/Conditions: Synchronous Trigger*3	Asynchronous Trigger	Software Trigger	Unit
Scan start processing time*1, *2	A/D conversion on group under group priority control.	The low-priority group is to be stopped. (The priority group is started after low-priority group B is stopped due to an A/D conversion start trigger of the priority group.)	t_D		2 PCLKB + 6 ADCLK (5 PCLKB + 3 ADCLK)*4	—	—	Cycle
		The low-priority group is not to be stopped. (Started by an A/D conversion start trigger of the priority group.)			2 PCLKB + 4 ADCLK	—	—	
	A/D conversion when self-diagnosis is enabled	A/D conversion for self-diagnosis is to be started.			2 PCLKB + 6 ADCLK	4 PCLKB + 6 ADCLK	6 ADCLK	
	Other than above				2 PCLKB + 4 ADCLK	4 PCLKB + 4 ADCLK	4 ADCLK	
Channel-dedicated sample-and-hold processing time*1	Sampling time		t_{SPLSH}	t_{SH}	When constant sampling is disabled: Setting values of the ADSHCR.SSTSH[7:0] bits (initial value:1Bh) × ADCLK When constant sampling is enabled: 0			
	Wait time between sampling and A/D conversion			t_W	12 ADCLK			
Disconnection detection assistance processing time			t_{DIS}		The setting of ADNDIS[3:0] (initial value = 00h) × ADCLK*5			
Auto-discharging processing time (Time required at the conversion of the temperature sensor output and the internal reference voltage)			t_{ADIS}		15 ADCLK			
Self-diagnosis conversion processing time*1	Sampling time		t_{DIAG}	t_{SPL}	The setting of ADSSTR0 (initial value = 1Bh) × ADCLK			
	Time for conversion by successive approximation			t_{SAM}	24 ADCLK			
	Normal A/D conversion is to be started after completion of self-diagnosis conversion.		t_{DED}		2 ADCLK			
	A/D conversion for self-diagnosis is to be started after completion of conversion for continuous scan on the last channel specified.		t_{DED}		2 ADCLK			
A/D conversion processing time*1	Sampling time		t_{CONV}	t_{SPL}	The setting of ADSSTRn (n = 0 to 11, L, T, O) (initial value = 1Bh) × ADCLK			
	Time for conversion by successive approximation			t_{SAM}	24 ADCLK			
Channel-dedicated sample-and-hold end processing time			t_{SHED}		2 ADCLK			
Scan end processing time*1			t_{ED}		1 PCLKB + 3 ADCLK (2 PCLKB + 2 ADCLK)*4			

Note 1. For t_D, t_{SPLSH}, t_{DIAG}, t_{CONV}, and t_{ED}, refer to Figure 42.39 and Figure 42.40.
Note 2. This is the maximum time required from software writing or trigger input to start A/D conversion.
Note 3. Time consumed on the paths from timer output to trigger input is not included.
Note 4. Maximum time for when ADCLK is faster than PCLKB (PCLKB to ADCLK frequency ratio = 1:2)
Note 5. When A/D-converting the temperature sensor output and the internal reference voltage, the value is fixed to 0Fh indicating 15 ADCLK.

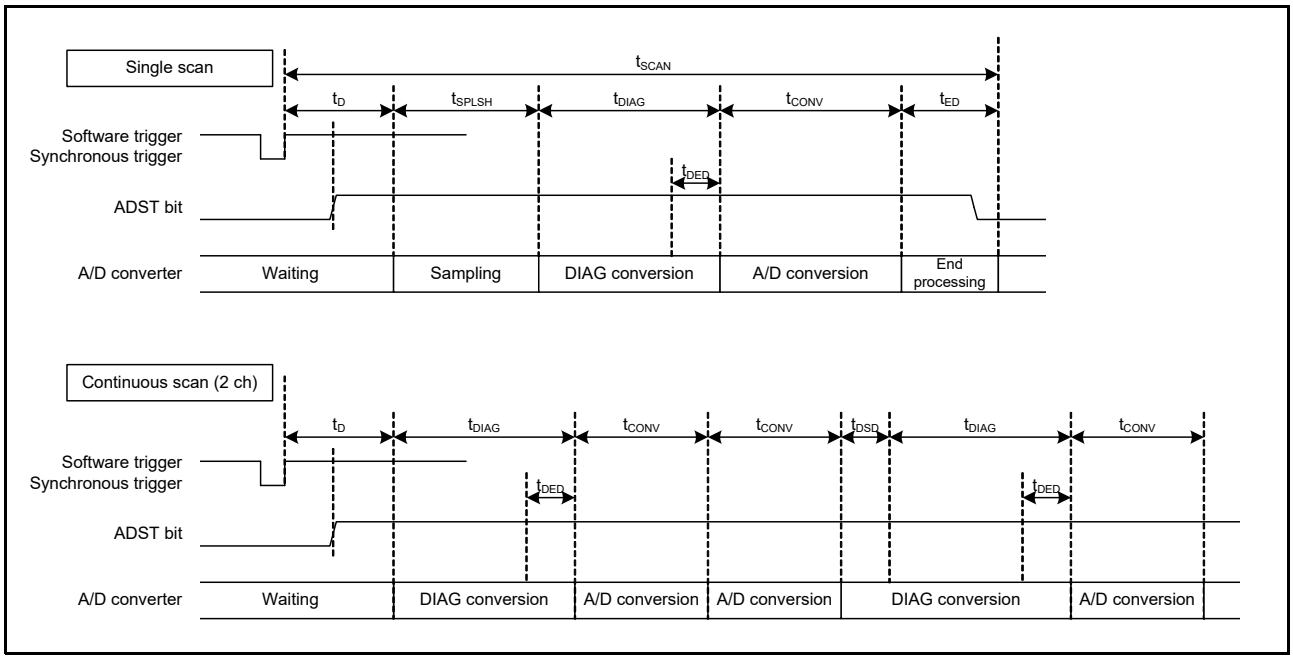


Figure 42.39 Scan Conversion Timing (Started by Software or Synchronous Trigger)

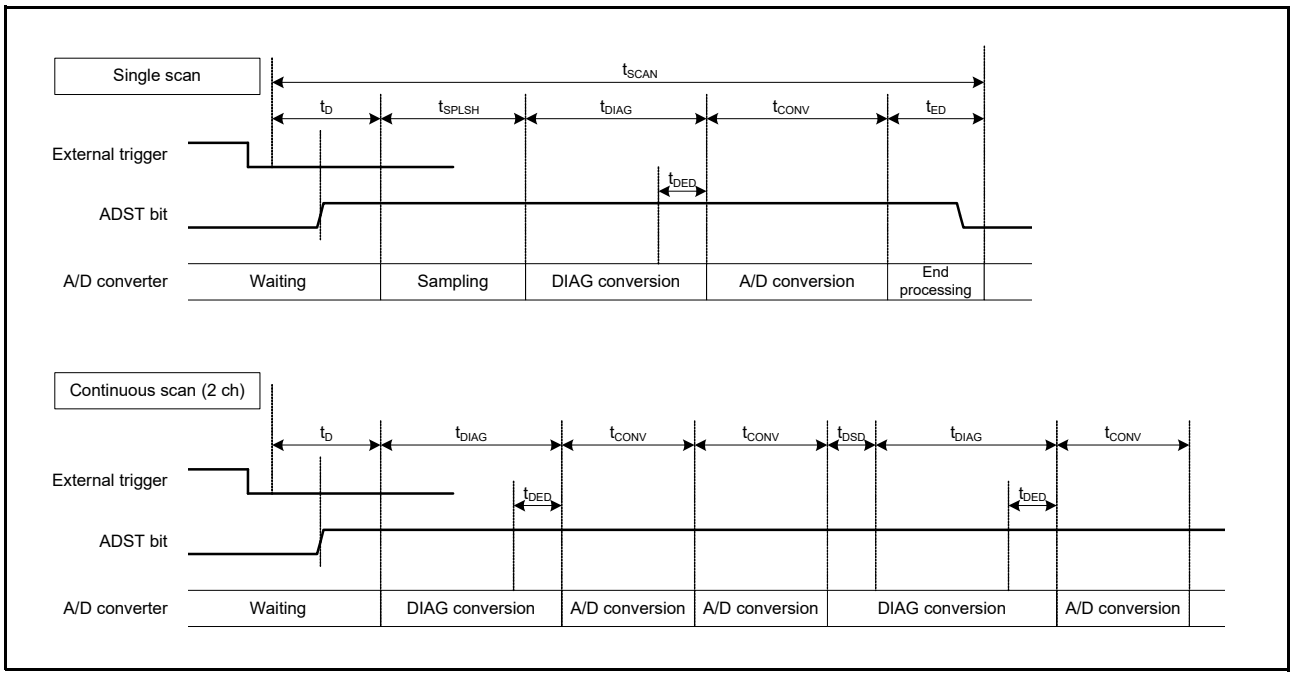


Figure 42.40 Scan Conversion Timing (Started by Software or Asynchronous Trigger)

42.3.7.1 Timing of Suspension and Starting of Scanning in Operation under Group Priority Control

The timings for suspension and starting of scanning in operation under group priority control that must considered are listed below.

1. The timing for suspending a scan of a group with a lower-priority and the timing for starting a scan of a group with a higher-priority.
2. The time at which scanning by the group with a lower-priority is resumed on completion of scanning by the higher-priority group when the trigger for scanning by the lower-priority group is accepted during scanning by the higher-priority group.
3. The timing for performing sequential single scans by a lower-priority group.

Figure 42.41 shows the timing diagram of each of the above cases.
The times at which scanning by group A is completed and scanning by group C resumes or scanning by group B is completed and scanning by group C resumes are the same as those for group A and group B in Figure 42.41.
The timing for consecutive single scans is the same for group B and group C.

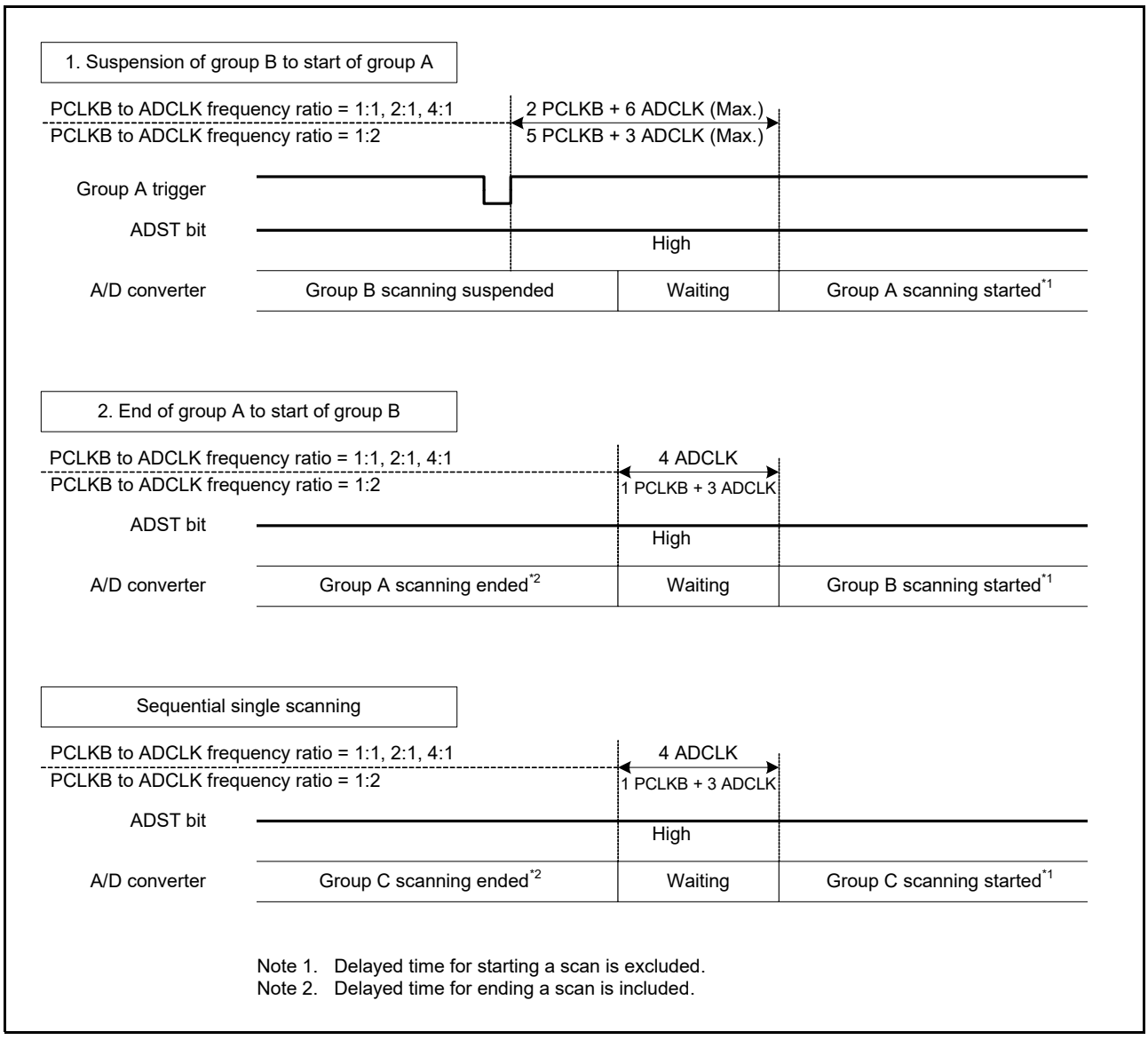


Figure 42.41 Timing of Stop/Start of Scanning in Group Priority Mode

42.3.8 Usage Example of A/D Data Register Automatic Clearing Function

Setting the ADCER.ACE bit to 1 automatically clears the A/D data registers (ADDRy, ADRD, ADTSDR, ADOCDR, ADDBLDR, ADDBLDRA, and ADDBLDRB) to 0000h when the A/D data registers (ADDRy, ADRD, ADTSDR, ADOCDR, ADDBLDR, ADDBLDRA, and ADDBLDRB) are read by the CPU, DTC, or DMAC.
This function enables detection of update failures of the A/D data registers (ADDRy, ADRD, ADTSDR, ADOCDR, ADDBLDR, ADDBLDRA, and ADDBLDRB). The following describes the examples in which the function to automatically clear the ADDRy register is enabled and disabled.
In a case where the ADCER.ACE bit is 0 (automatic clearing disabled), if the A/D conversion result (0222h) is not written to the ADDRy register for some reason, the old data (0111h) will be the ADDRy register value. Furthermore, if this ADDRy register value is read into a general register using an A/D scan end interrupt, the old data (0111h) can be saved in the general register. When checking whether there is an update failure, it is necessary to frequently save the old data in the RAM or a general register.
In a case where the ADCER.ACE bit is 1 (automatic clearing enabled), when ADDRy = 0111h is read by the CPU, DTC, or DMAC, the ADDRy register is automatically cleared to 0000h. After that, if the A/D conversion result (0222h) cannot be transferred to the ADDRy register for some reason, the cleared data (0000h) remains as the ADDRy register value. If this ADDRy register value is read into a general register using an A/D scan end interrupt at this point, 0000h will be saved in the general register.
Occurrence of an ADDRy register update failure can be determined by simply checking that the read data value is 0000h.

42.3.9 A/D-Converted Value Addition/Average Mode

In A/D-converted value addition mode, the same channel is A/D-converted 2, 3, 4, or 16 consecutive times and the sum of the converted values is stored in the data register. In A/D-converted value average mode, the same channel is A/D-converted two or four consecutive times and the mean of the converted values is stored in the data register. The use of the average of these results can improve the accuracy of A/D conversion, depending on the types of noise components that are present. This function, however, cannot always guarantee an improvement in A/D conversion accuracy.
The A/D-converted value addition/average mode can be used when A/D conversion of the channel select analog input, temperature sensor output (for unit 2 only), or internal reference voltage (for unit 2 only) is selected.

42.3.10 Disconnection Detection Assist Function

This converter incorporates the function to fix the charge for sampling capacitance to the specified state before start of A/D conversion. This function enables disconnection detection in wiring of analog inputs.
Figure 42.42 illustrates the A/D conversion operation when the disconnection detection assist function is used. Figure 42.43 shows an example of disconnection detection when precharge is selected. Figure 42.44 shows an example of disconnection detection when discharge is selected.

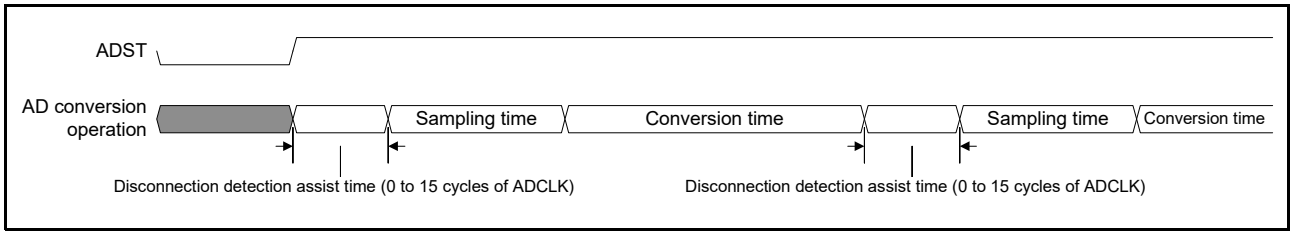
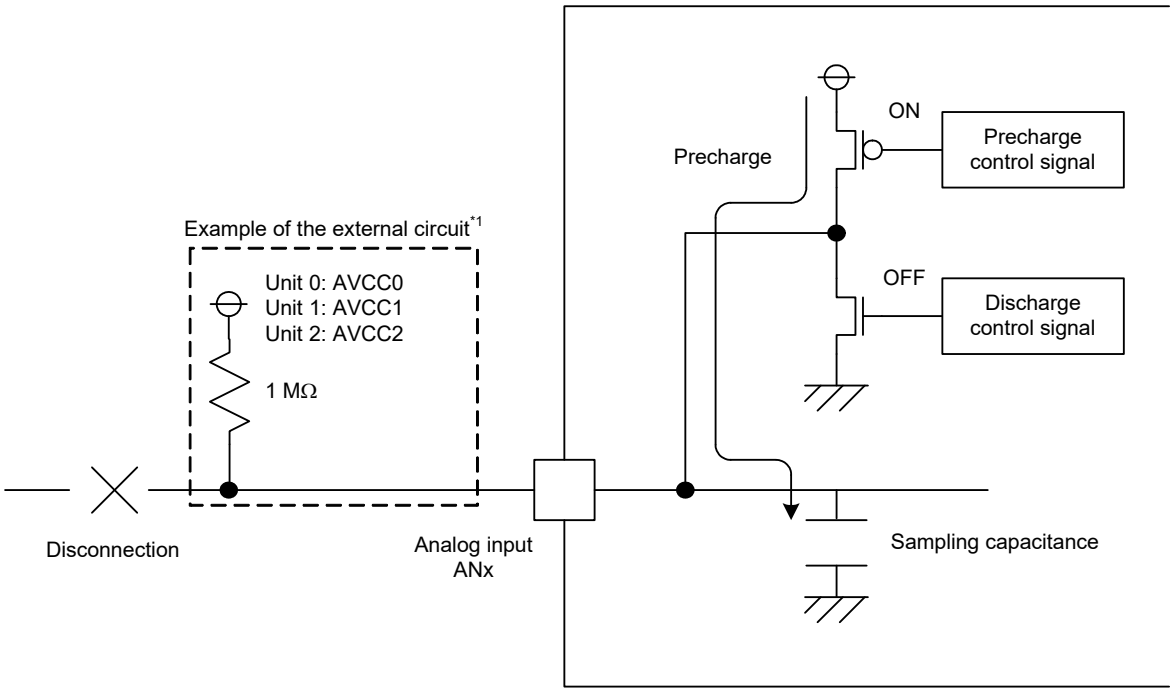
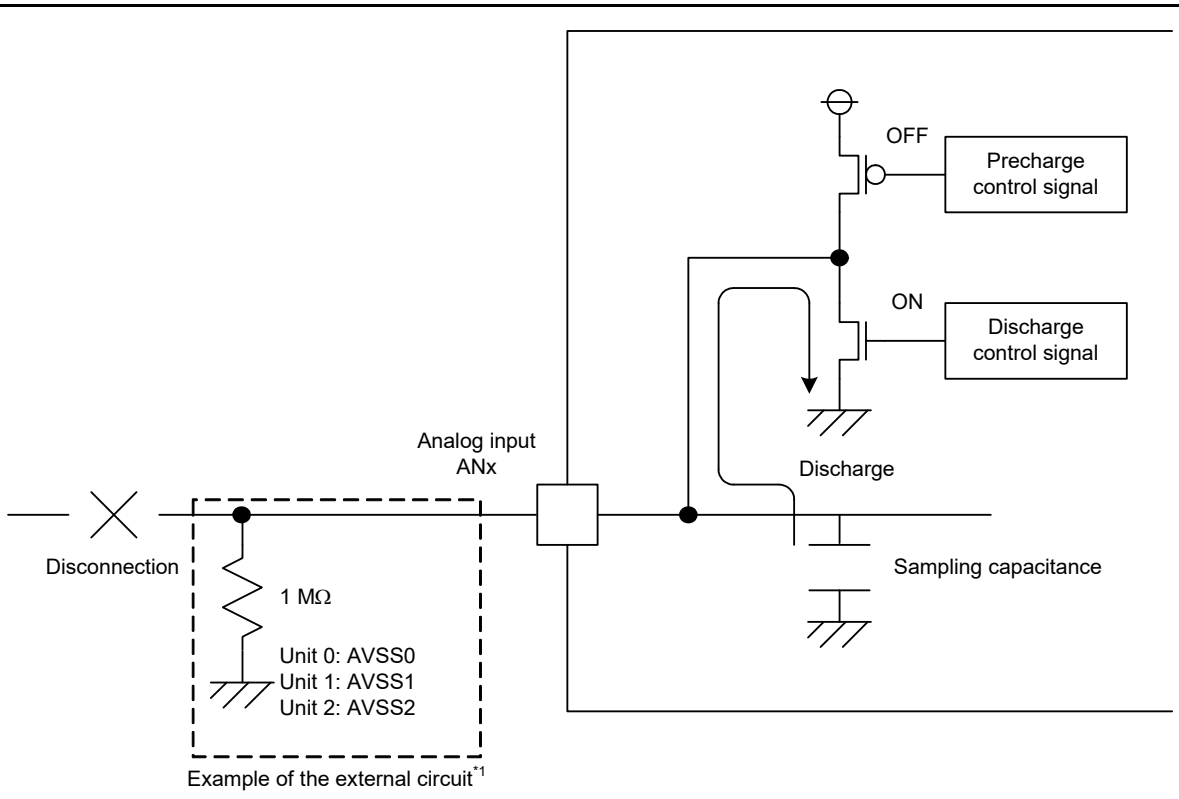


Figure 42.42 Operation of A/D Conversion When the Disconnection Detection Assist Function is Used



Note 1. Use the results of conversion after fully evaluating them because the resulting values when disconnection occurs depend on the external circuits.

Figure 42.43 Example of Disconnection Detection When Precharge is Selected



Note 1. Use the results of conversion after fully evaluating them because the resulting values when disconnection occurs depend on the external circuits.

Figure 42.44 Example of Disconnection Detection When Discharge is Selected

42.3.11 Starting A/D Conversion with Asynchronous Trigger

The A/D conversion can be started by the input of an asynchronous trigger. To start up the A/D converter by an asynchronous trigger, the A/D conversion start trigger select bits (ADSTRGR.TRSA[6:0]) should be set to 0000000b, and a high-level signal should be input to the asynchronous trigger (ADTRG0# pin). Then, the ADCSR.TRGE and ADCSR.EXTRG bits should be set to 1. Figure 42.45 shows a timing of the asynchronous trigger input.
For the time from when the ADST bit is set to 1 until conversion starts, refer to section 42.6.3, A/D Conversion Restarting Timing and Termination Timing. An asynchronous trigger cannot be selected for group B or group C in group scan mode.

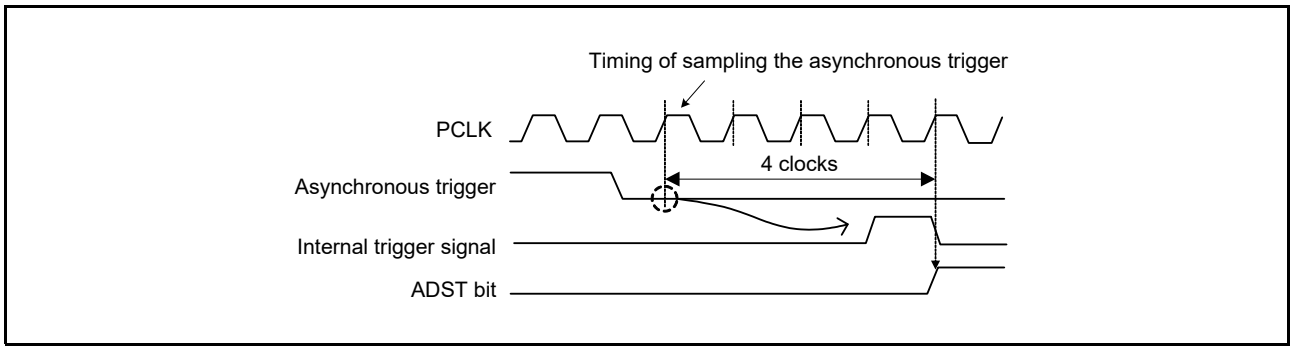


Figure 42.45 Timing of the Asynchronous Trigger Input

42.3.12 Starting A/D Conversion with Synchronous Trigger from Peripheral Module

The A/D conversion can be started by a synchronous trigger. To start the A/D conversion by a synchronous trigger, the ADCSR.TRGE bit should be set to 1, the ADCSR.EXTRG bit should be cleared to 0, and the relevant sources should be selected by the ADSTRGR.TRSA[6:0] bits, the ADSTRGR.TRSB[6:0] bits, and the ADGCTRGR2.TRSC6 and ADGCTRGR.TRSC[5:0 bits.

42.3.13 Conversion Function in the Order of Arbitrary Channel Number

Conversion function in the order of arbitrary channel number is used to convert analog channels of each unit in the 12-bit A/D converter in the order set by the ADSCSn register.

Setting the same channel in the ADSCSn register is prohibited. Channels whose order can be set with this function are analog channels only.

The conversion order for the self-diagnosis, temperature sensor, and internal reference voltage cannot be changed.

Table 42.24 Example of Conversion Order When the ADSCSn Register (n = 0 to 4) is Set to the Initial Value in the Scan Group A

ADSCSn Setting Value		A/D Conversion Order			ADANSA0 Setting Value
Symbol	Initial Value	Physical Channel	Priority	Conversion Order	Channel Select Bit
ADSCS0	00h	AN000	High	1	ANSA0[0] =1
ADSCS1	01h	AN001	↓	2	ANSA0[1] =1
ADSCS2	02h	AN002		3	ANSA0[2] =1
ADSCS3	03h	AN003		4	ANSA0[3] =1
ADSCS4	04h	AN004	Low	5	ANSA0[4] =1

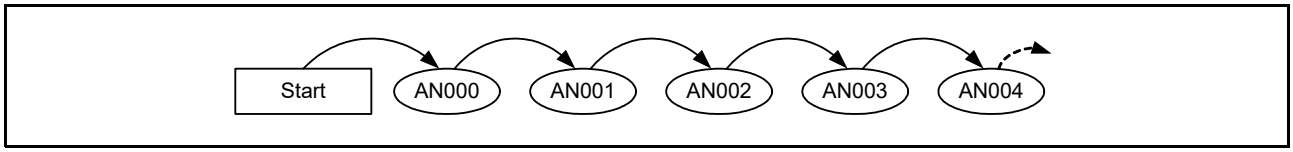


Figure 42.46 Example of Conversion Order (Setting Based on Table 42.24)

Table 42.25 Example of Conversion Order When a Setting Value of the ADSCSn Register (n = 0 to 4) is Modified in the Scan Group A

ADSCSn Setting Value		A/D Conversion Order			ADANSA0 Setting Value
Symbol	Value Set by User	Physical Channel	Priority	Conversion Order	Channel Select Bit
ADSCS0	04h	AN004	High	1	ANSA0[4] =1
ADSCS1	02h	AN002	↓	2	ANSA0[2] =1
ADSCS2	00h	AN000		3	ANSA0[0] =1
ADSCS3	01h	AN001		4	ANSA0[1] =1
ADSCS4	03h	AN003	Low	5	ANSA0[3] =1

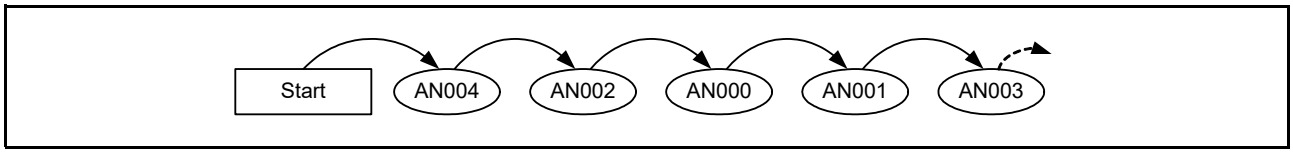


Figure 42.47 Example of Conversion Order (Setting Based on Table 42.25)

Table 42.26 Example of Conversion Order When a Setting Value of the ADSCSn Register (n = 0 to 4) is Modified in the Scan Group A (When the ANSA0[1] Bit is Set to Deselect.)

ADSCSn Setting Value		A/D Conversion Order			ADANSA0 Setting Value
Symbol	Value Set by User	Physical Channel	Priority	Conversion Order	Channel Select Bit
ADSCS0	04h	AN004	High	1	ANSA0[4] =1
ADSCS1	02h	AN002	↓	2	ANSA0[2] =1
ADSCS2	00h	AN000		3	ANSA0[0] =1
ADSCS3	01h	AN001		4	ANSA0[1] =0
ADSCS4	03h	AN003	Low	5	ANSA0[3] =1

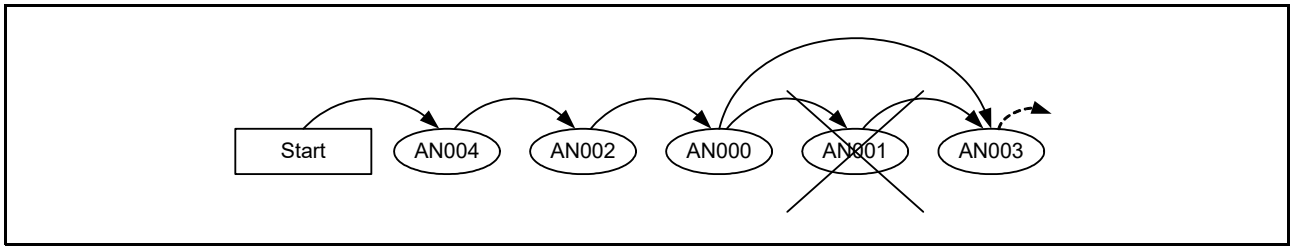


Figure 42.48 Example of Conversion Order (Setting Based on Table 42.26)

42.3.14 Internal Reference Voltage Monitoring Function

The internal reference voltage monitoring function is used to output the internal reference voltage to the A/D converter. Set the A/D internal reference voltage monitoring circuit enable register (ADVMONCR) and the A/D internal reference voltage monitoring circuit output enable register (ADVMONO), and enable the internal reference voltage A/D select bit (OCSA/B/C) to perform A/D conversion of the internal reference voltage.

Example of the operation is shown below.

(1) Set the VDE bit in the A/D internal reference voltage monitoring circuit enable register (ADVMONCR) to 1.
(2) Set the VDO bit in the A/D internal reference voltage monitoring circuit output enable register (ADVMONO) to 1.
(3) Select the internal reference voltage either by the internal reference voltage A/D conversion select bit (OCSA) in the A/D conversion extended input control register, the group B internal reference voltage A/D conversion select bit (OCSB), or the group C internal reference voltage A/D conversion select bit (OCSC) in the A/D group C extended input control register (ADGCEXCR).
(4) After inserting the waiting period of 500 ns, the ADCSR.ADST bit becomes 1 (starting A/D conversion) either by software, synchronous trigger (MTU, TMR, and ELC), or asynchronous trigger, and then, auto-discharging of the internal reference voltage is performed and A/D conversion of the internal reference voltage starts.
(5) When A/D conversion of the internal reference voltage is completed, A/D conversion result is stored to the corresponding A/D internal reference voltage data register (ADOCDR).
(6) S12ADI2 interrupt request is generated (pulse output and level output) if the ADCSR.ADST bit is set to 1 (scan end interrupt is enabled).
(7) The ADCSR.ADST bit is automatically cleared upon completion of A/D conversion, and the 12-bit A/D converter enters a wait state.
(8) Set the VDO bit in the A/D internal reference voltage monitoring circuit output enable register (ADVMONO) to 0.
(9) Set the VDE bit in the A/D internal reference voltage monitoring circuit enable register (ADVMONCR) to 0.

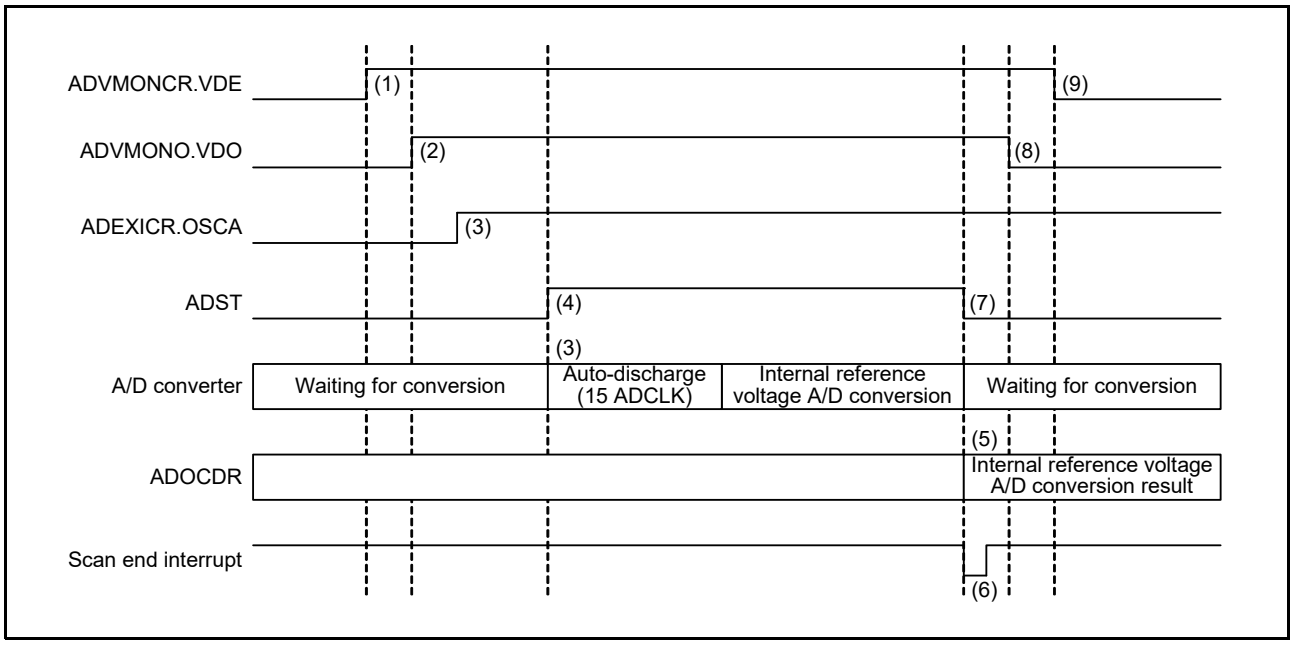


Figure 42.49 Example of Internal Reference Voltage Monitor Output A/D Conversion

42.4 Interrupt Sources and DTC/DMA Transfer Requests

42.4.1 Interrupt Requests

The 12-bit A/D converter can generate scan end interrupt requests S12ADI/S12ADI1/S12ADI2, S12GBADI/S12GBADI1/S12GBADI2, and S12GCADI/S12GCADI1/S12GCADI2 to the CPU.
The module also can generate the S12CMPAI or S12CMPBI interrupt which is the request of interrupting to the CPU in response to matches with a condition for comparison.
Setting the ADCSR.ADIE bit to 1 and 0 enables and disables an S12ADI/S12ADI1/S12ADI2 interrupt, respectively; similarly, setting the ADCSR.GBADIE bit to 1 and 0 enables and disables a S12GBADI/S12GBADI1/S12GBADI2 interrupt, respectively.
Setting the ADCMPCR.CMPAIE bit to 1 enables an S12CMPAI interrupt and setting the ADCMPCR.CMPAIE bit to 0 disables an S12CMPAI interrupt.
Setting the ADCMPCR.CMPBIE bit to 1 enables an S12CMPBI interrupt and setting the ADCMPCR.CMPBIE bit to 0 disables an S12CMPBI interrupt.
In addition, the DTC/DMA transfer can be triggered when an S12ADI/S12ADI1/S12ADI2, S12GBADI/S12GBADI1/S12GBADI2, or S12GCADI/S12GCADI1/S12GCADI2 interrupt is generated. Using an S12ADI/S12ADI1/S12ADI2, S12GBADI/S12GBADI1/S12GBADI2, or S12GCADI/S12GCADI1/S12GCADI2 interrupt to allow the DTC or DMAC to read the converted data enables sequence conversion without burden on software.
For details on DTC and DMAC settings, refer to section 18, Data Transfer Controller (DTCb) and section 17, DMA Controller (DMACAa).

42.4.2 Scan Complete Event Output to ELC

The ELC can set up linked operation of a module specified in advance by using the S12ADI, S12ADI1, or S12ADI2 interrupt request signal as an event signal.
The S12GBADI, S12GBADI1, or S12GBADI2 interrupt, S12CMPAI, S12CMPAI1, or S12CMPAI2 interrupt, and S12CMPBI, S12CMPBI1, or S12CMPBI2 interrupt request signals cannot be used as event signals. An event signal is output regardless of the setting of the corresponding interrupt request enable bit. The 12-bit A/D converter outputs the A/D conversion completed signals as event signals.

42.5 Allowable Impedance of Signal Source

Figure 42.50 shows an equivalent circuit of an analog input pin and an external sensor.
To perform A/D conversion accurately, the internal capacitor (Cs) must be fully charged within the sampling time. If the impedance (R0) of the signal source is high and it takes time to charge Cs, extend the sampling time with the ADSSTRn register. Conversely, if R0 is small, the sampling time can be shortened. Refer to the electrical characteristics for the permissible signal source impedance under various operating conditions.
When converting only a single pin input in single scan mode, the influence of R0 can be ignored because the input load becomes practically only the internal input resistor (Rs) by connecting an external high-capacity capacitor (C). However, because a low-pass filter is formed by R0 and C, it may not possible to follow the analog signal that changes at high speed. Insert a low-impedance buffer when converting high speed analog signals or when converting multiple pins in scan mode.

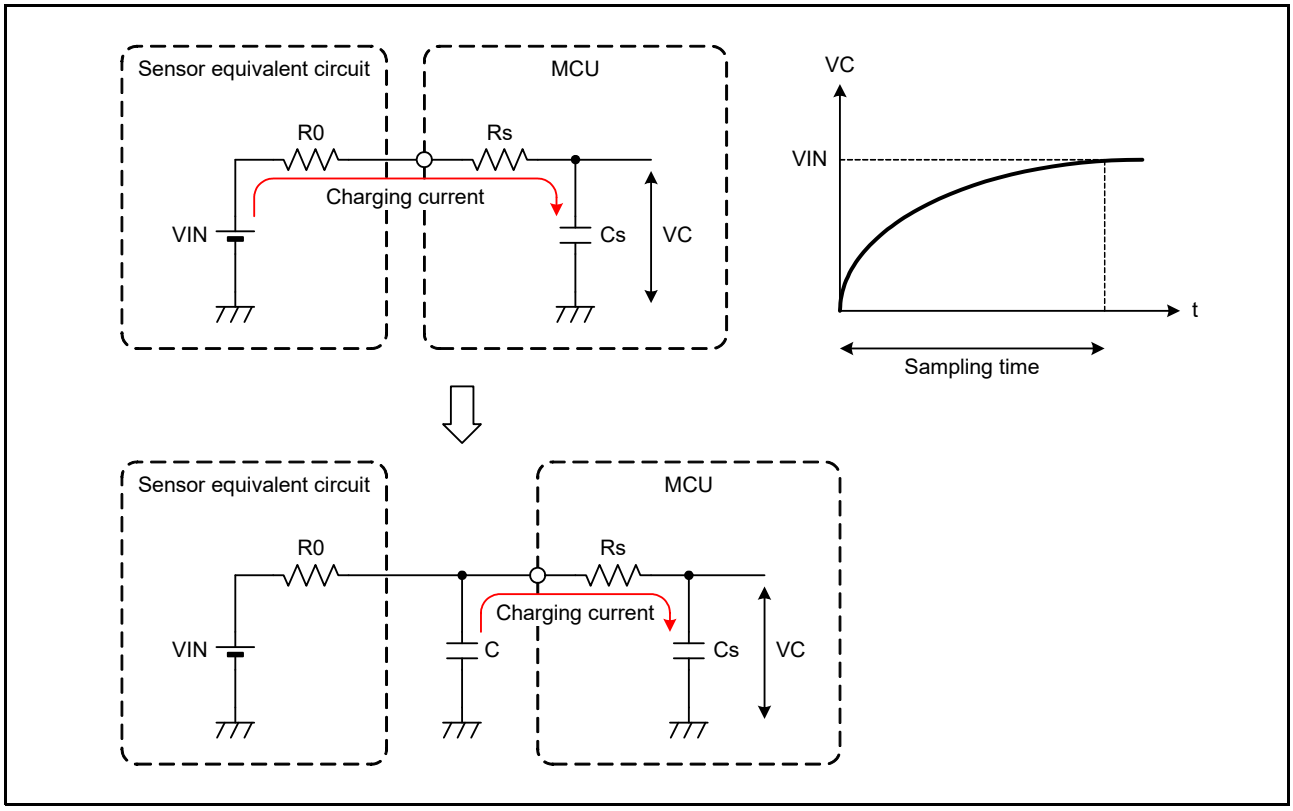


Figure 42.50 Equivalent Circuit of Analog Input Pin and External Sensor

42.6 Usage Notes

42.6.1 Notes on Reading Data Registers

The A/D data registers, A/D data duplication registers, A/D data duplication register A, A/D data duplication register B, A/D temperature sensor data register, A/D internal reference voltage data register, and A/D self-diagnosis data register should be read in 16-bit units. If a register is read twice in 8-bit units, that is, the higher-order byte and lower-order byte are separately read, the A/D-converted value having been read first may disagree with the A/D-converted value having been read for the second time. To prevent this, the data registers should never be read in 8-bit units.

42.6.2 Notes on Stopping A/D Conversion

42.6.2.1 Procedure of Stopping A/D Conversion

To stop A/D conversion when an asynchronous trigger or a synchronous trigger has been selected as the condition for starting A/D conversion, follow the procedure in Figure 42.51.

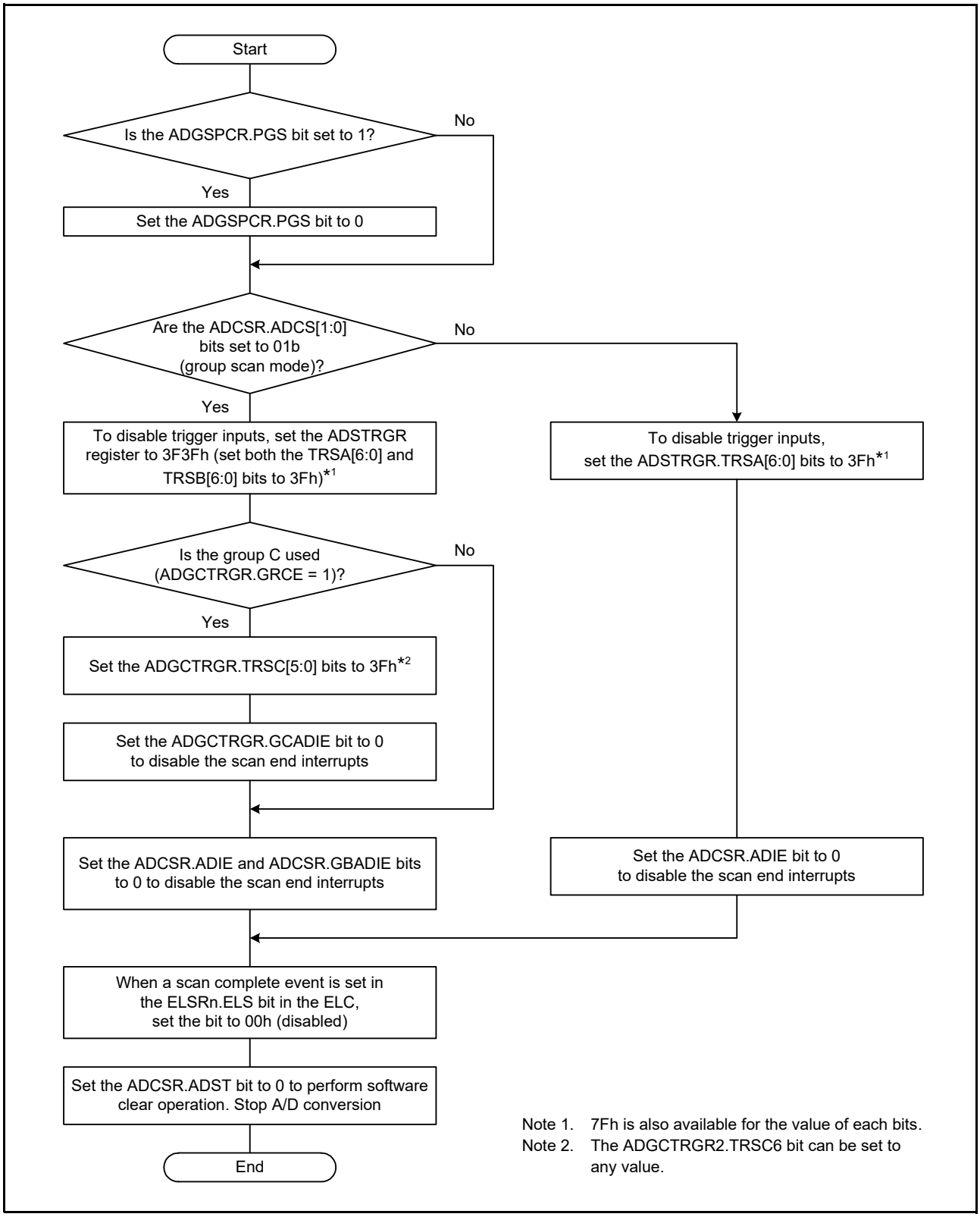


Figure 42.51 Procedure for Clear Operation by Software Through the ADCSR.ADST Bit

42.6.2.2 Notes on Modes and Status Flags

The following values should be initialized or reset as described below: the voltage status used in self-diagnosis, the odd-even determination for the double-trigger mode, the monitor flags for the comparison function.

- Re-set the voltage for use in self-diagnosis by setting the ADCER.DIAGLD bit to 1 and setting the ADCER.DIAGVAL[1:0] bits.
- Make double-trigger mode effective starting from the first scan by changing the ADCSR.DBLE bit from 0 to 1.
- Set the ADCMPCR.CMPAE and ADCMPCR.CMPBE bits to 0, which clears the monitor flags for the comparison function (MONCMPA, MONCMPB, MONCOMB).
- Initialize the constant sampling function (enabled by setting the ADSHMSR.SHMD bit to 1) by setting the ADSHMSR.SHMD bit to 0. After this, when using constant sampling function again, wait for at least 1 cycle of ADCLK and then set the ADSHMSR.SHMD bit to 1.

42.6.3 A/D Conversion Restarting Timing and Termination Timing

Starting operation of the analog blocks of the 12-bit A/D converter after setting the ADCSR.ADST bit to 1 takes up to six cycles of ADCLK. Stopping operation of the analog blocks of the 12-bit A/D converter by setting the ADCSR.ADST bit to 0 for a forced stop of the converter takes up to two cycles of ADCLK.
A period from forcible termination to idling of analog blocks takes a maximum of 1 PCLKB + 1 ADCLK cycles when PCLKB to ADCLK frequency ratio is 1:2 (ADCLK is faster).

42.6.4 Notes on Scan End Interrupt Handling

When scanning the same analog input twice using any trigger, the first A/D-converted data is overwritten with the second A/D-converted data in the case that the CPU does not complete reading the A/D-converted data by the time the A/D conversion of the first analog input for the second scan ends after the first scan end interrupt is generated.

42.6.5 Module Stop Function Setting

Operation of the 12-bit A/D converter can be disabled or enabled by setting module stop control register. The 12-bit A/D converter is stopped after released from the reset state. Register access is enabled by releasing the module stop state.
After the module stop state is released, wait for 1 µs to start A/D conversion. To stop the operation of the 12-bit A/D converter using module stop function, set the ADVMONCR.VDE and ADVMONO.VDO bits to 0.
For details, refer to section 11, Low Power Consumption.

42.6.6 Notes on Entering Low Power Consumption States

Before entering the module stop state or software standby mode, make sure to stop A/D conversion. Here, set the ADCSR.ADST bit to 0, and secure certain period of time until the analog unit of the 12-bit A/D converter is stopped. Follow the procedure given below to secure this time.
Set the ADCSR.ADST bit to 0 by following the procedure in Figure 42.51 procedure for clear operation by software through the ADCSR.ADST bit. Then wait for two clock cycles of ADCLK before entering the module stop mode or software standby mode.

42.6.7 Notes on Canceling Software Standby Mode

After releasing the A/D converter from software standby, wait for the oscillation stabilization flag for the main clock or PLL clock to become 1, wait for at least a further 1 µs after that, and then start A/D conversion.
For details, refer to section 11, Low Power Consumption.

42.6.8 Error in Absolute Accuracy When Disconnection Detection Assistance is in Use

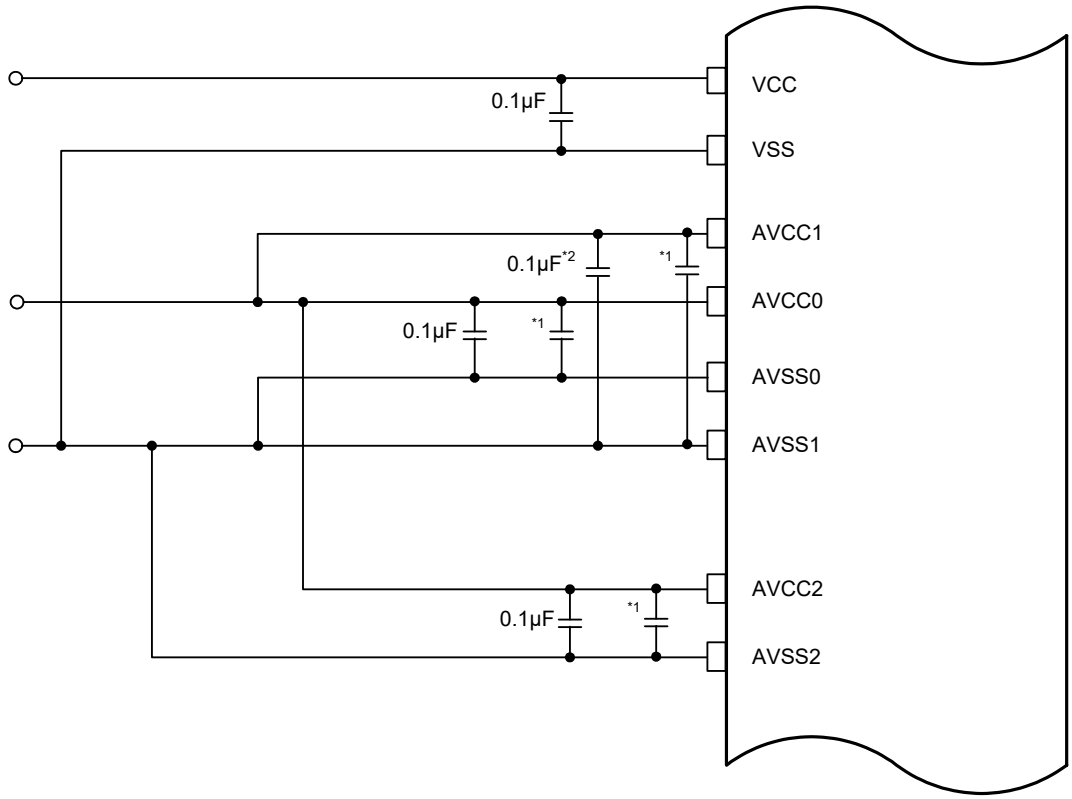
Using disconnection detection assistance leads to an error in absolute accuracy of the A/D converter. This is because an error voltage is input to the analog input pins due to the resistive voltage division between the pull-up or pull-down resistor (Rp) and the resistance of the signal source (Rs).
This error in absolute accuracy is calculated from the following formula. Only use disconnection detection assistance after thorough evaluation.

Maximum error in absolute accuracy (LSB) = 4095 × Rs/Rp

42.6.9 Voltage Range of Analog Power Supply Pins

If this MCU is used with the voltages outside the following ranges, the reliability of the MCU may be affected.

- Analog input voltage range
  Set the voltage within the range from AVSSn to AVCCn (n = 0 to 2) when applying voltages to analog pins AN000 to AN006, AN100 to AN103, AN200 to AN211, AN216, and AN217.
- Relationship between power supply pin pairs (AVCCn–AVSSn, VCC–VSS)
  The following condition should be satisfied: AVSSn = VSS (n = 0 to 2). When performing A/D conversion of analog input pins AN216 and AN217, the following condition should be satisfied: AVCCn = VCC.
  A 0.1-µF capacitor should be connected between each pair of power supply pins to create a closed loop with the shortest route possible as shown in Figure 42.52, and connection should be made so that the following condition is satisfied at the supply side: AVSSn = VSS
  When the 12-bit A/D converter is not used, the following conditions should be satisfied:
  AVCCn = VCC and AVSSn = VSS



Note1. When using the A/D converter with the A/D conversion clock at a frequency above 40 MHz, add a 0.01-µF capacitor between each 0.1-µF capacitor and power-supply pin.
Note2. Place the capacitors between AVCC1 and AVSS1 closer to the MCU than those for AVCC0 and AVSS0.

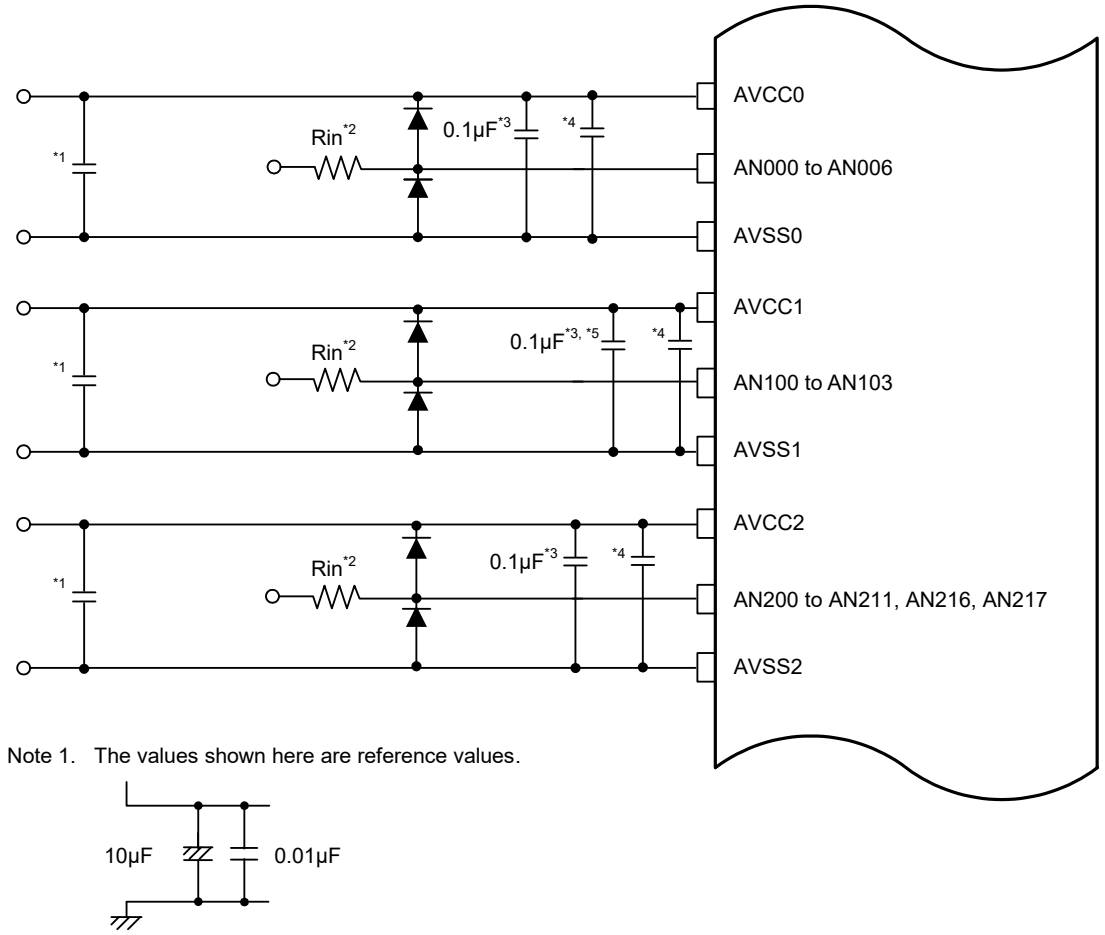
Figure 42.52 Example for Connecting Power Supply Pins

42.6.10 Notes on Board Design

The board should be designed so that digital circuits and analog circuits are separated from each other as far as possible. In addition, digital circuit signal lines and analog circuit signal lines should not intersect or placed near each other. If these rules are not followed, noise will be produced on analog signals and A/D conversion accuracy will be affected. The analog input pins (AN000 to AN006, AN100 to AN103, AN200 to AN211, AN216, and AN217) and analog power supply (AVCCn) should be separated from digital circuits using the analog ground (AVSSn). The analog ground (AVSSn) should be connected to a stable digital ground (VSS) on the board (single-point ground plane connection).

42.6.11 Notes on Noise Prevention

To prevent the analog input pins (AN000 to AN006, AN100 to AN103, AN200 to AN211, AN216, and AN217) from being destroyed by abnormal voltage such as excessive surge, capacitors should be inserted between AVCCn and AVSSn, and protection circuits should be connected to protect the above analog input pins as shown Figure 42.53.



Note 1. The values shown here are reference values.

Note2. Rin: Signal source impedance
Note3. Place the capacitors to be placed between the power supply pins AVCC0 and AVSS0, AVCC1 and AVSS1, and AVCC2 and AVSS2 as close to the pins as possible to improve the precision of A/D conversion.
Note4. When using the A/D converter with the A/D conversion clock at a frequency above 40 MHz, add a 0.01-µF capacitor between each 0.1-µF capacitor and power-supply pin.
Note5. Place the capacitors between AVCC1 and AVSS1 closer to the MCU than those for AVCC0 and AVSS0.

Figure 42.53 Sample Protection Circuit for Analog Inputs

42.6.12 Notes on Using Channel-Dedicated Sample-and-Hold Circuit

Enable use of the channel-dedicated sample-and-hold circuit (ADSHCR.SHANS[2:0] ≠ 000b), enable normal sampling by the channel-dedicated sample-and-hold circuit (ADSHMSR.SHMD = 1), and wait for 6 cycles of ADCLK to pass before input of a trigger signal.

To use the channel-dedicated sample-and-hold circuit, set the corresponding channel in any of the ADSCS0, ADSCS1, or ADSCS2 register.

43. 12-Bit D/A Converter (R12DAb)

43.1 Overview

This MCU includes two channels of 12-bit D/A converter.
Table 43.1 lists the specifications of the 12-bit D/A converter and Figure 43.1 shows a block diagram of the 12-bit D/A converter.

Table 43.1 Specifications of 12-Bit D/A Converter

Item	Specifications
Resolution	12 bits
Output channels	Two channels
Measure against mutual interference between analog modules	Measure against interference between D/A and A/D conversion D/A converted data update timing is controlled by the 12-bit A/D converter synchronous D/A conversion enable signal from the 12-bit A/D converter (unit 2). Therefore, the degradation of A/D conversion accuracy due to interference is reduced by controlling the timing in which the 12-bit D/A converter inrush current occurs, with the enable signal.
Low power consumption function	Module stop state can be set.
Event link function (input)	DA0 conversion can be started when an event signal is input.
Destination Selection	Outputs to the external pin and to the comparator C are separately controllable.

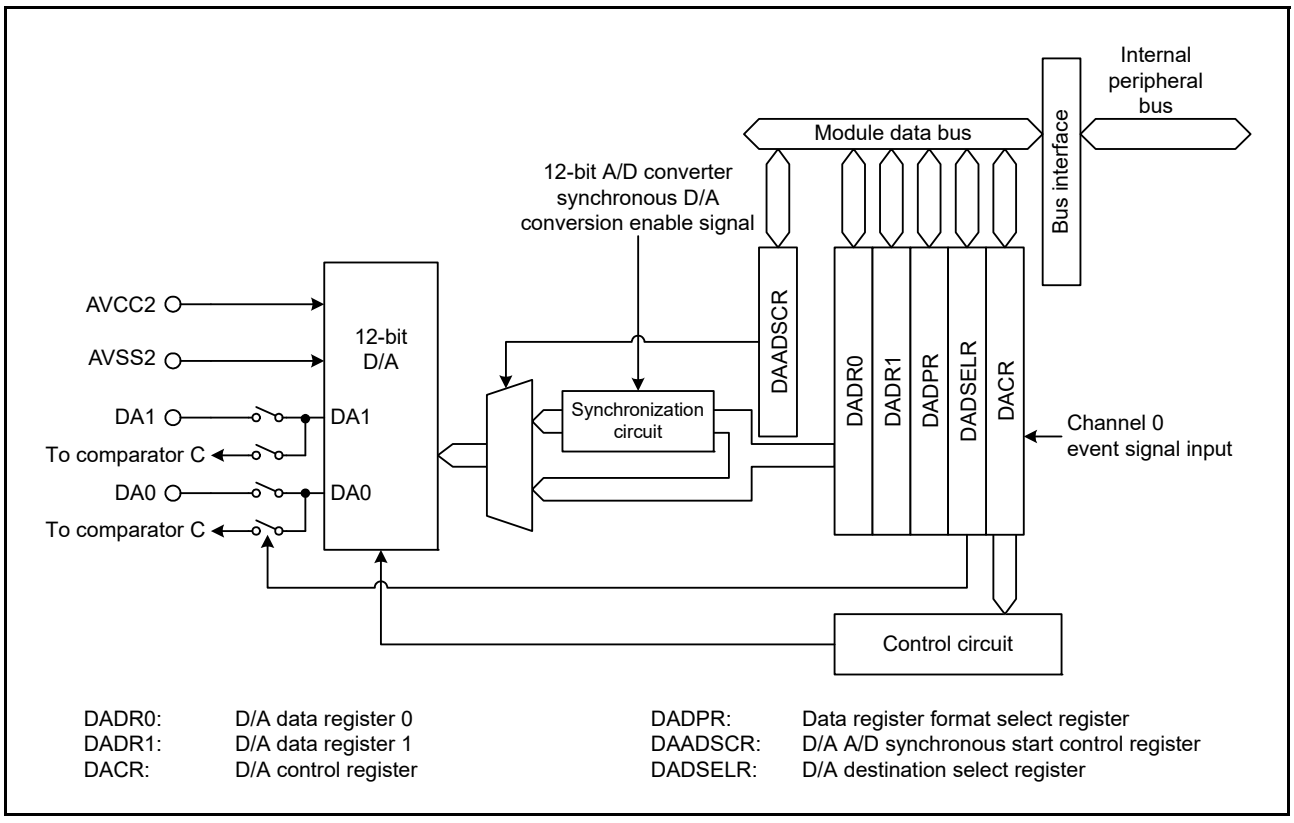


Figure 43.1 Block Diagram of 12-Bit D/A Converter

Table 43.2 lists the pin configuration of the 12-bit D/A converter.

Table 43.2 Pin Configuration of 12-Bit D/A Converter

Pin Name	I/O	Function
AVCC2	Input	Analog power supply pin
AVSS2	Input	Analog ground pin
DA0	Output	Channel 0 analog output pin
DA1	Output	Channel 1 analog output pin

43.2 Register Descriptions

43.2.1 D/A Data Register m (DADRm) (m = 0, 1)

Address(es): DA.DADR0 0008 8040h, DA.DADR1 0008 8042h

• DADPR.DPSEL bit = 0 (data is right-justified)

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

• DADPR.DPSEL bit = 1 (data is left-justified)

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
													—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The DADRm register is a 16-bit readable/writable register, which stores data to which D/A conversion is to be performed. Whenever an analog output is enabled, the values in DADRm are converted and output from the D/A converter.

12-bit data can be relocated by setting the DADPR.DPSEL bit.

Bits "—" are read as 0. The write value should be 0.

43.2.2 D/A Control Register (DACR)

Address(es): DA.DACR 0008 8044h

	b7	b6	b5	b4	b3	b2	b1	b0
	DAOE1	DAOE0	DAE	—	—	—	—	—
Value after reset:	0	0	0	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b5	DAE	D/A Enable*1	0: D/A conversion of channels 0 and 1 is controlled individually. 1: D/A conversion of channels 0 and 1 is enabled collectively.	R/W
b6	DAOE0	D/A Output Enable 0	0: Analog output of channel 0 (DA0) is disabled. 1: D/A conversion of channel 0 is enabled. Analog output of channel 0 (DA0) is enabled.	R/W
b7	DAOE1	D/A Output Enable 1	0: Analog output of channel 1 (DA1) is disabled. 1: D/A conversion of channel 1 is enabled. Analog output of channel 1 (DA1) is enabled.	R/W

Note 1. This bit controls D/A conversion in combination with the DAOEm bit (m = 0, 1). The DAOEm bit controls output of the results of conversion. For details, see Table 43.3.

Table 43.3 Controls of D/A Conversion

b5	b7	b6	
DAE	DAOE1	DAOE0	Description
0	0	0	D/A conversion and analog output pins (DA0, DA1) are disabled.*1
		1	D/A conversion of channel 0 is enabled. D/A conversion of channel 1 is disabled. Analog output of channel 0 (DA0) is enabled. Analog output of channel 1 (DA1) is disabled.*1
	1	0	D/A conversion of channel 0 is disabled. D/A conversion of channel 1 is enabled. Analog output of channel 0 (DA0) is disabled.*1 Analog output of channel 1 (DA1) is enabled.
		1	D/A conversion of channels 0 and 1 is enabled. Analog output of channels 0 and 1 (DA0, DA1) is enabled.
1	x	x	D/A conversion of channels 0 and 1 is enabled. Analog output of channels 0 and 1 (DA0, DA1) is enabled collectively.

x: Don't care

Note 1. When analog output is disabled, the analog output signal is placed in the Hi-Z state.

This register should be set when the DAADSCR.DAADST bit is 1 (measure against interference between D/A and A/D conversion is enabled) while the 12-bit A/D converter (unit 2) is halted (the ADCSR.ADST bit is 0). At that time, the software trigger should be selected for the 12-bit A/D converter (unit 2) trigger to securely stop the 12-bit A/D converter (unit 2).
Select the destination for each channel by using the DADSELR register before setting this register.

DAE Bit (D/A Enable)

The DAE bit controls D/A conversion and analog output in combination with the DAOEm bit (m = 0, 1).
When the measure against an interference between D/A and A/D conversions is enabled (DAADSCR.DAADST = 1), set the DAE bit while the ADCSR.ADST bit in the 12-bit A/D converter (unit 2) is 0. At that time, the software trigger should be selected for the 12-bit A/D converter (unit 2) trigger to securely stop the 12-bit A/D converter (unit 2).

DAOEm Bit (D/A Output Enable m) (m = 0, 1)

The DAOEm bit controls D/A conversion and analog output in combination with the DAE bit.

When both the DAOEm bit and DAE bit are 0, D/A conversion of channel m is not done and no conversion result is output.

When the measure against an interference between D/A and A/D conversions is enabled (DAADSCR.DAADST = 1), set the DAOEi bit while the ADCSR.ADST bit in the 12-bit A/D converter (unit 2) is 0. At that time, the software trigger should be selected for the 12-bit A/D converter (unit 2) trigger to securely stop the 12-bit A/D converter (unit 2).

The event link function can be used to set the DAOE0 bit to 1. The DAOE0 bit becomes 1 when the event specified by setting the ELSR16 register of the ELC occurs, and output of the D/A conversion results starts.

43.2.3 Data Register Format Select Register (DADPR)

Address(es): DA.DADPR 0008 8045h

	b7	b6	b5	b4	b3	b2	b1	b0
	DPSEL	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	DPSEL	Format Select	0: Data is right-justified. 1: Data is left-justified.	R/W

43.2.4 D/A A/D Synchronous Start Control Register (DAADSCR)

Address(es): DA.DAADSCR 0008 8046h

	b7	b6	b5	b4	b3	b2	b1	b0
	DAADST	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	DAADST	D/A A/D Synchronous Conversion	0: 12-bit D/A converter operation does not synchronize with 12-bit A/D converter (unit 2) operation. (measure against interference between D/A and A/D conversion is disabled) 1: 12-bit D/A converter operation synchronizes with 12-bit A/D converter (unit 2) operation. (measure against interference between D/A and A/D conversion is enabled)	R/W

As a measure against interference between D/A and A/D conversion, the DAADSCR register selects whether or not the timing for starting 12-bit D/A conversion is synchronized with the 12-bit A/D converter synchronous D/A conversion enable signal from the 12-bit A/D converter (unit 2).
This register should be set while the 12-bit A/D converter (unit 2) is halted (while the ADCSR.ADST bit is 0 after selecting software trigger as the 12-bit A/D converter (unit 2) trigger).

DAADST Bit (D/A A/D Synchronous Conversion)

Setting the DAADST bit to 0 allows the DADRm register value (m = 0, 1) to be converted into analog data at any time.
Setting the DAADST bit to 1 allows synchronous D/A conversion with the 12-bit A/D converter synchronous D/A conversion enable signal from the 12-bit A/D converter (unit 2). Therefore, even if the DADRm register value is modified, D/A conversion does not start until the 12-bit A/D converter (unit 2) completes A/D conversion.
Set this bit while the ADCSR.ADST bit is set to 0. At this time, the software trigger should be selected for the 12-bit A/D converter trigger to securely stop the 12-bit A/D converter (unit 2).
The event link function cannot be used when the DAADST bit is set to 1. Stop the event link function by setting the ELSR16 register of the ELC. The setting of the DAADST bit is common to channels 0 and 1 of the 12-bit D/A converter.

43.2.5 D/A Destination Select Register (DADSELR)

Address(es): DA.DADSELR 0008 8049h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	OUTREF1	OUTREF0	OUTDA1	OUTDA0
Value after reset:	0	0	0	0	0	0	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	OUTDA0	DA0 pin output enable	0: Does not output the channel 0 analog voltage from the DA0 pin 1: Output the channel 0 analog voltage from the DA0 pin	R/W
b1	OUTDA1	DA1 pin output enable	0: Does not output the channel 1 analog voltage from the DA1 pin 1: Output the channel 1 analog voltage from the DA1 pin	R/W
b2	OUTREF0	Reference voltage 0 output enable	0: Does not use the channel 0 analog voltage as a reference voltage for the comparator C 1: Use the channel 0 analog voltage as a reference voltage for the comparator C	R/W
b3	OUTREF1	Reference voltage 1 output enable	0: Does not use the channel 1 analog voltage as a reference voltage for the comparator C 1: Use the channel 1 analog voltage as a reference voltage for the comparator C	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R

The DADSELER register is used to set the destinations of the analog voltage generated by the D/A converter. Select such destinations with this register before the output is enabled with the DACR register.

43.3 Operation

The 12-bit D/A converter includes D/A conversion circuits for two channels, each of which can operate independently. When the DACR.DAOEm bit (m = 0, 1) is set to 1, D/A converter is enabled and the conversion result is output. An operation example of D/A conversion on channel 0 is shown below. Figure 43.2 shows the timing of this operation.

(1) Set the data for D/A conversion in the DADPR.DPSEL bit and the DADR0 register.
(2) Select the destination of the D/A converter analog voltage by using the DADSELR register.
(3) Set the DACR.DAOE0 bit to 1 to start D/A conversion. The DA0 output settles to the voltage corresponding to the setting value after the conversion time tDCONV has elapsed. The DA0 output voltage is held at this level until the DADR0 register is updated or the DAOE0 bit is set to 0. The output voltage (reference) is expressed by the following formula:

$$\frac{\text{Value of DADRm register}}{4096} \times \text{AVCC2}$$

(4) When the DADR0 register is updated, the conversion starts. The DA0 output settles at the new output voltage after the conversion time tDCONV has elapsed.
When the DAADSCR.DAADST bit is 1 (measure against interference between D/A and A/D conversion is enabled), it takes a maximum of one A/D conversion time for D/A conversion to start. When ADCLK is faster than the peripheral module clock, it may take longer than one A/D conversion time.
(5) When the DAOE0 bit is set to 0, analog output is disabled.

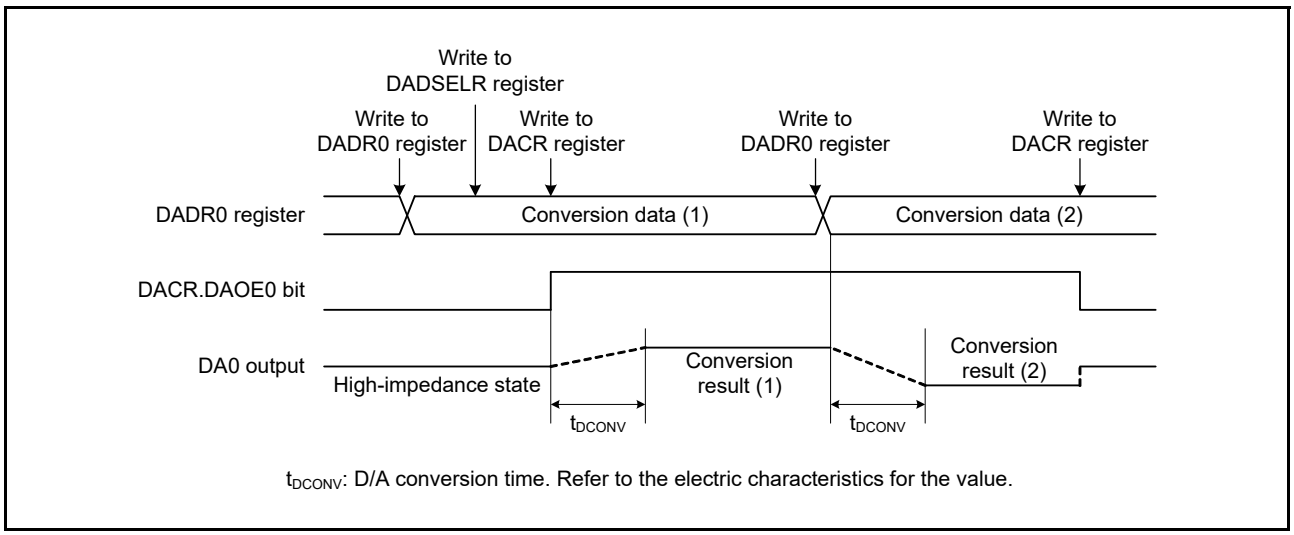


Figure 43.2 Example of 12-Bit D/A Converter Operation

43.3.1 Measure against Interference between D/A and A/D Conversion

When D/A conversion starts, an inrush current occurs to the 12-bit D/A converter. Since the same analog power supply is shared by the 12-bit D/A converter and 12-bit A/D converter (unit 2), the inrush current may interfere with the proper operation of the 12-bit A/D converter (unit 2).

With the DAADSCR.DAADST bit being 1, even if the DADRm register data (m = 0, 1) is modified during 12-bit A/D converter (unit 2) operation, D/A conversion does not start immediately but starts synchronously with A/D conversion completion. It takes a maximum of one A/D conversion time for the DADRm register data update to be reflected as the D/A conversion circuit input. Before reflection, the DADRm register value does not correspond to the analog output value.

When this function is enabled, it is impossible to check by any software means whether the DADRm register value has been D/A converted or not.

Even with the DAADSCR.DAADST bit being 1, when the DADRm register data is modified while the 12-bit A/D converter (unit 2) is halted, D/A conversion starts in one PCLKB cycle.

Figure 43.3 shows an example of channel 0 D/A conversion, in which the 12-bit D/A converter operates synchronously with the 12-bit A/D converter (unit 2).

(1) Confirm that the 12-bit A/D converter (unit 2) is halted. Set the DAADSCR.DAADST bit to 1.
(2) Confirm that the 12-bit A/D converter (unit 2) is halted. Set the DACR.DAOE0 bit to 1.
(3) Set the DADR0 register. When ADCLK is faster than the peripheral module clock, it may take longer than one A/D conversion time.

- If the 12-bit A/D conversion is halted (ADCSR.ADST bit = 0) when the DADR0 register is modified, D/A conversion starts in one PCLKB cycle.
- If the 12-bit A/D conversion is in progress (ADCSR.ADST bit = 1) when the DADR0 register is modified, D/A conversion starts upon A/D conversion completion. If the DADR0 register is modified twice during A/D conversion, the first update may not be converted.

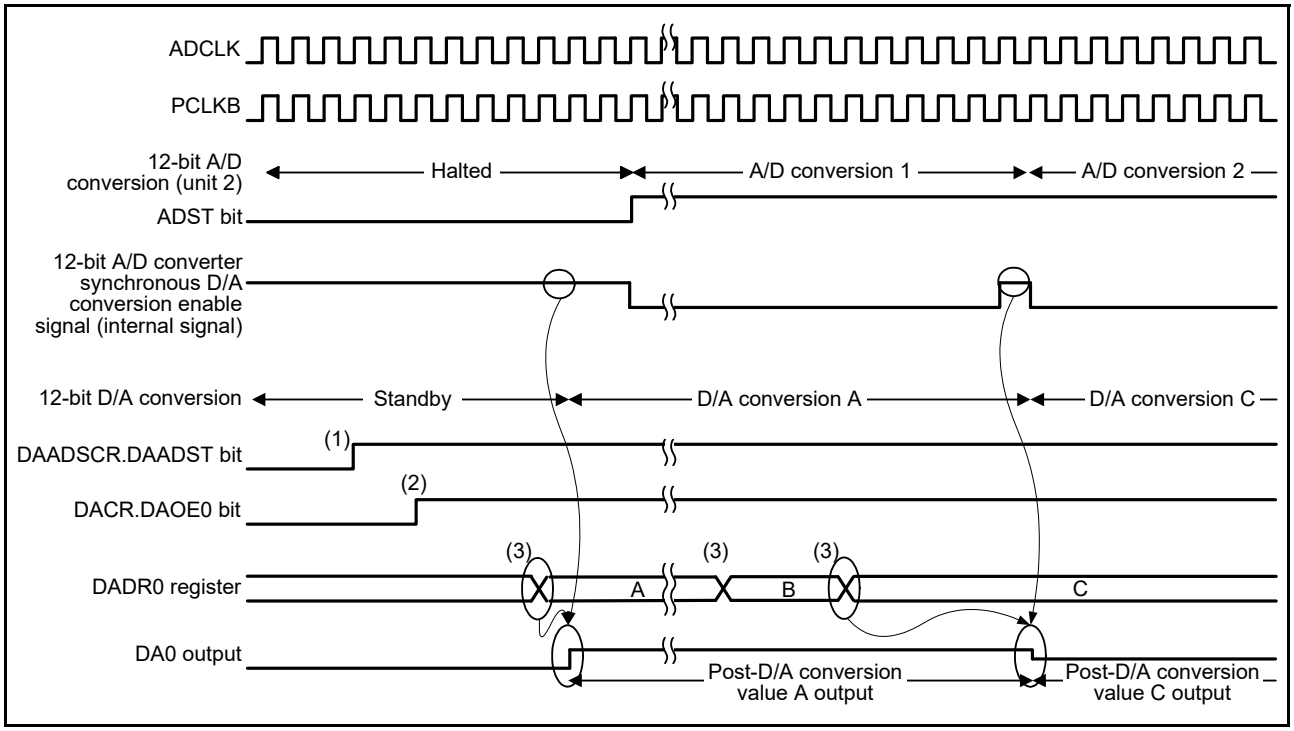


Figure 43.3 Example of Conversion When the 12-Bit D/A Converter is Synchronized with the 12-Bit A/D Converter (Unit 2)

When ADCLK is faster than PCLKB, the 12-bit D/A converter may not be able to capture a 12-bit A/D converter synchronous D/A conversion enable signal for one ADCLK cycle which is output between A/D conversion 1 and A/D conversion 2. Figure 43.4 shows example when the 12-bit D/A converter cannot capture the 12-bit A/D converter synchronous D/A conversion enable signal. In this case, the DA0 output is held at the level of the post-D/A conversion value A.

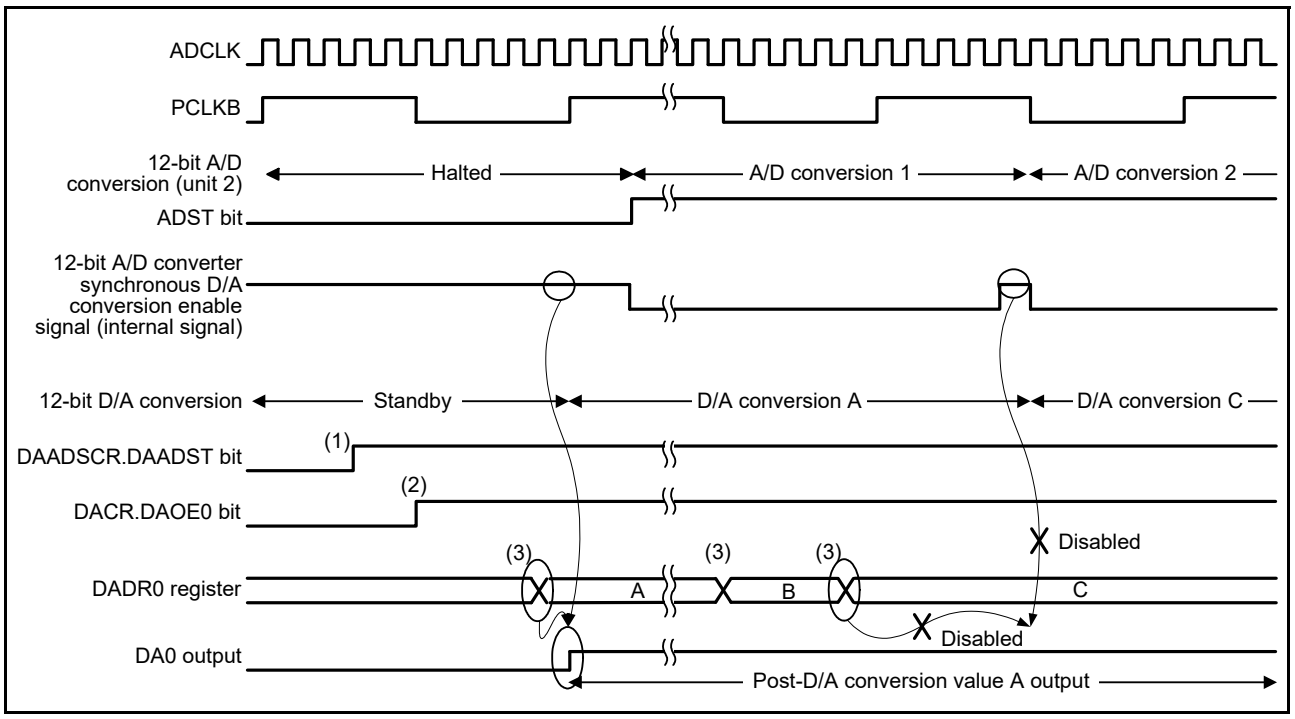


Figure 43.4 Example When the 12-Bit D/A Converter Cannot Capture the 12-Bit A/D Converter Synchronous D/A Conversion Enable Signal

43.4 Event Link Operation Setting Procedure

The event link operation procedure is described below.

(1) Set the DADPR.DPSEL bit and set the data for D/A conversion in the DADR0 register.
(2) Select the destination of the D/A converter analog voltage by using the DADSELR register.
(3) Set the bit value of the ELSR16 setting event signal to link the ELSR16 register of the ELC.
(4) Set the ELCR.ELCON bit to 1. This procedure enables event link operation for all modules with the event link function selected.
(5) Set the event output source module to activate the event link. After the event is output from the module, the DACR.DAOE0 bit becomes 1, and D/A conversion on channel 0 starts.
(6) Set the ELSR16.ELS[7:0] bits to 0000 0000b to stop event link operation of 12-bit D/A converter channel 0. All event link operation is stopped when the ELCR.ELCON bit is set to 0.

43.5 Usage Notes on Event Link Operation

(1) When the event link function is used, set the DACR.DAE bit to 0.
(2) When the event specified by the ELSR16 register is generated while the write cycle is performed to the DACR.DAOE0 bit, the write cycle is stopped, and the setting to 1 by the generated event takes precedence.
(3) Use of the event link function is prohibited when the DAADSCR.DAADST bit is set to 1 as the measure against an interfere between D/A and A/D conversions.

43.6 Usage Notes

43.6.1 Module Stop Function Setting

Operation of the 12-bit D/A converter can be disabled or enabled using the module stop control register. The initial setting is for operation of the 12-bit D/A converter to be stopped. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

43.6.2 Operation of the D/A Converter in Module Stop State

When the MCU enters the module stop state with D/A conversion enabled, the D/A converter outputs are retained, and the analog power supply current is the same as during D/A conversion. If the analog power supply current has to be reduced in the module stop state, disable D/A conversion by setting the DACR.DAOE1, DAOE0, and DAE bits to 0.

43.6.3 Operation of the D/A Converter in Software Standby Mode

When the MCU enters software standby mode with D/A conversion enabled, the D/A converter outputs are retained, and the analog power supply current is the same as during D/A conversion. If the analog power supply current has to be reduced in software standby mode, disable D/A conversion by setting the DACR.DAOE1, DAOE0, and DAE bits to 0.

43.6.4 Note on Usage When Measure against Interference between D/A and A/D Conversion is Enabled

When the DAADSCR.DAADST bit is 1 (measure against interference between D/A and A/D conversion is enabled), do not place the 12-bit A/D converter (unit 2) in the module stop state. It may halt D/A conversion in addition to A/D conversion.

43.6.5 Note on Using the D/A Converter Analog Voltage as a Reference Voltage for the Comparator C

When setting the DADSELR.OUTREFn bit (n = 0, 1) to 1, refer to section 45.4.5, Setting the D/A Converter.

43.6.6 Note on Enabling the Outputs Both to the DAn pin (n = 0, 1) and Comparator C

When the DADSELR.OUTDAn and OUTREFn bits are set to 1 simultaneously, the characteristics specified in section 49.8, D/A Conversion Characteristics may not be satisfied due to the influence of the load connected to the DAn pin. A full evaluation is recommended before use when both outputs are to be enabled.

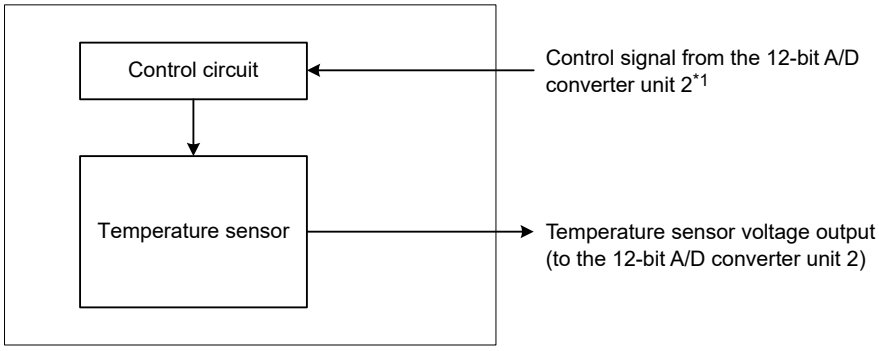
44. Temperature Sensor (TEMPS)

44.1 Overview

This MCU has a built-in temperature sensor. The temperature sensor outputs a voltage proportional to temperature. By converting the output voltage of the temperature sensor into a digital value with a 12-bit A/D converter unit 2 and converting it into temperature, the internal temperature of the MCU can be obtained.
Table 44.1 lists the specifications of the temperature sensor, and Figure 44.1 shows a block diagram of the temperature sensor.

Table 44.1 Specifications of Temperature Sensor

Item	Description
Temperature sensor voltage output	Temperature sensor outputs a voltage to the 12-bit A/D converter unit 2.
Temperature Sensor Calibration Data	Reference data measured for each chip at factory shipment is stored.



Note 1. When any of TSSA or TSSB bit in the ADEXICR register, or ADGCEXCR.TSSC bit for 12-bit A/D converter is set to 1, the temperature sensor is activated.

Figure 44.1 Block Diagram of Temperature Sensor

44.2 Register Descriptions

44.2.1 Temperature Sensor Calibration Data Register (TSCDR)

Address(es): TEMPS.TSCDR 007F B17Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—												
Value after reset:	0	0	0	0	Unique value for each chip											

The TSCDR register stores temperature sensor calibration data measured for each chip at factory shipment. The TSCDR register is a 32-bit read-only register and should be read in 32-bit units.
Temperature sensor calibration data is a digital value obtained using the 12-bit A/D converter unit 2 to convert the voltage output by the temperature sensor under the condition of Tj = 125°C and AVCC2 = 5 V.
The voltage V1 output by the temperature sensor under the condition of Tj = 125°C can be calculated from the value of the TSCDR register according to the formula below.

V1 = 5 × value of the TSCDR register/4096 [V]

Note that the AVCC2 voltage does not affect voltage V1.

44.3 Using the Temperature Sensor

The temperature sensor outputs a voltage proportional to temperature. By converting this voltage into a digital value with a 12-bit A/D converter unit 2 and converting it into temperature, the internal temperature of the MCU can be obtained.

44.3.1 Preparation for Using the Temperature Sensor

Perform a calibration of the temperature sensor as shown below. The voltage output by the temperature sensor is proportional to temperature, which can be calculated according to the following formula.

Formula for the temperature characteristic:
T = (Vs – V1)/Slope + T1

T: Measured temperature (°C)
Vs: Voltage output by the temperature sensor when the temperature is measured (V)
T1: Sample temperature measurement at first point (°C)
V1: Voltage output by the temperature sensor when T1 is measured (V)
T2: Sample temperature measurement at second point (°C)
V2: Voltage output by the temperature sensor when T2 is measured (V)
Slope: Temperature slope of the temperature sensor (V/°C); Slope = (V2 – V1)/(T2 – T1)

Since there are individual differences in temperature sensors, it is recommended to prepare a temperature slope by performing trial measurements at two different temperatures as shown below.
Use the 12-bit A/D converter unit 2 to measure the voltage V1 output by the temperature sensor at temperature T1. Again, using the 12-bit A/D converter unit 2, measure the voltage V2 output by the temperature sensor at a different temperature T2. Obtain the temperature slope (Slope = (V2 – V1)/(T2 – T1)) from these results.
Subsequently, obtain temperatures by substituting the slope into the formula for the temperature characteristic (T = (Vs – V1)/Slope + T1).

If you are using the temperature slope given in Table 49.50 of section 49, Electrical Characteristics, use the 12-bit A/D converter unit 2 to measure the voltage V1 output by the temperature sensor at temperature T1, and then calculate the temperature characteristic by using the formula below.
However, this calibration gives less accurate temperatures than two-point calibration.

T = (Vs – V1)/Slope + T1

T: Measured temperature (°C)
Vs: Voltage output by the temperature sensor when the temperature is measured (V)
T1: Sample temperature measurement at first point (°C)
V1: Voltage output by the temperature sensor when T1 is measured (V)
Slope: Temperature slope given in Table 49.50 ÷ 1000 (V/°C)

In this MCU, the TSCDR register stores the temperature value (CAL_{125}) of the temperature sensor measured under the condition of Tj = 125°C and AVCC2 = 5 V. By using this value as the sample measurement result at the first point, preparation before using the temperature sensor can be omitted.
If V1 is calculated from CAL_{125},

$V1 = 5 \times CAL_{125}/4096$ [V]
Using this, the measured temperature can be calculated according to the formula below.

T = (Vs – V1)/Slope + 125 [°C]

T: Measured temperature (internal temperature of the MCU) (°C)
Vs: Voltage output by the temperature sensor when the temperature is measured (V)
V1: Voltage output by the temperature sensor when Tj = 125°C and AVCC2 = 5 V (V)
Slope: Temperature slope given in Table 49.50 ÷ 1000 (V/°C)

Error in the measured temperature (the range of variation is 3σ) is shown in Figure 44.2.

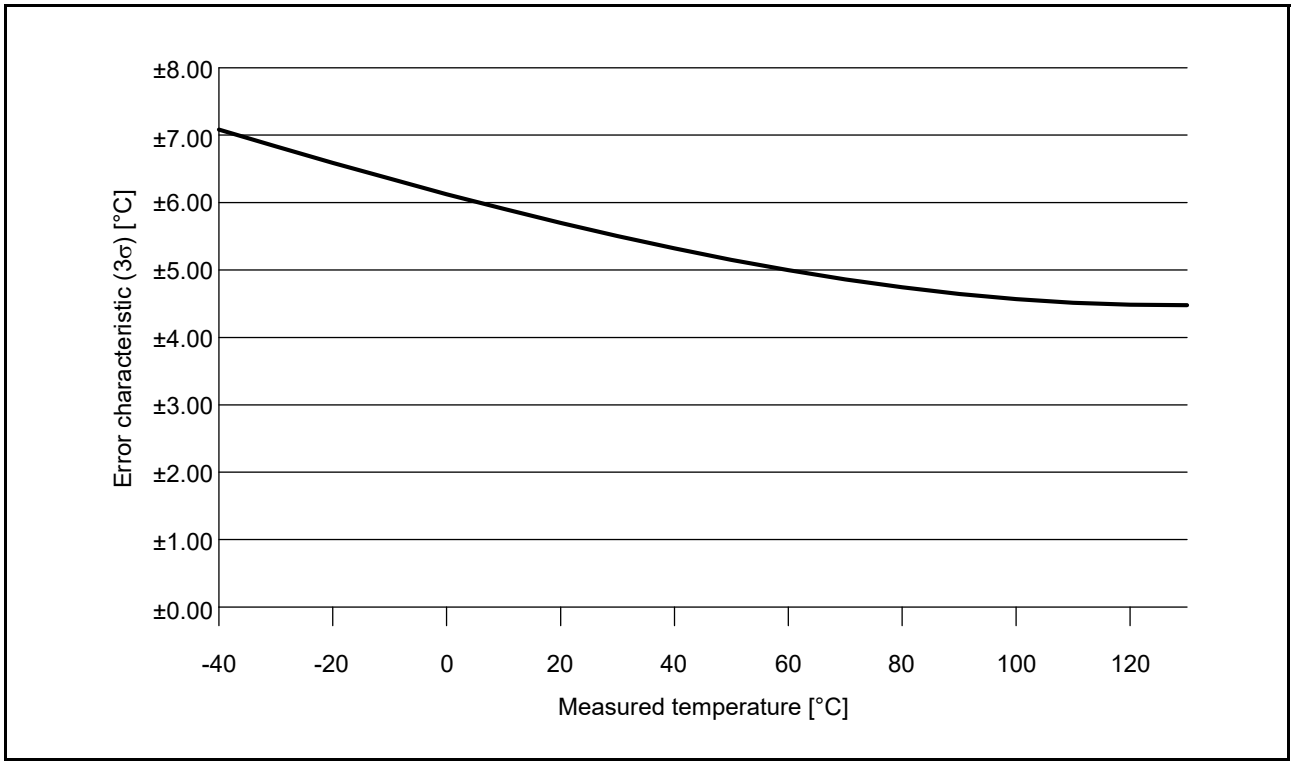


Figure 44.2 Error in the Measured Temperature

44.3.2 Setting of 12-Bit A/D Converter Unit 2

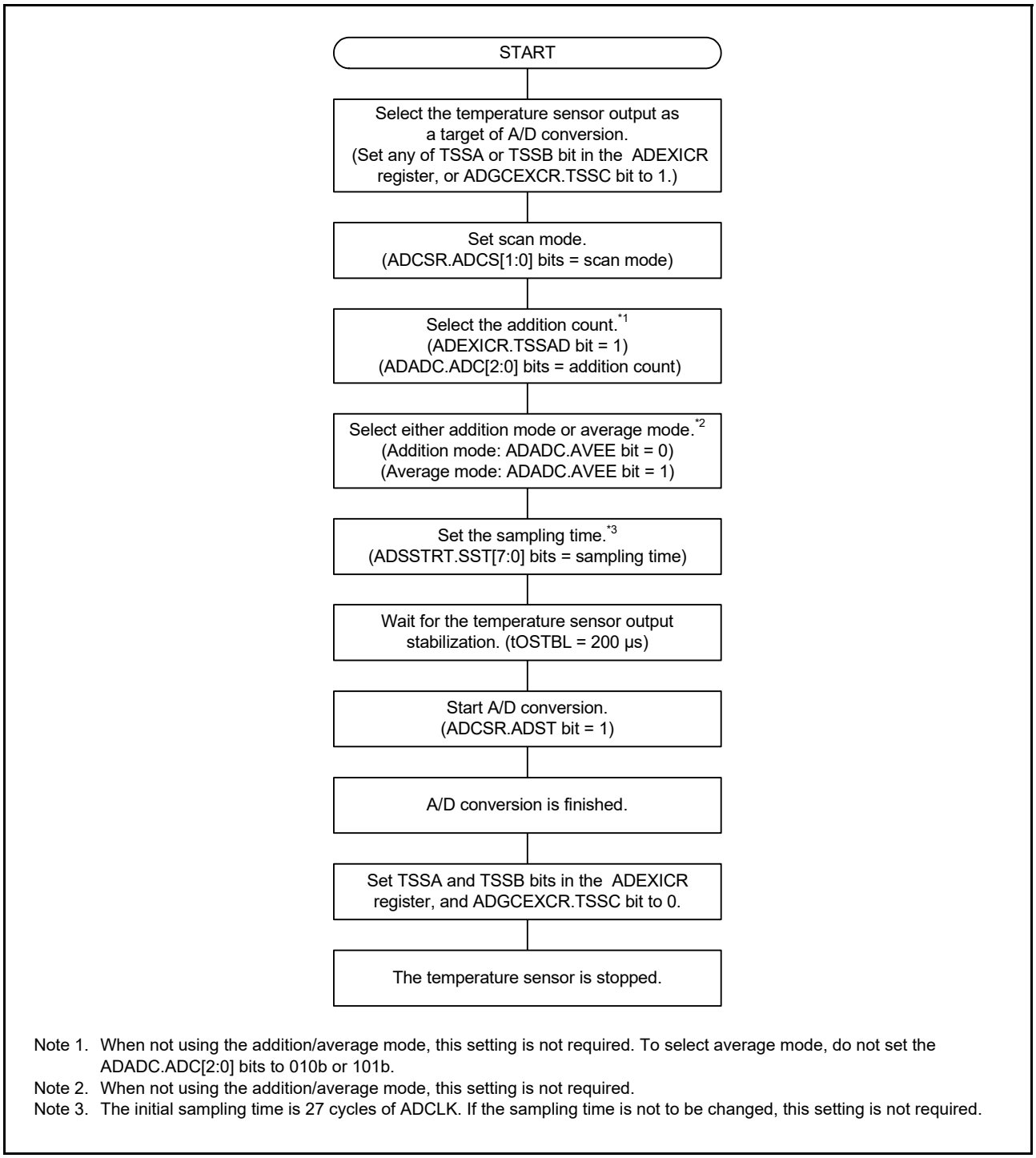
For A/D conversion of temperature sensor output voltages, 12-bit A/D converter unit 2 registers should be set as follows.

- Selecting the Temperature Sensor Voltage as a Source for A/D Conversion
  Select the temperature sensor voltage as a source for A/D conversion by setting the temperature sensor output A/D conversion select bit in the A/D conversion extended input control register (TSSA or TSSB in the ADEXICR register, or ADGCEXCR.TSSC) to 1.
- Setting Scan Mode
  Select scan mode by setting the scan mode select bits in the A/D control register (ADCSR.ADCS[1:0]).
  Set the bits to the single scan mode or the group scan mode.
- Setting Addition/Average Mode
  For A/D conversion of the temperature sensor output, additional or average mode is selectable. To use either additional or average mode, set the temperature sensor output A/D converted value addition mode select bit in the A/D conversion extended input control register (ADEXICR.TSSAD) to 1, and the addition count select bits in the A/D converted value addition count select register (ADADC.ADC[2:0]) to the desired number of addition. Furthermore, clear the AVEE bit in ADADC to 0 to select addition mode; set the AVEE bit in ADADC to 1 to select average mode. In average mode, however, the ADC[2:0] bits in ADADC should not be set to 010b or 101b.
- Setting the Sampling Time of the 12-bit A/D converter Unit 2
  The sampling time of the 12-bit A/D converter can be changed at converting the output from the temperature sensor. The initial sampling time is 27 cycles of ADCLK. To change the setting of sampling time from 27 cycles of ADCLK, set the A/D sampling state register T (ADSSTRT) while the ADCSR.ADST bit is 0. Refer to section 42.2.20, A/D Sampling State Register n (ADSSTRn) (n = 0 to 11, L, T, O) for the setting range of the sampling time. Set the sampling time to satisfy the specifications described in section 49, Electrical Characteristics.

Setting the A/D conversion start bit in the A/D control register (ADCSR.ADST) to 1 starts A/D conversion, and the result is stored in the A/D temperature sensor data register (ADTSDR). If you will be using A/D conversion of the output from the temperature sensor, do so in accord with section 44.3.3, Procedure for Using the Temperature Sensor.

44.3.3 Procedure for Using the Temperature Sensor

Figure 44.3 shows the procedure for using the temperature sensor.



Note 1. When not using the addition/average mode, this setting is not required. To select average mode, do not set the ADADC.ADC[2:0] bits to 010b or 101b.
Note 2. When not using the addition/average mode, this setting is not required.
Note 3. The initial sampling time is 27 cycles of ADCLK. If the sampling time is not to be changed, this setting is not required.

Figure 44.3 Procedure for Using the Temperature Sensor

44.3.4 Timing of A/D Conversion of Temperature Sensor Output

Figure 44.4 shows the timing from the start of temperature-sensor operation until the completion of A/D conversion when only the output from the temperature sensor is to be A/D converted and conversion is in single-scan mode. The times shown in the figure are described in Table 44.2.
When setting the ADST bit to 1 to perform the A/D conversion of the temperature sensor, auto-discharge is performed before sampling. The A/D conversion time of the temperature sensor includes the time for the auto discharge.

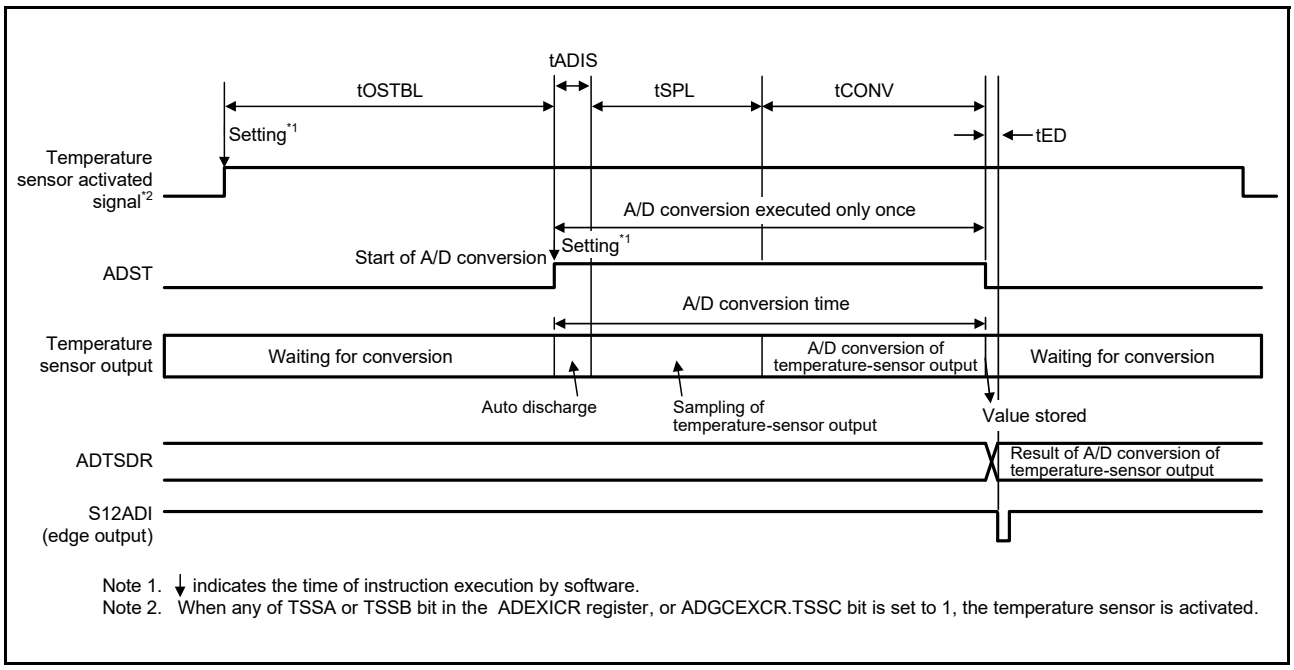


Figure 44.4 Timing from the Start of Temperature-Sensor Operation until Completion of A/D Conversion

Table 44.2 Time until Completion of A/D Conversion after the Start of Temperature-Sensor Operation

Item	Symbol	Time
Temperature-sensor output stabilization wait time	tOSTBL	200 µs (min)
Auto discharge time	tADIS	15 × tC(ADCLK)
12-bit A/D converter unit 2 input sampling time	tSPL	ADSSTRT setting × tC(ADCLK)
A/D conversion time	tCONV	Refer to Table 42.23, Times for Conversion During Scanning (in Numbers of Cycles of ADCLK and PCLKB) in section 42.3.7, Analog Input Sampling Time and Scan Conversion Time.
Scan conversion end delay time	tED	Refer to Table 42.23, Times for Conversion During Scanning (in Numbers of Cycles of ADCLK and PCLKB) in section 42.3.7, Analog Input Sampling Time and Scan Conversion Time.

44.4 Usage Note

44.4.1 Activating the Temperature Sensor

The temperature sensor is controlled by the registers in the 12-bit A/D converter unit 2 (S12AD2). To activate the temperature sensor, set any of TSSA or TSSB bit in the ADEXICR register, or ADGCEXCR.TSSC bit to 1 after releasing the S12AD2 from the module stop state. For details, refer to section 42, 12-Bit A/D Converter (S12ADHa).

45. Comparator C (CMPCa)

45.1 Overview

Comparator C compares a reference input voltage to an analog input voltage.
The comparison result can be read by software and output externally, and an interrupt request can be generated upon any changes to the comparison result.
The reference input voltage of comparator C can be selected from the input from the CVREFC0 or CVREFC1 pin, or the output from the on-chip D/A converter 0 or the on-chip D/A converter 1.
There are four analog inputs, one of which is to be selected.
Table 45.1 lists the specification of comparator C, Figure 45.1 shows a block diagram of comparator C, Table 45.2 shows a comparator C pin configuration, and Table 45.3 shows the analog input pin connections for comparator C.
In this section, “PCLK” is used to refer to PCLKB.

Table 45.1 Comparator C Specifications

Item	Specification
Number of channels	Six (comparator C0 to comparator C5)
Analog input voltages	Input voltage from the CMPCnm pin (n = channel number; m = 0 to 3)
Reference input voltage	Input voltage from the CVREFC0 or CVREFC1 pin, or output voltage from on-chip D/A converter 0 or on-chip D/A converter 1
Comparison result	The comparison result can be output externally.
Digital filter function	• One of three sampling periods can be selected. • The filter function can also be disabled. • A noise-filtered signal can be used to generate the interrupt request output, event output to the ELC, and POE source output*1, and the signal can be used to read the comparison result via registers.
Interrupt request	• An interrupt request is generated upon detecting a valid edge of the comparison result. • A valid edge can be selected from a rising or a falling edge or both edges.
Low power consumption function	Module stop state can be set.

Note 1. The POE only uses the level detection signal, and the POEG uses the level detection and edge detection signals.

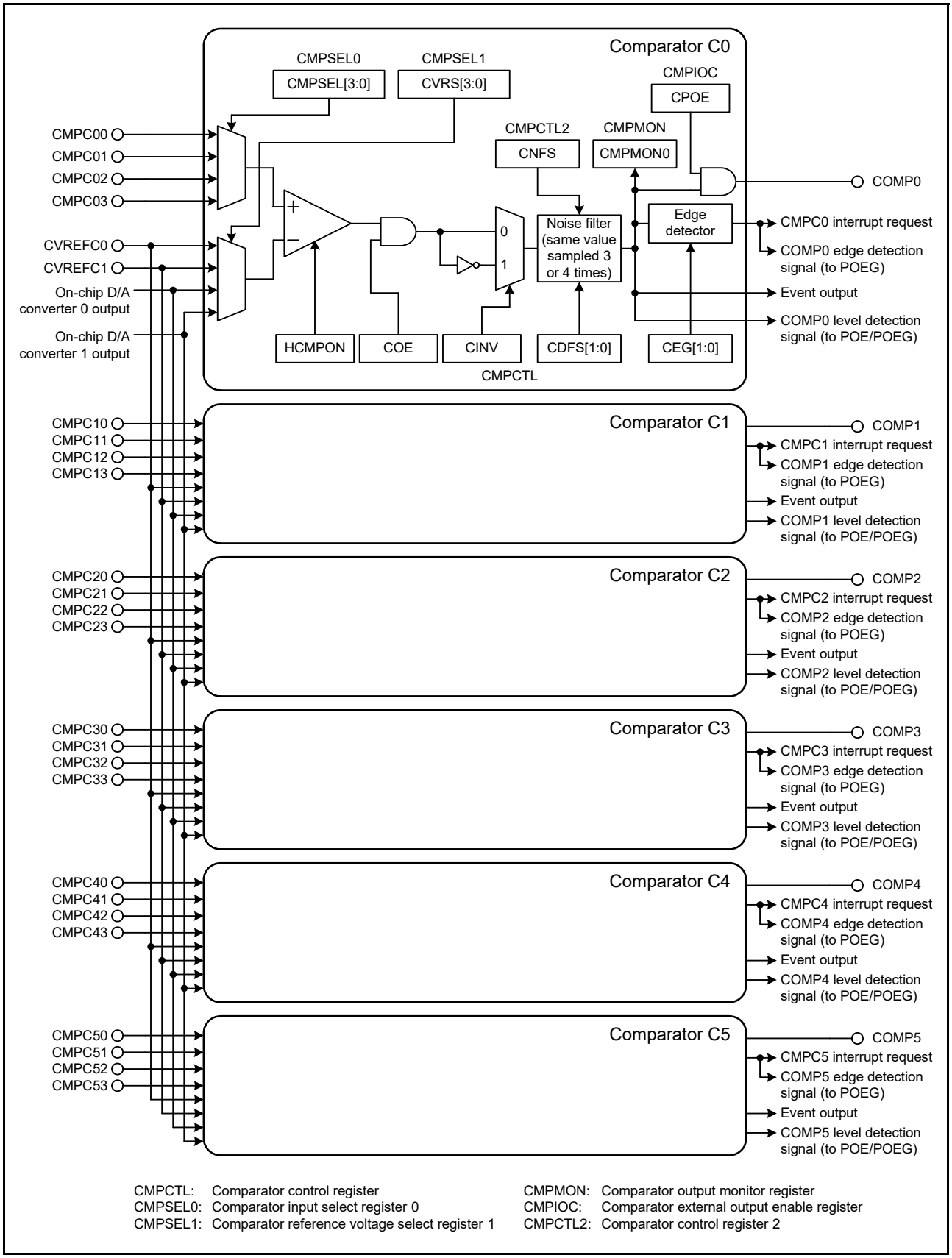


Figure 45.1 Block Diagram of Comparator C

Table 45.2 Comparator C Pin Configuration

Pin Name	I/O	Function
CMPC00, CMPC01, CMPC02, CMPC03	Input	Comparator C0 analog input pins
CMPC10, CMPC11, CMPC12, CMPC13	Input	Comparator C1 analog input pins
CMPC20, CMPC21, CMPC22, CMPC23	Input	Comparator C2 analog input pins
CMPC30, CMPC31, CMPC32, CMPC33	Input	Comparator C3 analog input pins
CMPC40, CMPC41, CMPC42, CMPC43	Input	Comparator C4 analog input pins
CMPC50, CMPC51, CMPC52, CMPC53	Input	Comparator C5 analog input pins
CVREFC0	Input	Reference input voltage pin 0
CVREFC1	Input	Reference input voltage pin 1
COMP0	Output	Comparator C0 comparison result output pin
COMP1	Output	Comparator C1 comparison result output pin
COMP2	Output	Comparator C2 comparison result output pin
COMP3	Output	Comparator C3 comparison result output pin
COMP4	Output	Comparator C4 comparison result output pin
COMP5	Output	Comparator C5 comparison result output pin

Table 45.3 Analog Input Pin Connections for Comparator C

	Connection	
Analog Input Pin	Products with 48-Kbyte RAM	Products with 64-Kbyte RAM
CMPC00	AN000 pin	AN000 pin
CMPC01	AN004 pin	Programmable gain amplifier output for AN000 pin
CMPC02	AN200 pin	AN200 pin
CMPC03	AN206 pin	AN206 pin
CMPC10	AN001 pin	AN001 pin
CMPC11	AN005 pin	Programmable gain amplifier output for AN001 pin
CMPC12	AN201 pin	AN201 pin
CMPC13	AN000 pin	AN207 pin
CMPC20	AN002 pin	AN002 pin
CMPC21	AN006 pin	Programmable gain amplifier output for AN002 pin
CMPC22	AN202 pin	AN202 pin
CMPC23	AN003 pin	AN209 pin
CMPC30	—	AN100 pin
CMPC31	—	Programmable gain amplifier output for AN100 pin
CMPC32	—	AN203 pin
CMPC33	—	AN210 pin
CMPC40	—	AN101 pin
CMPC41	—	Programmable gain amplifier output for AN101 pin
CMPC42	—	AN204 pin
CMPC43	—	AN208 pin
CMPC50	AN003 pin	AN102 pin
CMPC51	AN208 pin	Programmable gain amplifier output for AN102 pin
CMPC52	AN210 pin	AN205 pin
CMPC53	AN211 pin	AN211 pin

45.2 Register Descriptions

45.2.1 Comparator Control Register (CMPCTL)

Address(es): CMPC0.CMPCTL 000A 0C80h, CMPC1.CMPCTL 000A 0CA0h, CMPC2.CMPCTL 000A 0CC0h, CMPC3.CMPCTL 000A 0CE0h, CMPC4.CMPCTL 000A 0D00h, CMPC5.CMPCTL 000A 0D20h

	b7	b6	b5	b4	b3	b2	b1	b0
	HCMPON	CDFS[1:0]		CEG[1:0]		—	COE	CINV
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CINV	Comparator Output Polarity Select*1, *4	0: Comparator output not inverted 1: Comparator output inverted	R/W
b1	COE	Comparator Output Enable	0: Comparator output disabled (the output signal is fixed to 0) 1: Comparator output enabled	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4, b3	CEG[1:0]	Comparator Edge Interrupt Detection Select	b4 b3 0 0: Interrupt request is not generated. 0 1: Rising edge 1 0: Falling edge 1 1: Both edges	R/W
b6, b5	CDFS[1:0]	Noise Filter Sampling Select*1, *2, *4	b6 b5 0 0: Noise filter not used 0 1: Sampling frequency is PCLK/8. 1 0: Sampling frequency is PCLK/16. 1 1: Sampling frequency is PCLK/32.	R/W
b7	HCMPON	Comparator Operation Enable*3	0: Operation stopped (the output signal is fixed to 0) 1: Operation enabled (input to the comparator pins is enabled)	R/W

Note: Set this register before setting registers in the ELC when using event output.
Note: Set this register before setting registers in the POE when using comparator output as a POE source.
Note 1. Rewrite the CDFS[1:0] and CINV bits only after disabling the comparator output (COE bit = 0).
Note 2. If the CDFS[1:0] bits are changed from 00b (noise filter not used) to a value other than 00b (noise filter used), allow four sampling times to elapse until the filter output is updated, and then use the CMPCn interrupt request output, an event output, and a POE source output.
Note 3. The operation stabilization wait time is required after enabling comparator operation (HCMPON bit = 1). As for the value, refer to section 49, Electrical Characteristics.
Note 4. Rewriting the CINV bit or CDFS[1:0] bits may generate a CMPCn interrupt request, event output, and POE source. Before changing these bits, set the registers in the ELC so that the comparator output is not linked and set the registers in the POE/POEG so that comparator output is not used for output disabling control. After changing these bits, also set the corresponding interrupt status flag (IR) in the interrupt request register, the POE comparator channel n output detection flag (n = 0 to 5), and the GPTW or CMPC output stop request detection flag in the POEG to 0.

CEG[1:0] Bits (Comparator Edge Interrupt Detection Select)

These bits select which edge of comparator output signal is used to generate an interrupt request.

The valid edge is set for the signal after the comparator polarity is selected by the CINV bit and the filter is selected by CDFS[1:0] bits.

45.2.2 Comparator Input Select Register (CMPSEL0)

Address(es): CMPC0.CMPSEL0 000A 0C84h, CMPC1.CMPSEL0 000A 0CA4h, CMPC2.CMPSEL0 000A 0CC4h, CMPC3.CMPSEL0 000A 0CE4h, CMPC4.CMPSEL0 000A 0D04h, CMPC5.CMPSEL0 000A 0D24h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	CMPSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	CMPSEL[3:0]	Comparator Input Select*1, *2, *3	• Comparator C0 b3 b0 0 0 0 0: No input 0 0 0 1: CMPC00 selected 0 0 1 0: CMPC01 selected 0 1 0 0: CMPC02 selected 1 0 0 0: CMPC03 selected Settings other than above are prohibited. • Comparator C1 b3 b0 0 0 0 0: No input 0 0 0 1: CMPC10 selected 0 0 1 0: CMPC11 selected 0 1 0 0: CMPC12 selected 1 0 0 0: CMPC13 selected Settings other than above are prohibited. • Comparator C2 b3 b0 0 0 0 0: No input 0 0 0 1: CMPC20 selected 0 0 1 0: CMPC21 selected 0 1 0 0: CMPC22 selected 1 0 0 0: CMPC23 selected Settings other than above are prohibited. • Comparator C3 b3 b0 0 0 0 0: No input 0 0 0 1: CMPC30 selected 0 0 1 0: CMPC31 selected 0 1 0 0: CMPC32 selected 1 0 0 0: CMPC33 selected Settings other than above are prohibited. • Comparator C4 b3 b0 0 0 0 0: No input 0 0 0 1: CMPC40 selected 0 0 1 0: CMPC41 selected 0 1 0 0: CMPC42 selected 1 0 0 0: CMPC43 selected Settings other than above are prohibited. • Comparator C5 b3 b0 0 0 0 0: No input 0 0 0 1: CMPC50 selected 0 0 1 0: CMPC51 selected 0 1 0 0: CMPC52 selected 1 0 0 0: CMPC53 selected Settings other than above are prohibited.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Rewrite the CMPSEL[3:0] bits in the following procedure. Writing a value other than 0000b while the value of these bits is not

0000b is invalid. Writing 1 to two or more bits is also invalid. In both cases, the previous value is retained.
(1) Set the CMPCTL.COE bit to 0.
(2) Set the CMPSEL[3:0] bits to 0000b.
(3) Set a new value to the CMPSEL[3:0] bits (with 1 set in only one of the bits).
(4) Wait for the stabilization time for input selection. As for the value, refer to section 49, Electrical Characteristics.
(5) Set the CMPCTL.COE bit to 1.
(6) Set the corresponding interrupt status flag (IR) in the interrupt request register to 0.
(7) Set the GPTW or CMPC output stop request detection flag (POEGGj.IOCF (j = A to D)) in the corresponding POEG to 0.

Note 2. When using the event output, note that writing to these bits after the corresponding register in the ELC is set may trigger the output of an event signal.

Note 3. If COMPn level detection signal is used as a POE source, note that write access to these bits after the setting of any register in the POE may generate a POE source.

45.2.3 Comparator Reference Voltage Select Register (CMPSEL1)

Address(es): CMPC0.CMPSEL1 000A 0C88h, CMPC1.CMPSEL1 000A 0CA8h, CMPC2.CMPSEL1 000A 0CC8h, CMPC3.CMPSEL1 000A 0CE8h, CMPC4.CMPSEL1 000A 0D08h, CMPC5.CMPSEL1 000A 0D28h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	CVRS[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	CVRS[3:0]	Reference Input Voltage Select*1, *2, *3, *4	b3 b0 0 0 0 0: No input 0 0 0 1: On-chip D/A converter 1 output voltage selected as reference input voltage 0 0 1 0: On-chip D/A converter 0 output voltage selected as reference input voltage 0 1 0 0: Input voltage to the CVREFC1 pin selected as reference input voltage 1 0 0 0: Input voltage to the CVREFC0 pin selected as reference input voltage Settings other than above are prohibited.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When the on-chip D/A converter output voltage is used, set the D/A converter for generating comparator C reference voltage before enabling comparator operation (CMPCTL.HCMPON bit = 1). For details on setting the D/A converter, refer to section 43, 12-Bit D/A Converter (R12DAb).

Note 2. Rewrite the CVRS[3:0] bits in the following procedure. Be sure to set the CVRS[3:0] bits to 0000b before changing the set value. Direct rewriting from 0001b to 0010b or 0010b to 0001b, for example, will be ignored even if it is tried.
(1) Set the CMPCTL.COE bit to 0.
(2) Set the CVRS[3:0] bits to 0000b.
(3) Set a new value to the CVRS[3:0] bits (with 1 set in only one of the bits).
(4) Wait for the stabilization time for input selection. As for the value, refer to section 49, Electrical Characteristics.
(5) Set the CMPCTL.COE bit to 1.
(6) Set the corresponding interrupt status flag (IR) in the interrupt request register to 0.
(7) Set the GPTW or CMPC output stop request detection flag (POEGGj.IOCF (j = A to D)) in the corresponding POEG to 0.

Note 3. When using the event output, note that writing to these bits after the corresponding register in the ELC is set may trigger the output of an event signal.

Note 4. If COMPn level detection signal is used as a POE source, note that write access to these bits after the setting of any register in the POE may generate a POE source.

45.2.4 Comparator Output Monitor Register (CMPMON)

Address(es): CMPC0.CMPMON 000A 0C8Ch, CMPC1.CMPMON 000A 0CACh, CMPC2.CMPMON 000A 0CCCh, CMPC3.CMPMON 000A 0CECh, CMPC4.CMPMON 000A 0D0Ch, CMPC5.CMPMON 000A 0D2Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CMPMON0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMPMON0	Comparator Output Monitor Flag *1	0: Comparator output is 0. 1: Comparator output is 1.	R
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When comparator operation is enabled (CMPCTL.HCMPON and COE bits are 1) while the noise filter is disabled (CMPCTL.CDFS[1:0] bits are 00b), read the CMPMON0 bit twice and use the value only when the results match.

45.2.5 Comparator External Output Enable Register (CMPIOC)

Address(es): CMPC0.CMPIOC 000A 0C90h, CMPC1.CMPIOC 000A 0CB0h, CMPC2.CMPIOC 000A 0CD0h, CMPC3.CMPIOC 000A 0CF0h, CMPC4.CMPIOC 000A 0D10h, CMPC5.CMPIOC 000A 0D30h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CPOE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CPOE	External Pin Output Enable	Comparison result by the comparator is output to an external pin. 0: Output to the comparator external pin is disabled (the output signal is fixed to low) 1: Output to the comparator external pin is enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

45.2.6 Comparator Control Register 2 (CMPCTL2)

Address(es): CMPC0.CMPCTL2 000A 0C98h, CMPC1.CMPCTL2 000A 0CB8h, CMPC2.CMPCTL2 000A 0CD8h, CMPC3.CMPCTL2 000A 0CF8h, CMPC4.CMPCTL2 000A 0D18h, CMPC5.CMPCTL2 000A 0D38h

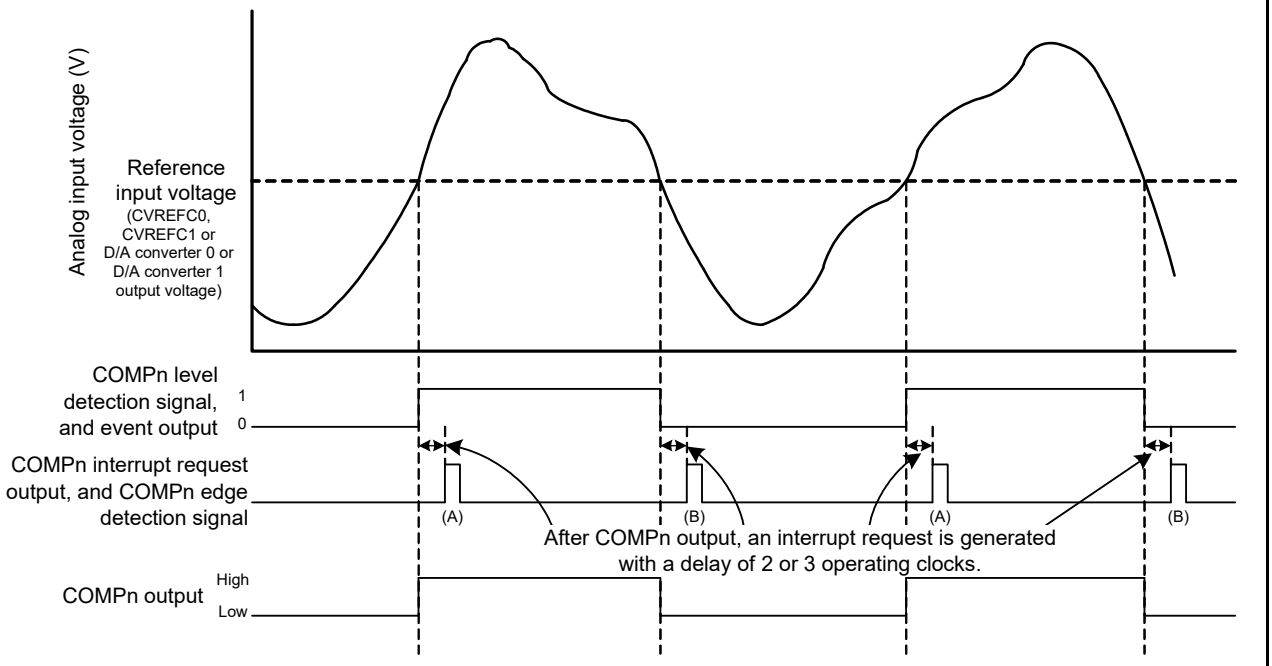
	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CNFS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CNFS	Noise Filter Judgment Standard Setting	0: Output changes when the same value is sampled 3 times. 1: Output changes when the same value is sampled 4 times.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

45.3 Operation

45.3.1 Comparator Operation Example

Figure 45.2 shows an operation example of the comparator. The COMPn level detection signal (n = 0 to 5) becomes high when the analog input voltage is higher than the reference input voltage, and the COMPn level detection signal becomes low when the analog input voltage is lower than the reference input voltage (when the CMPCTL.CINV bit is 0). The COMPn level detection signal can also be used as event output signal. When the CPOE bit in the corresponding CMPIOC register is 1, the signal is output from the COMPn pin. Interrupt request and COMPn edge detection signal are output in response to changes in the comparator output.



Note: The above diagram applies when the CMPIOC.CPOE bit is 1 (pin output enabled), the CMPCTL.CDFS[1:0] bits are 00b (filter not used), the CMPCTL.CEG[1:0] bits are 11b (both edges selected), and the CMPCTL.CINV bit is 0 (comparator output not inverted).

Figure 45.2 Comparator Operation Example (n = 0 to 5)

45.3.2 Noise Filter

Comparator C contains a noise filter. The sampling clock can be selected by the CMPCTL.CDFS[1:0] bits and the number of matches can be selected by the CMPCTL2.CNFS bit.

The comparator output signal is sampled every sampling clock, and if the same value is sampled three or four times, that value is determined as the noise filter output at the next sampling clock.

Figure 45.3 shows the configuration of the noise filter and edge detector and Figure 45.4 shows an example of the comparator noise filter and interrupt operation.

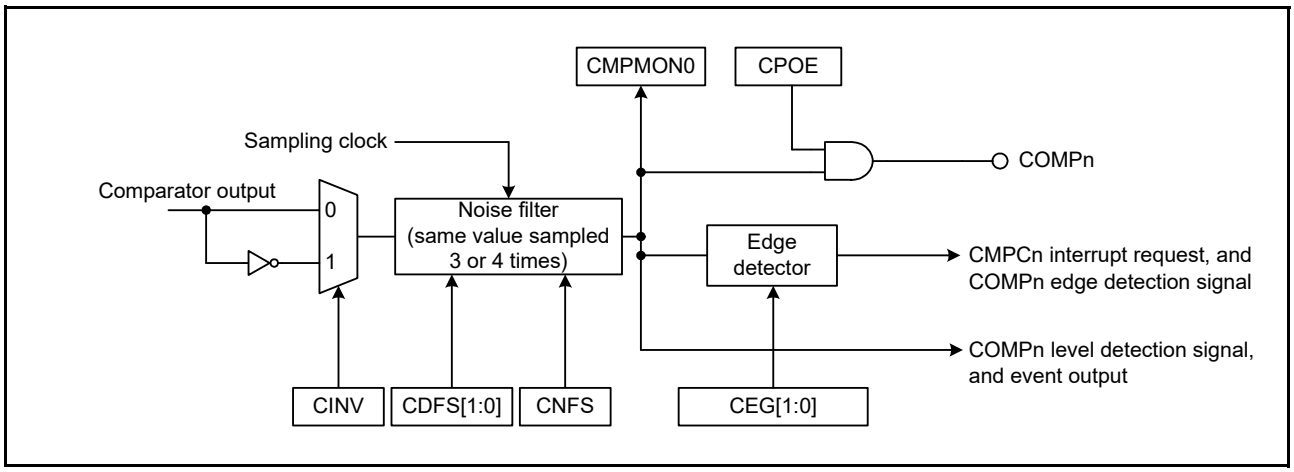
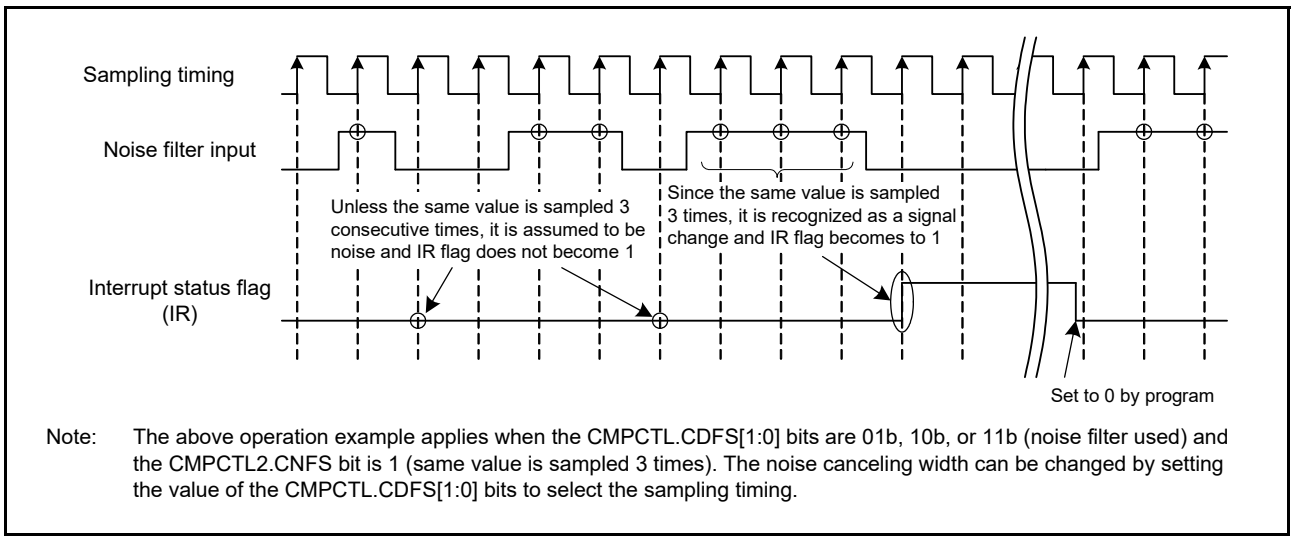


Figure 45.3 Noise Filter and Edge Detector Configuration (n = 0 to 5)



Note: The above operation example applies when the CMPCTL.CDFS[1:0] bits are 01b, 10b, or 11b (noise filter used) and the CMPCTL2.CNFS bit is 1 (same value is sampled 3 times). The noise canceling width can be changed by setting the value of the CMPCTL.CDFS[1:0] bits to select the sampling timing.

Figure 45.4 Noise Filter and Interrupt Operation Example

45.3.3 Interrupts

Comparator C generates an interrupt request upon detecting any changes in the comparison result.
When using the CMPCn interrupt, set at least one of bits CMPCTL.CEG[1:0] to 1 (to a value other than 00b (interrupt request is not generated)).
To use the CMPCn interrupt, use the following setting procedure. Note that steps (1), (2), and (3) can be set in any order.

(1) When using the on-chip D/A converter output voltage as the reference input voltage, set the D/A converter for generating comparator C reference voltage and enable operation.
(2) Set the CMPSEL0 or CMPSEL1 register to set the input of the comparator.
(3) Set the CMPCTL.CINV and CDFS[1:0] bits and the CMPCTL2.CNFS bit to select inversion or non-inversion processing and the sampling timing and number of matches of the noise filter.
(4) Enable the edge for interrupt detection (set the CMPCTL.CEG[1:0] bits to a value other than 00b).
(5) Enable input of the comparator (set the CMPCTL.HCMPON bit to 1) and wait for the time until the comparator operation is stabilized. As for the value, refer to section 49, Electrical Characteristics.
(6) Enable output of the comparator (set the CMPCTL.COE bit to 1).

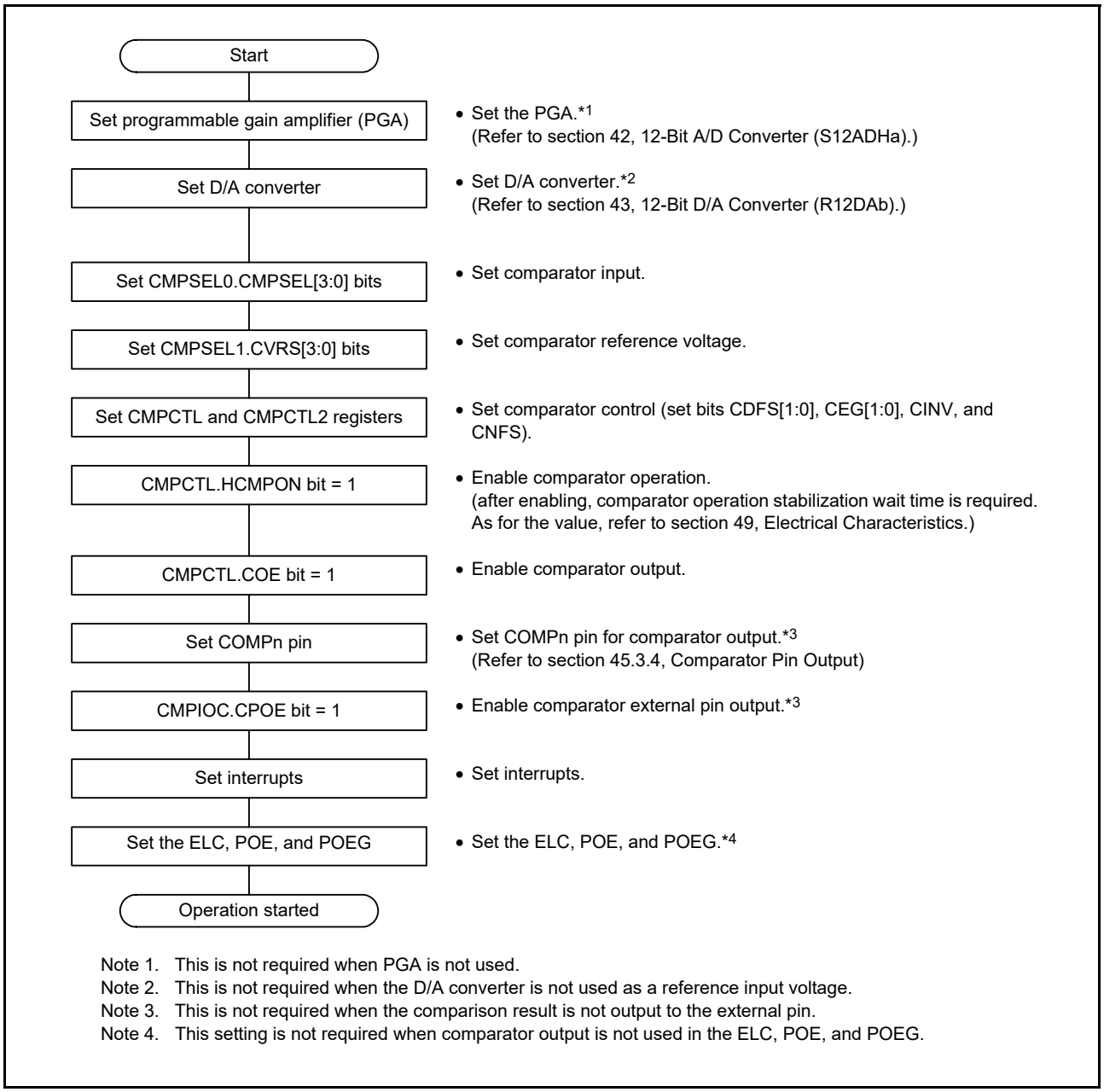
45.3.4 Comparator Pin Output

The comparison results can be output to the COMPn pins (n = 0 to 5). The CMPCTL.CINV bit can be used to set the output polarity (non-inverted output or inverted output), and the CMPIOC.CPOE bit can be used to enable or disable the output.
To output the comparison result to the external pin COMPn, use the following setting procedure. Note that the ports are set to input after reset.

(1) Execute steps (1) to (3) and steps (5) and (6) shown in section 45.3.3, Interrupts.
(2) Output the comparison result by the comparator to an external pin (set the CMPIOC.CPOE bit to 1).
(3) Set the port mode register and the pin function control register corresponding to each comparator output pin.

45.3.5 Comparator Setting Flowchart

Figure 45.5 shows the flowchart for setting the comparator-related registers.



Note 1. This is not required when PGA is not used.
Note 2. This is not required when the D/A converter is not used as a reference input voltage.
Note 3. This is not required when the comparison result is not output to the external pin.
Note 4. This setting is not required when comparator output is not used in the ELC, POE, and POEG.

Figure 45.5 Comparator Operation Setting Flowchart (n = 0 to 5)

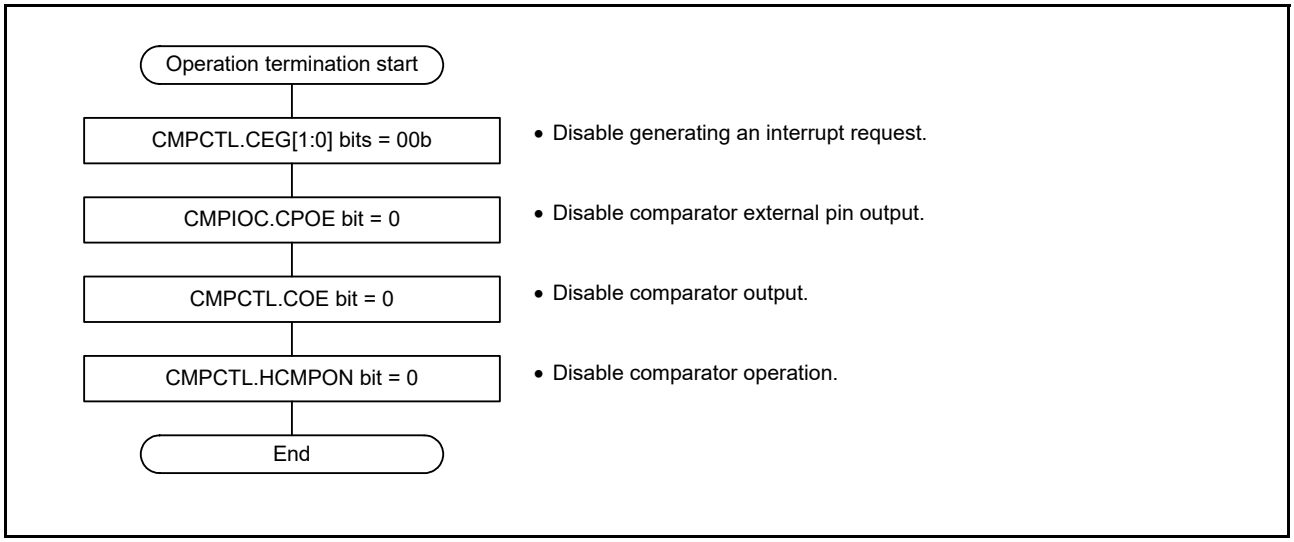


Figure 45.6 Comparator Operation Termination Flowchart

45.4 Usage Notes

45.4.1 Module Stop Function Setting

Operation of comparator C can be disabled or enabled using module stop control register B (MSTPCRB). After the reset, comparator C is halted. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

45.4.2 Comparator C Operation in Module Stop State

When the module stop state is entered while comparator C is operating, analog circuits in the comparator C is not stopped and the analog power supply current is the same as that when comparator C is being used. If the analog power supply current needs to be reduced in the module stop state, set the CMPCTL.HCMPON bit to 0 to stop comparator C.

45.4.3 Comparator C Operation in Software Standby Mode

When software standby mode is entered while comparator C is operating, analog circuits in the comparator C is not stopped and the analog power supply current is the same as that when comparator C is being used. If the analog power supply current needs to be reduced in software standby mode, set the CMPCTL.HCMPON bit to 0 to stop comparator C.

45.4.4 Comparator Operation while the 12-Bit A/D Converter is in the Module-Stop State

The same module stop signal controls the programmable gain amplifiers (PGAs) and the 12-bit A/D converter. The comparison of PGA output for the following pins is not possible while the 12-bit A/D converter is in the module stop state.

- PGA output for AN000 pin
- PGA output for AN001 pin
- PGA output for AN002 pin
- PGA output for AN100 pin
- PGA output for AN101 pin
- PGA output for AN102 pin

The comparison for the following analog input pins is not possible when the 12-bit A/D converter is in the module-stop state.

- AN000 pin
- AN001 pin
- AN002 pin
- AN100 pin
- AN101 pin
- AN102 pin

45.4.5 Setting the D/A Converter

Set the D/A converter, set the output to the comparator C using the D/A destination select register (DADSELR), and wait for the D/A converter output settling time. Similarly, before making any changes to the settings of the D/A converter stop the comparator temporarily, and after the changes are made, wait for the D/A converter output settling time before enabling the comparator.

46. Data Operation Circuit (DOCA)

46.1 Overview

The data operation circuit (DOC) is used to compare, add, or subtract 16- or 32-bit values.

Table 46.1 lists the specifications of the DOC and Figure 46.1 is a block diagram of the DOC.

An interrupt can be generated if the result of 16- or 32-bit comparison meets one of the set interrupt conditions or if the result of the addition or subtraction result of 16- or 32-bit values is an overflow or underflow.

Table 46.1 DOC Specifications

Item	Description
Data operation function	• Comparison of 16- or 32-bit values (equal or not equal, greater or less than, or within or beyond a range) • Addition or subtraction of 16- or 32-bit values
Lower power consumption function	The DOC can be placed in a module-stop state.
Interrupts	• The result of data comparison meets the detection condition. • The result of data addition is greater than FFFFh (DOCR.DOPSZ = 0) or FFFF FFFFh (DOCR.DOPSZ = 1), which is an overflow. • The result of data subtraction is less than 0000h (DOCR.DOPSZ = 0) or 0000 0000h (DOCR.DOPSZ = 1), which is an underflow.
Event link function (output)	• The result of data comparison meets the detection condition. • The result of data addition is greater than FFFFh (DOCR.DOPSZ = 0) or FFFF FFFFh (DOCR.DOPSZ = 1), which is an overflow. • The result of data subtraction is less than 0000h (DOCR.DOPSZ = 0) or 0000 0000h (DOCR.DOPSZ = 1), which is an underflow.

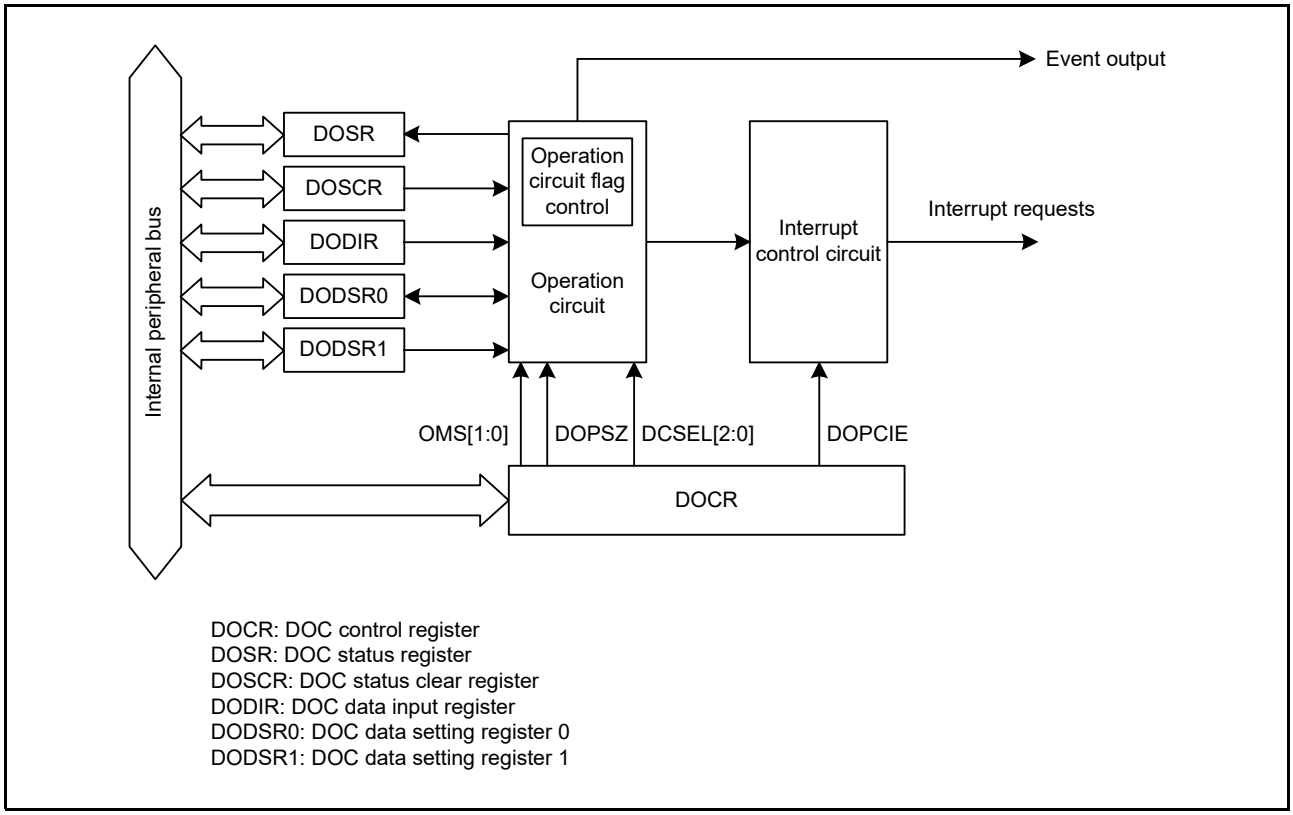
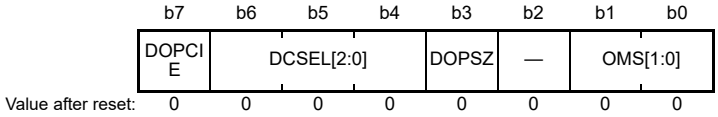


Figure 46.1 DOC Block Diagram

46.2 Register Descriptions

46.2.1 DOC Control Register (DOCR)

Address(es): DOC.DOCR 000A 0580h



Bit	Symbol	Bit Name	Description	R/W
b1, b0	OMS[1:0]	Operating Mode Select	b1 b0 0 0: Data comparison mode 0 1: Data addition mode 1 0: Data subtraction mode 1 1: Setting prohibited	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	DOPSZ	Data Operation Size Select	0: 16-bit width 1: 32-bit width	R/W
b6-b4	DCSEL[2:0]	Detection Condition Select*1	b6 b4 0 0 0: Not equal to (DODIR ≠ DODSR0) 0 0 1: Equal to (DODIR = DODSR0) 0 1 0: Less than (DODIR < DODSR0) 0 1 1: Greater than (DODIR > DODSR0) 1 0 0: Within the range (DODSR0 < DODIR < DODSR1) 1 0 1: Beyond the range (DODIR < DODSR0, DODSR1 < DODIR) Settings other than above are prohibited.	R/W
b7	DOPCIE	Data Operation Circuit Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/W

Note 1. Valid only when data comparison mode is selected.

The DOCR register specifies the operation of DOC, or enabling or disabling of the interrupt.

OMS[1:0] Bits (Operating Mode Select)

These bits select the operating mode of the DOC.

DOPSZ Bit (Data Operation Size Select)

This bit selects the size of the data operation.

DCSEL[2:0] Bits (Detection Condition Select)

This bit is valid only when data comparison mode is selected.

This bit selects the condition for detection in data comparison mode.

DOPCIE Bit (Data Operation Circuit Interrupt Enable)

Setting this bit to 1 enables interrupts from the DOC.

46.2.2 DOC Status Register (DOSR)

Address(es): DOC.DOSR 000A 0584h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DOPCF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DOPCF	Data Operation Result Flag	Indicates the result of an operation.	R
b7-b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

The DOSR register indicates the results of data operation.

DOPCF Flag (Data Operation Result Flag)

[Setting conditions]

- The setting of the DOCR.OMS[1:0] bits is 00b (data comparison mode) and the result of data comparison meets the condition selected by the DOCR.DCSEL[2:0] bits.
- The setting of the DOCR.OMS[1:0] bits is 01b (data addition mode) and the result of addition was greater than FFFFh (DOCR.DOPSZ = 0) or FFFF FFFFh (DOCR.DOPSZ = 1).
- The setting of the DOCR.OMS[1:0] bits is 10b (data subtraction mode) and the result of subtraction is below 0000h (DOCR.DOPSZ = 0) or 0000 0000h (DOCR.DOPSZ = 1).

[Clearing condition]

- Writing 1 to the DOSCR.DOPCFCL bit

46.2.3 DOC Status Clear Register (DOSCR)

Address(es): DOC.DOSCR 000A 0588h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DOPCFCL
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DOPCFCL	Data Operation Result Clear	0: Retain the value of the DOPCF flag. 1: Clears the DOPCF flag.	W
b7-b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

The DOSCR register is for clearing the DOPCF flag. It is always read as 00h.

DOPCFCL Bit (Data Operation Result Clear)

Writing 1 to this bit clears the DOSR.DOPCF flag.

46.2.4 DOC Data Input Register (DODIR)

Address(es): DOC.DODIR 000A 058Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The DODIR register is a readable and writable register that holds values for use in operations.
Access the DODIR register with the data operation size selected by the DOCR.DOPSZ bit.

46.2.5 DOC Data Setting Register 0 (DODSR0)

Address(es): DOC.DODSR0 000A 0590h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The DODSR0 register is a readable and writable register that holds values for use in comparison or the results of other operations.
Access the DODSR0 register with the data operation size selected by the DOCR.DOPSZ bit.
In data comparison mode, store the standard value for use in comparison in this register. When either ‘within the range’ (DOCR.DCSEL[2:0] = 100b) or ‘beyond the range’ (DOCR.DCSEL[2:0] = 101b) is selected, specify the lower boundary of the range.
In data addition or data subtraction mode, this register holds the results of operations.

46.2.6 DOC Data Setting Register 1 (DODSR1)

Address(es): DOC.DODSR1 000A 0594h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The DODSR1 register is a readable and writable register that holds a value for use in range comparison.
Access the DODSR1 register with the data operation size selected by the DOCR.DOPSZ bit.
When either ‘within the range’ (DOCR.DCSEL[2:0] = 100b) or ‘beyond the range’ (DOCR.DCSEL[2:0] = 101b) is selected in data comparison mode, specify the upper boundary of the range.
This register is only used when either ‘within the range’ or ‘beyond the range’ is selected.

46.3 Operation

46.3.1 Data Comparison Mode

Figure 46.2 to Figure 46.7 show an example of the steps involved in data comparison mode operation by the DOC*1. An example of operation with the operation size of 32 bits is shown below.

(1) Writing 00b to the DOCR.OMS[1:0] bits places the DOC in the data comparison mode. At the same time, write to the DOCR.DCSEL[2:0] bits to select the condition to be detected.
(2) Specify the standard values for comparison in the DODSR0 and DODSR1 registers.*2
(3) Write the value for comparison in the DODIR register.
(4) If the value written to the DODIR register satisfies the condition set in the DOCR.DCSEL[2:0] bits, the DOSR.DOPCF flag becomes 1, and an ELC event is generated. If the DOCR.DOPCIE bit is 1, a data operation circuit interrupt is also issued.

Note 1. Comparison is made to proceed at the same time a value is written to the DODIR register. Writing values to the DODSR0 and DODSR1 registers does not make comparison proceed.
Note 2. Setting of the DODSR1 register is only required when either 'within the range' or 'beyond the range' is selected. Set a greater value for the DODSR1 register than that of the DODSR0 register.

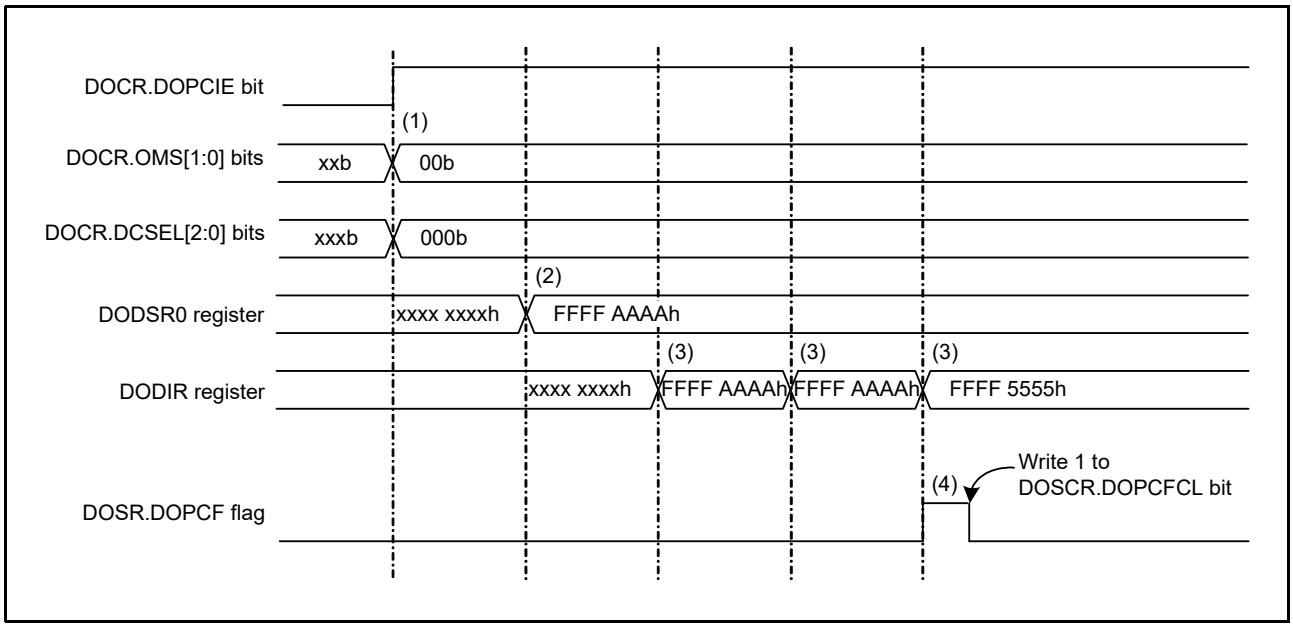


Figure 46.2 Example of the Operation of 'Not Equal to' as the Detection Condition in Data Comparison Mode

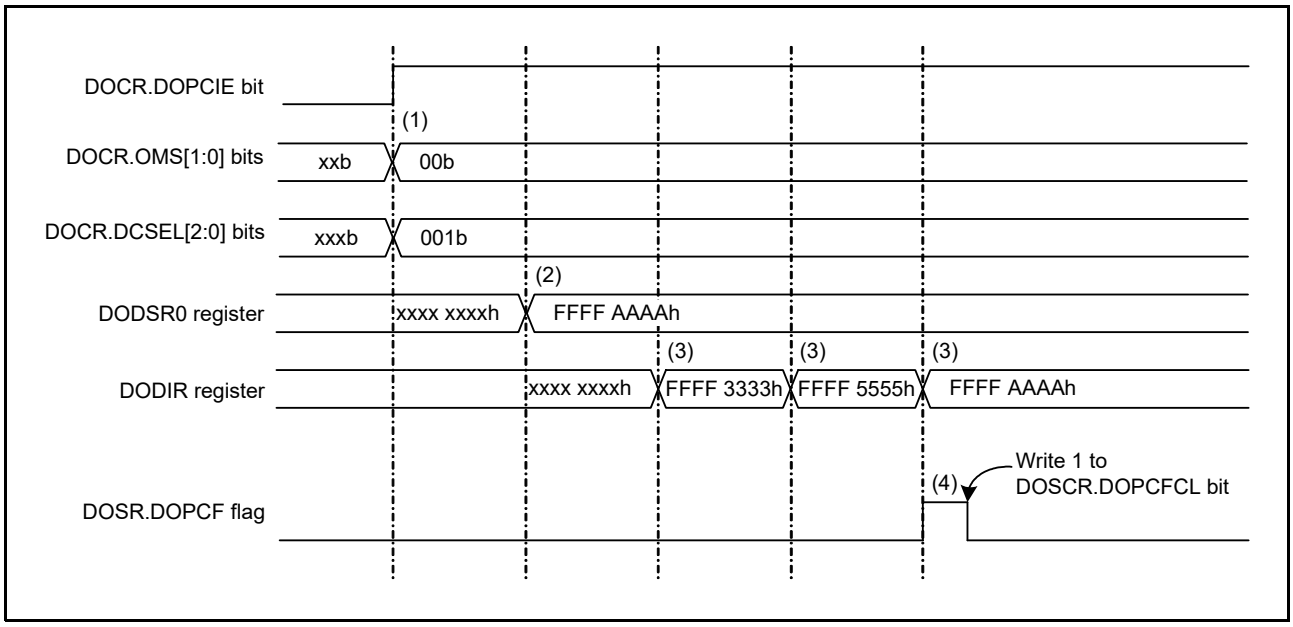


Figure 46.3 Example of the Operation of 'Equal to' as the Detection Condition in Data Comparison Mode

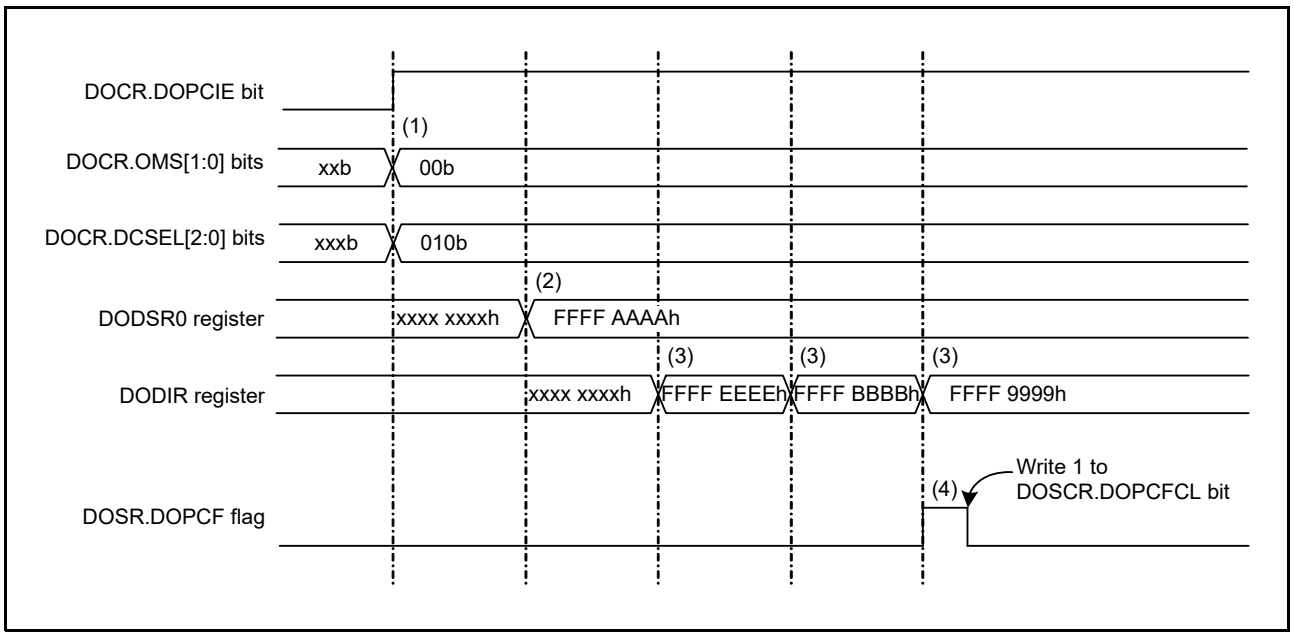


Figure 46.4 Example of the Operation of 'Less Than' as the Detection Condition in Data Comparison Mode

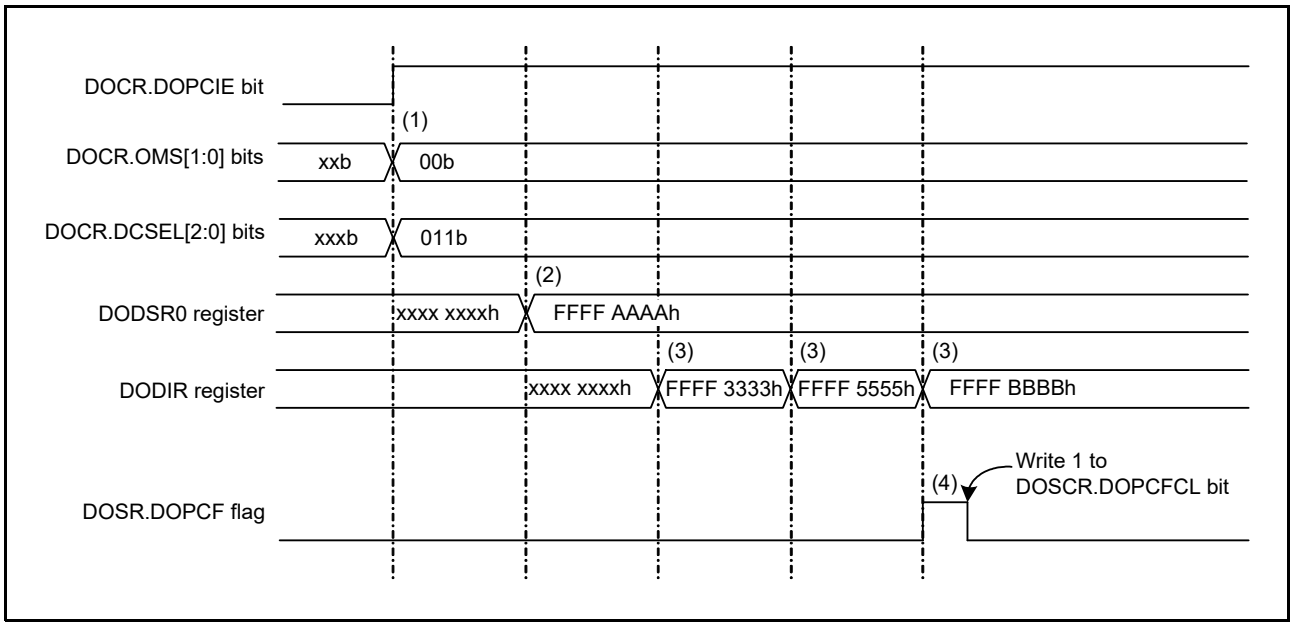


Figure 46.5 Example of the Operation of 'Greater Than' as the Detection Condition in Data Comparison Mode

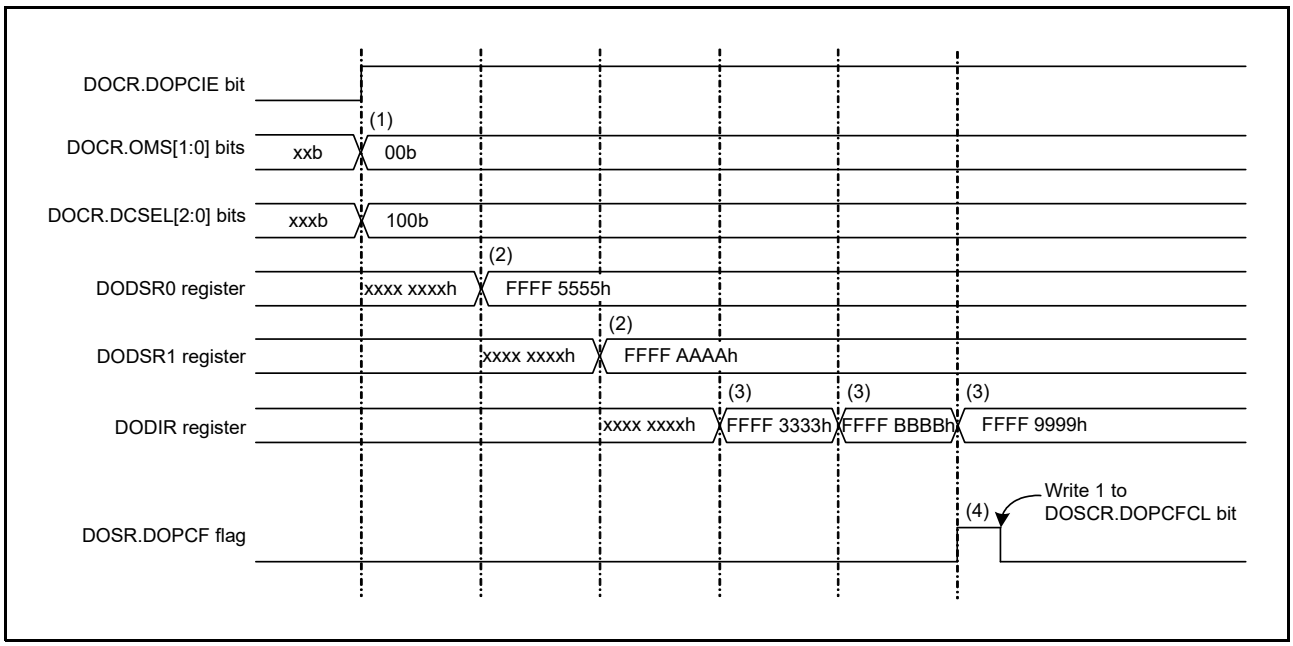


Figure 46.6 Example of the Operation of 'Within the Range' as the Detection Condition in Data Comparison Mode

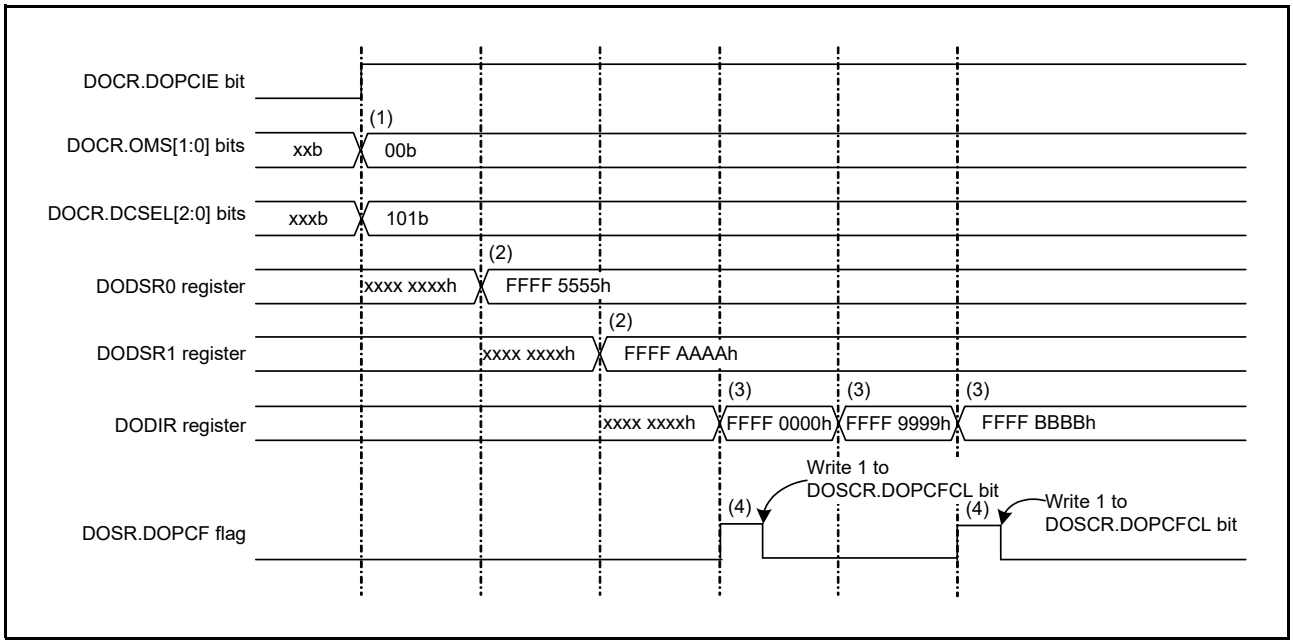


Figure 46.7 Example of the Operation of 'Beyond the Range' as the Detection Condition in Data Comparison Mode

46.3.2 Data Addition Mode

Figure 46.8 shows an example of the steps involved in data addition mode operation*1 by the DOC.
An example of operation when the data operation size is 32 bits is shown below.

(1) Writing 01b to the DOCR.OMS[1:0] bits selects data addition mode.
(2) Set the initial value in the DODSR0 register.
(3) Write the value for addition in the DODIR register. The result of the operation is stored in DODSR0.
(4) Write all values for use in addition to the DODIR register.
(5) If the result of the operation is greater than FFFF FFFFh, the DOSR.DOPCF flag becomes 1, and an ELC event is generated. If the DOCR.DOPCIE bit is 1, a data operation circuit interrupt is also issued.

Note 1. Addition is made to proceed at the same time as a value is written to the DODIR register. Writing a value to the DODSR0 register does not make addition proceed.

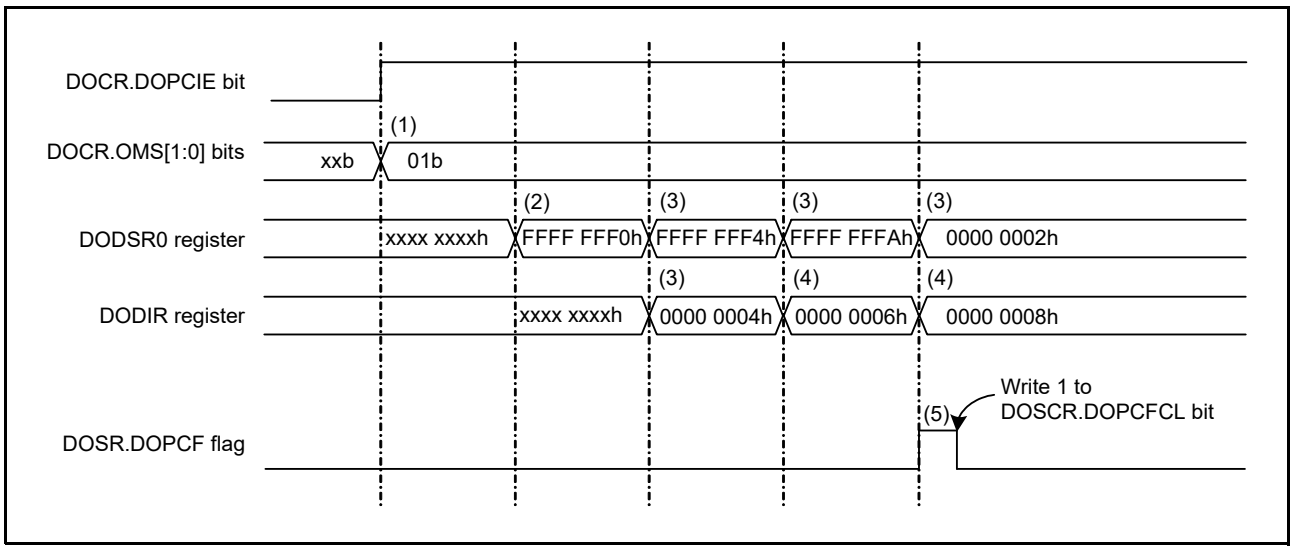


Figure 46.8 Example of Operation in Data Addition Mode

46.3.3 Data Subtraction Mode

Figure 46.9 shows an example of the steps involved in data subtraction mode operation*1 by the DOC.
An example of operation when the data operation size is 32 bits is shown below.

(1) Writing 10b to the DOCR.OMS[1:0] bits selects data subtraction mode.
(2) Set the initial value in the DODSR0 register.
(3) Write the value for subtraction in the DODIR register. The result of the operation is stored in DODSR0.
(4) Write all values for use in subtraction to the DODIR register.
(5) If the result of the operation is less than 0000 0000h, the DOSR.DOPCF flag becomes 1, and an ELC event is generated. If the DOCR.DOPCIE bit is 1, a data operation circuit interrupt is also issued.

Note 1. Subtraction is made to proceed at the same time a value is written to the DODIR register. Writing a value to the DODSR0 register does not make subtraction proceed.

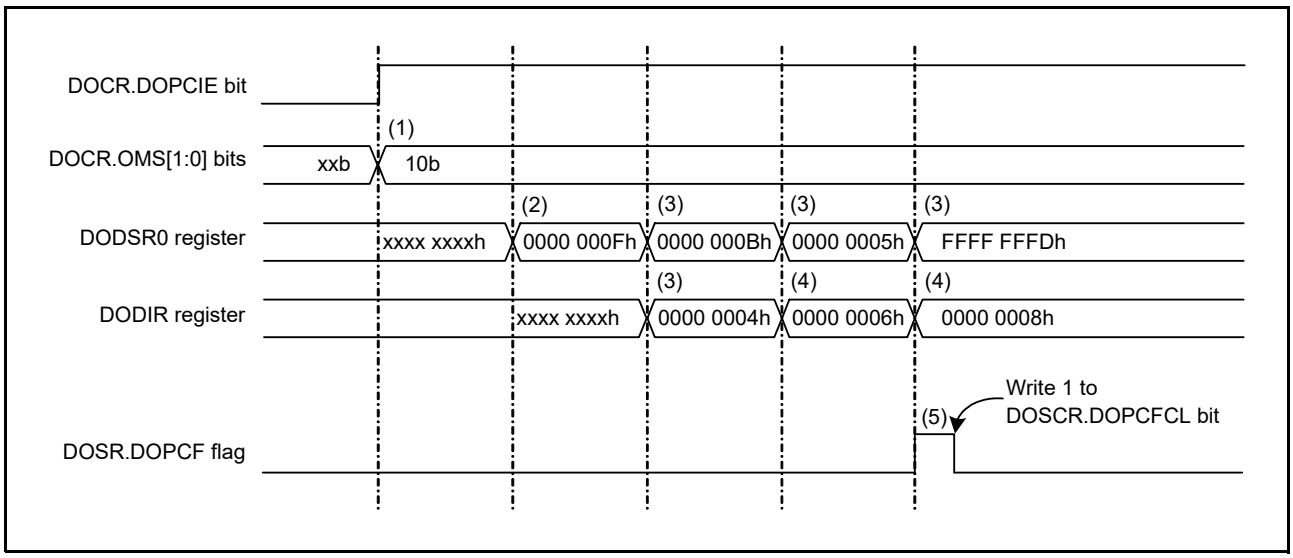


Figure 46.9 Example of Operation in Data Subtraction Mode

46.4 Interrupt Requests

The data operation circuit interrupt (DOPCI) is the interrupt request generated by the DOC. The DOSR.DOPCF flag becomes 1 when the interrupt source condition is satisfied, and an interrupt request is also issued if the DOCR.DOPCIE bit is 1.
Table 46.2 lists the details of the interrupt request.

Table 46.2 Interrupt Request from DOC

Interrupt Request	Data Operation Result Flag	Interrupt Generation Timing
Data operation circuit interrupt (DOPCI)	DOPCF	• The result of data comparison meets the detection condition. • The result of data addition is greater than FFFFh (DOCR.DOPSZ = 0) or FFFF FFFFh (DOCR.DOPSZ = 1). • The result of data subtraction is less than 0000h (DOCR.DOPSZ = 0) or 0000 0000h (DOCR.DOPSZ = 1).

46.5 Event Link Output

The DOC outputs event signals for the event link controller (ELC) under the following conditions, and these can be used to initiate operations by other modules selected in advance.

- The result of data comparison meets the detection condition.
- The result of data addition is greater than FFFFh (DOCR.DOPSZ = 0) or FFFF FFFFh (DOCR.DOPSZ = 1).
- The result of data subtraction is less than 0000h (DOCR.DOPSZ = 0) or 0000 0000h (DOCR.DOPSZ = 1).

46.5.1 Interrupt Handling and Event Linking

The DOC has a bit to enable or disable interrupts. When an interrupt source condition is satisfied while the interrupt is enabled, the interrupt request signal is issued to the CPU.
In contrast, an event link output signal is sent to other modules as an event signal via the ELC when an interrupt source is generated, regardless of the setting of the corresponding interrupt enable bit.

46.6 Usage Note

46.6.1 Module Stop Function Setting

Operation of the DOC can be enabled or disabled by setting the MSTPB6 bit in module stop control register B (MSTPCRB). The DOC is initially disabled after a reset. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

47. RAM

This MCU has a 64-Kbyte high-speed static RAM (RAM), which runs at the frequency of 120 MHz in the no-wait state.

47.1 Overview

Table 47.1 lists the specifications of the RAM.

Table 47.1 Specifications of RAM

Item	RAM
Capacity	64 Kbytes, 48 Kbytes
Address	• Products with 64 Kbytes of RAM 0000 0000h to 0000 FFFFh • Products with 48 Kbytes of RAM 0000 0000h to 0000 BFFFh
Memory bus	Memory bus 1
Access	• Single-cycle access is possible for both reading and writing.*1 • Enabling or disabling of the RAM is selectable.*2
Low power consumption function	Transitions to the module stop state are possible.
Error checking	• Parity check: Detection of 1-bit errors • A non-maskable interrupt or interrupt is generated in response to an error.

Note 1. When accessing across the 8-byte boundary, the number of cycles is doubled.
Note 2. Selectable by the RAME bit in SYSCR1. For details on SYSCR1, see section 3.2.2, System Control Register 1 (SYSCR1).

47.2 Register Descriptions

47.2.1 RAM Operating Mode Control Register (RAMMODE)

Address(es): RAM.RAMMODE 0008 1200h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	RAMMODE[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	RAMMODE[1:0]	RAM Operating Mode Select	b1 b0 0 0: Parity checking is disabled. 0 1: Parity checking is enabled. Settings other than above are prohibited.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R

The RAMMODE register is write-protected by the RAM protection register (RAMPRCR). Before writing to the RAMMODE register, set the RAMPRCR.RAMPRCR bit to 1 to enable writing to it. Set the RAMMODE register before starting access to the RAM. If this register is modified after accessing to the RAM, RAM operation is not guaranteed.

47.2.2 RAM Error Status Register (RAMSTS)

Address(es): RAM.RAMSTS 0008 1201h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	RAMERR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RAMERR	RAM Error Status Flag	0: A parity check error has not occurred. 1: A parity check error has occurred.	R/(W)*1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Only 0 can be written to clear the flag.

When parity checking is enabled, the RAMERR flag is set to 1 if a parity check error is detected. The RAM error interrupt request is also generated at this time.
When parity checking is disabled, the RAMERR flag is not set to 1 because no parity check error is detected.
Writing 0 to the RAMERR flag clears the RAM error interrupt request corresponding to the parity check error.

47.2.3 RAM Error Address Capture Register (RAMECAD)

Address(es): RAM.RAMECAD 0008 1208h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	READ		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	READ													—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0.	R
b18 to b3	READ	Error Address	The address where an error is found is read.	R
b31 to b19	—	Reserved	These bits are read as 0.	R

When parity checking is enabled, this register will hold the address where a parity check error was found.
The address of the 8-byte boundary below the location where the error was found is stored in this register at the same time the RAMSTS.RAMERR flag is set to 1.
The error address is not updated when the RAMERR flag is 1 (error has occurred). Its value does not change when parity checking is disabled because no parity check error is detected.
The RAMECAD register is initialized only by a reset.

47.2.4 RAM Protection Register (RAMPRCR)

Address(es): RAM.RAMPRCR 0008 1204h

	b7	b6	b5	b4	b3	b2	b1	b0
	KW[6:0]							RAMPRCR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RAMPRCR	RAMMODE Register Write Control	0: Disables writing to the RAMMODE register. 1: Enables writing to the RAMMODE register.	R/W
b7 to b1	KW[6:0]	Write Key Word	Enables or disables the rewriting of the RAMPRCR register. When rewriting the RAMPRCR register, write 1111000b to the KW[6:0] bits.	R/W

Writing 1 to the RAMPRCR bit is possible when KW[6:0] = 1111000b. Otherwise writing to RAMPRCR clears the bit to 0. The value of KW[6:0] is read as 0000000b.
The targets for write protection by the RAMPRCR register is the RAM operating mode control register (RAMMODE). Once the RAMPRCR bit is set to 1, writing to RAMMODE register is enabled until the RAMPRCR bit is cleared to 0. Clear the RAMPRCR bit to 0 after writing to RAMMODE register.

47.3 Operation

47.3.1 Parity Checking

Enabling and disabling of parity checking can be selected through the RAMMODE register setting. In the initial state, parity checking is disabled. Even parity checking is used in this device.
1-bit parity check code is added to each 1-byte data for writing, and the parity is checked for reading.
If a 1-bit error is detected in the 1 byte when the parity is checked for reading, a RAM error interrupt can be generated. If a 2-bit error or more is detected in the 1 byte, errors cannot be correctly detected.
After power-on, parity check code is undefined until data is written. To use parity checking, write the initial value to all areas while parity checking is enabled before accessing to the RAM immediately after a reset.
Operation cannot be guaranteed if access is made to an area where the initial value is not written.

47.3.2 RAM Error Interrupt Function

A RAM error interrupt is generated when the RAMSTS.RAMERR bit that indicates a parity check error has been changed to 1 while parity checking is enabled.
Writing 0 to the bit clears the RAM interrupt.

47.3.3 Interrupt Source

Of the RAM interrupt sources, that due to the detection of an error through parity checking can be used as either a non-maskable interrupt or a maskable interrupt. For details, see section 14, Interrupt Controller (ICUG).

Table 47.2 RAM Interrupt Source

Name	Interrupt Source	DTC Activation	DMAC Activation
RAMERR	RAM error	Not possible	Not possible

47.4 Usage Notes

47.4.1 Low Power Consumption Function

Power consumption can be reduced by setting module stop control register C (MSTPCRC) to stop supply of the clock signal to the RAM.
Setting the MSTPCRC.MSTPC0 bit to 1 stops supply of the clock signal to the RAM.
Stopping supply of the clock signal places the RAM in the module stop state.
The RAM operates after a reset.
The RAM is not accessible in the module stop state.
Do not allow transitions to the module stop state while accessing to the RAM.
Access to the RAM in the module stop state is prohibited. If access is attempted, correct operation is not guaranteed.
For details on the MSTPCRC register, refer to section 11, Low Power Consumption.

47.4.2 Notes on Using Error Checking of RAM

Data in RAM are undefined when the power is turned on. Therefore, parity check errors occur if the data are read before initialization. The RAM is read in 8-byte (64-bit) units. Initialize it on 8-byte boundaries.
When a program is executed in the RAM with the parity check enabled, initialize the RAM in consideration of possible instruction prefetching by the CPU. Instruction prefetching can be performed up to 32 bytes. The initialization must thus cover extra 24 to 31 bytes from the last address of the program.

47.4.3 Notes on Self-Diagnosis of the RAM

A write buffer is mounted for the RAM. When the same address is read after a write operation, data in the write buffer, rather than in the memory cell of the RAM may be read. When the RAM is self-diagnosed, confirm that the data have been written by following the procedure below so that data will not be read from the write buffer.

(1) Write data to the address targeted for diagnosis.
(2) Write data to an address which is at least 4 addresses away from the that in (1).
(3) Read the data from the address in (1).

48. Flash Memory (FLASH)

This MCU incorporates code flash memory, data flash memory, and option-setting memory.
The code flash memory stores instructions and operands, and the data flash memory stores only data. For option-setting memory, refer to section 7, Option-Setting Memory (OFSM).

48.1 Overview

Table 48.1 lists the specifications of the code flash memory/data flash memory, and Figure 48.1 is a block diagram of the flash memory related modules.
The I/O pins used in boot mode, refer to Table 48.18.
The flash sequencer controls the program and erase operation of the flash memory. The FACI (flash application command interface) controls the flash sequencer according to the specified FACI commands.
Regarding the configuration of the code flash memory, refer to Figure 48.2, and for the configuration of the data flash memory, refer to Figure 48.4.

Table 48.1 Specifications of Code Flash Memory and Data Flash Memory

Item	Code Flash Memory	Data Flash Memory
Memory capacity	Up to 512 Kbytes	16 Kbytes
Read cycle	One cycle	A read operation takes eight cycles of FCLK in 16- or 8-bit access
Value after erase	FFh	Undefined
Program/erase method	• Code flash memory and data flash memory can be programmed or erased with the FACI commands set in the FACI command issuing area (007E 0000h). • Program/erase via a serial interface transfer by a flash-memory programmer (serial programming) • Program/erase of flash memory by user program (self-programming)	
Security function	Protects against illicit tampering with or reading out of data in flash memory	
Protection function	Protects against erroneous rewriting of the flash memory	
Dual bank function	The dual-bank structure makes a safe update possible in cases where program operation is suspended. • Linear mode: the code flash memory is used as one area • Dual mode: the code flash memory is divided into two areas	Not available
Trusted memory (TM) function	Protects against illicit reading of the code flash memory • Linear mode: blocks 8 and 9 • Dual mode: blocks 8, 9, 30, and 31	Not available
Background operations (BGOs)*1	• The code flash memory can be read while the code flash memory is being programmed or erased.*2 • The data flash memory can be read while the code flash memory is being programmed or erased. • The code flash memory can be read while the data flash memory is being programmed or erased.	
Program/erase unit	• Program unit: 128 bytes • Erase unit: per block	• Program unit: 4 bytes • Erase unit: per block
Other functions	Interrupts can be accepted during self-programming.	
On-board programming (serial programming/ self-programming)	Program/erase in boot mode (SCI interface) • The asynchronous serial interface (SCI1) is used. • The transfer rate is adjusted automatically. Program/erase in boot mode (FINE interface) • FINE is used. Program/erase in single-chip mode • Code flash memory and data flash memory can be programmed or erased by the rewrite routine in the user program.	
Unique ID	A 12-byte ID code provided for each MCU	

Note 1. The BGO function cannot be used to rewrite the configuration setting area.
Note 2. Limitations apply to the combinations of the address ranges for program/erase process and reading process. Refer to Table 48.24.

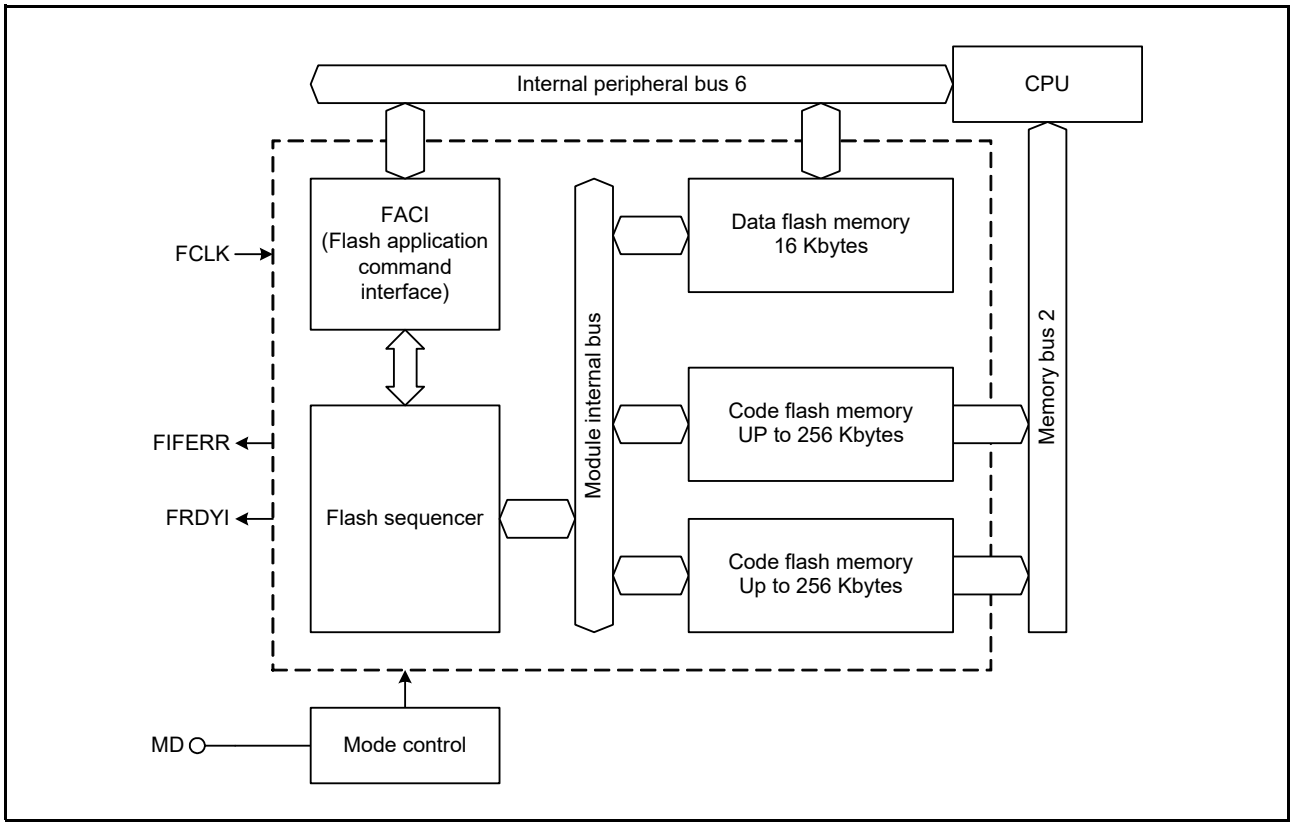


Figure 48.1 Block Diagram of Flash Memory Related Modules

48.2 Hardware Interface Area

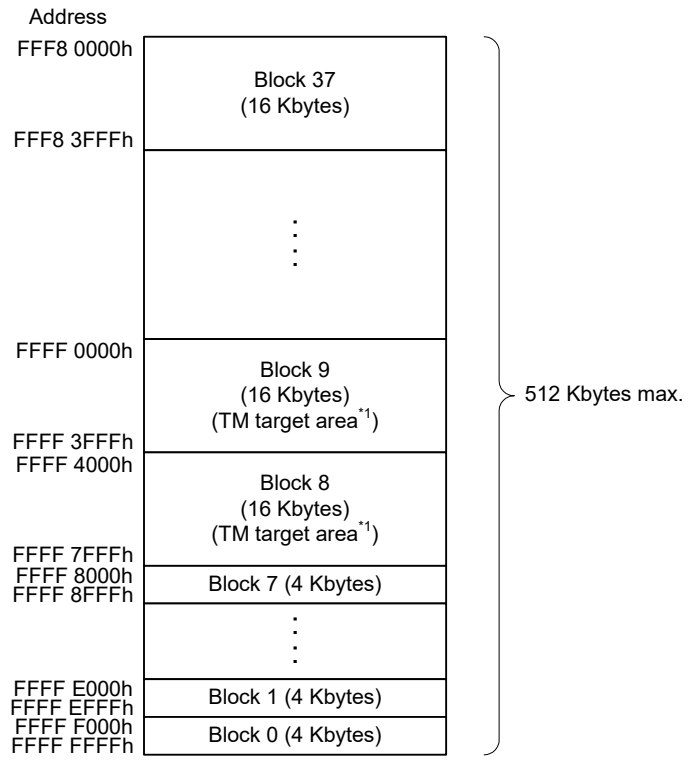
Using the hardware interface with the flash memory requires accessing to the area containing registers of the hardware, that for the issuing of FACI commands. Table 48.2 summarizes information on all of these areas.

Table 48.2 Information on the Hardware Interface Area

Area	Address	Capacity
Area containing the various registers of the hardware	Refer to section 48.4, Register Descriptions.	Refer to section 48.4, Register Descriptions.
FACI command-issuing area	007E 0000h	4 bytes
Configuration setting area	0012 0040h to 0012 00FFh	192 bytes

48.3 Structure of Memory

Figure 48.2 shows the mapping of the code flash memory in linear mode. Figure 48.3 show the memory map of code flash memory in dual mode. The code flash memory can be used as 2 bank areas by using the dual bank function. This dual-bank structure allows a safe update of a program while a user program is running.
The code flash memory in this MCU is divided into 4- and 16-Kbyte blocks, which are used as erase units. The code flash memory is available as an area for storing the user program.
When the TM function is enabled in linear mode, blocks 8 and 9 are the TM target areas.

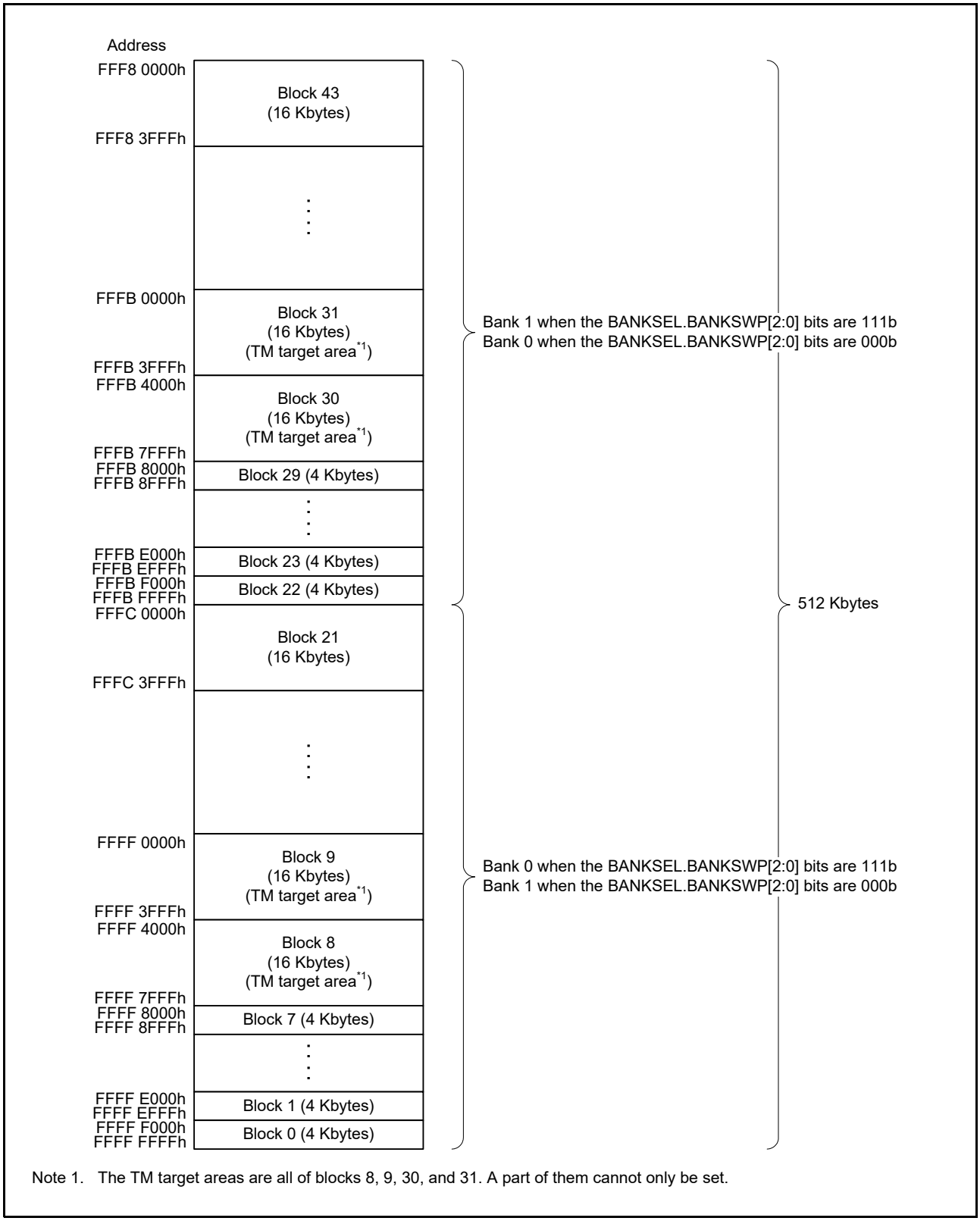


Code Flash Memory Capacity	Address	Number of Blocks
512 Kbyte	FFF8 0000h to FFFF FFFFh	0 to 37
256 Kbytes	FFFC 0000h to FFFF FFFFh	0 to 21
128 Kbytes	FFFE 0000h to FFFF FFFFh	0 to 13

Note 1. Both blocks 8 and 9 are the target areas for handling as TM. Setting only block 8 or only block 9 is not allowed.

Figure 48.2 Mapping of the Code Flash Memory in Linear Mode

When the TM function is enabled in dual mode, blocks 8 and 9, and blocks 30 and 31 are the TM target areas.



Note 1. The TM target areas are all of blocks 8, 9, 30, and 31. A part of them cannot only be set.

Figure 48.3 Mapping of the Code Flash Memory in Dual Mode (Products with 512 Kbytes of code flash memory)

The data flash memory in this MCU is divided into 64-byte blocks, each of which is an erase unit. Figure 48.4 shows the mapping of the data flash memory.

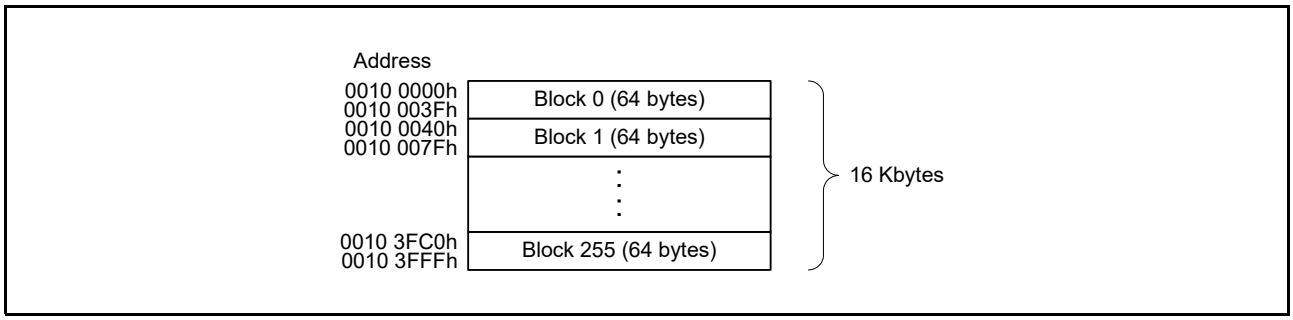


Figure 48.4 Mapping of the Data Flash Memory

48.4 Register Descriptions

48.4.1 Flash P/E Protect Register (FWEPROR)

Address(es): FLASH.FWEPROR 0008 C296h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	FLWE[1:0]	
Value after reset:	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	FLWE[1:0]	Flash Program/Erase Enable	b1 b0 0 0: Program, block erase, and blank check are disabled. 0 1: Program, block erase, and blank check are enabled. 1 0: Program, block erase, and blank check are disabled. 1 1: Program, block erase, and blank check are disabled.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register enables or disables the following three FACI commands for flash memory: program, block erase, and blank check.
This register is initialized not only by a reset but also by a transition to software standby mode.

48.4.2 Flash Access Status Register (FASTAT)

Address(es): FLASH.FASTAT 007F E010h

	b7	b6	b5	b4	b3	b2	b1	b0
	CFAE	—	—	CMDLK	DFAE	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	DFAE	Data Flash Memory Access Violation Flag	0: No data flash memory access violation has occurred. 1: A data flash memory access violation has occurred.	R/W*1
b4	CMDLK	Command Lock Flag	0: The flash sequencer is not in the command-locked state. 1: The flash sequencer is in the command-locked state.	R
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	CFAE	Code Flash Memory Access Violation Flag	0: No code flash memory access violation has occurred. 1: A code flash memory access violation has occurred.	R/W*1

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

This register indicates whether a code flash memory or data flash memory access violation has occurred. If either of the CFAE, and DFAE flags is 1, the CMDLK flag is set to 1 and the flash sequencer enters the command-locked state (refer to section 48.5.3.2, Error Protection). To release it from the command-locked state, a status clear command or forced stop command must be issued by the FACI after clearing the CFAE and DFAE flags in the FASTAT register to 0.

DFAE Flag (Data Flash Memory Access Violation Flag)

This flag indicates whether a data flash memory access violation has occurred. If this flag is set to 1, the FSTATR.ILGLERR flag is set to 1, placing the flash sequencer in the command-locked state.

[Setting Condition]

- Refer to Table 48.10, Error Protection Type

[Clearing Condition]

- When 0 is written after reading of 1

CMDLK Flag (Command Lock Flag)

This flag indicates that the flash sequencer is in the command-locked state.

[Setting Condition]

- When the flash sequencer detects any of errors listed in Table 48.10, Error Protection Type and transitions to the command-locked state

[Clearing Condition]

- When the flash sequencer starting to process a status clear or forced stop command while the CFAE or DFAE flag in the FASTAT register is 0

CFAE Flag (Code Flash Memory Access Violation Flag)

This flag indicates whether a code flash memory access violation has occurred. If this flag is set to 1, the FSTATR.ILGLERR flag is set to 1, placing the flash sequencer in the command-locked state.

[Setting Conditions]

- Refer to Table 48.10, Error Protection Type

[Clearing Condition]

- When 0 is written after reading of 1

48.4.3 Flash Access Error Interrupt Enable Register (FAEINT)

Address(es): FLASH.FAEINT 007F E014h

	b7	b6	b5	b4	b3	b2	b1	b0
	CFAEIE	—	—	CMDLKIE	DFAEIE	—	—	—
Value after reset:	1	0	0	1	1	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	After a reset is released, write 0 to this bit.	R/W
b2, b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	DFAEIE	Data Flash Memory Access Violation Interrupt Enable	0: Generation of an FIFERR interrupt request is disabled when FASTAT.DFAE is set to 1. 1: Generation of an FIFERR interrupt request is enabled when FASTAT.DFAE is set to 1.	R/W
b4	CMDLKIE	Command Lock Interrupt Enable	0: Generation of an FIFERR interrupt request is disabled when FASTAT.CMDLK is set to 1. 1: Generation of an FIFERR interrupt request is enabled when FASTAT.CMDLK is set to 1.	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	CFAEIE	Code Flash Memory Access Violation Interrupt Enable	0: Generation of an FIFERR interrupt request is disabled when FASTAT.CFAE is set to 1. 1: Generation of an FIFERR interrupt request is enabled when FASTAT.CFAE is set to 1.	R/W

This register enables or disables generation of a flash access error (FIFERR) interrupt request.

DFAEIE Bit (Data Flash Memory Access Violation Interrupt Enable)

This bit enables or disables generation of an FIFERR interrupt request when a data flash memory access violation occurs and the FASTAT.DFAE flag is set to 1.

CMDLKIE Bit (Command Lock Interrupt Enable)

This bit enables or disables generation of an FIFERR interrupt request when the flash sequencer enters the command-locked state and the FASTAT.CMDLK flag is set to 1.

CFAEIE Bit (Code Flash Memory Access Violation Interrupt Enable)

This bit enables or disables generation of an FIFERR interrupt request when a code flash memory access violation occurs and the FASTAT.CFAE flag is set to 1.

48.4.4 Flash Ready Interrupt Enable Register (FRDYIE)

Address(es): FLASH.FRDYIE 007F E018h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	FRDYIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	FRDYIE	Flash Ready Interrupt Enable	0: Generation of an FRDY interrupt request is disabled. 1: Generation of an FRDY interrupt request is enabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

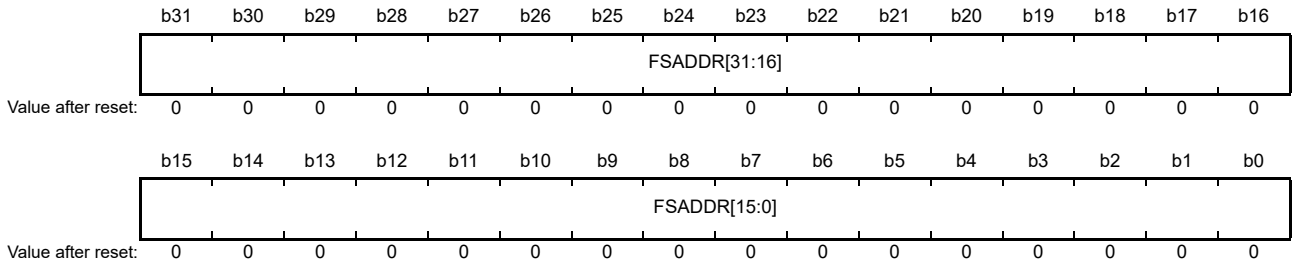
This register enables or disables generation of a flash ready (FRDY) interrupt request.

FRDYIE Bit (Flash Ready Interrupt Enable)

This bit is used to enable or disable generation of an FRDY interrupt request when the FASTAT.FRDY flag is changed from 0 to 1 upon completion of processing by the flash sequencer of program, block erase, and blank check command.

48.4.5 FACI Command Processing Start Address Register (FSADDR)

Address(es): FLASH.FSADDR 007F E030h



Bit	Symbol	Bit Name	Description	R/W
b31 to b0	FSADDR[31:0]	FACI Command Processing Start Address	Start Address for FACI Command Processing	R/W*1

Note 1. Writing to these bits is possible only when the FSTATR.FRDY flag is 1. Writing to these bits while the FSTATR.FRDY flag = 0 is ignored. The value of the lower 2 bits (b1 and b0) is fixed to 00b.

This register specifies the address where the target area for command processing starts when the FACI command for program, block erase, blank check, or configuration set is issued.
The FSADDR register is initialized when the FSUINITR.SUINIT bit is set to 1. It is also initialized by a reset.

FSADDR[31:0] Bits (FACI Command Processing Start Address)

These bits specify the start address for FACI command processing. Bits 31 to 24 are ignored in FACI command processing for the code flash memory. Bits 31 to 19 are ignored in FACI command processing for the data flash memory. Bits that do not reach the address boundaries are also ignored. Table 48.3 shows the address boundary for each command.

Table 48.3 Address Boundary for Each of the Commands

Command	Address Boundary
Program (code flash memory)	128-byte
Program (data flash memory)	4-byte
Block erase (code flash memory)	4-Kbyte or 16-Kbyte
Block erase (data flash memory)	64-byte
Blank check	4-byte
Configuration set	16-byte

Refer to Table 48.17, Address Used by Configuration Set Command for the start address of the configuration setting area.

48.4.6 FACI Command Processing End Address Register (FEADDR)

Address(es): FLASH.FEADDR 007F E034h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	FEADDR[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FEADDR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	FEADDR[31:0]	FACI Command Processing End Address	The end address for FACI command processing	R/W*1

Note 1. Writing to these bits is possible only when the FSTATR.FRDY flag is 1. Writing to these bits while the FSTATR.FRDY flag = 0 is ignored. The value of the lower 2 bits (b1 and b0) is fixed to 00b.

This register is used to specify the end address of the area targeted for the blank check command handling. When the FBCCNT.BCDIR bit is 0, the setting value of the FSADDR register must be the setting of the FEADDR register or lower. When the FBCCNT.BCDIR bit is 1, the value of the FSADDR register must be at least that of the FEADDR register. If the settings of the FBCCNT.BCDIR bit and the FSADDR and FEADDR registers are inconsistent with the above rules, the flash sequencer enters the command-locked state (refer to section 48.5.3.2, Error Protection).
The FEADDR register is initialized when the FSUINITR.SUINIT bit is set to 1. It is also initialized by a reset.

FEADDR[31:0] Bits (FACI Command Processing End Address)

These bits are used to specify the end address for handling of the blank check command. In command processing, bits 31 to 19, 1, and 0 are ignored.

48.4.7 Flash Status Register (FSTATR)

Address(es): FLASH.FSTATR 007F E080h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FRDY	ILGLERR	ERSERR	PRGERR	SUSRDY	DBFULL	ERSSPD	PRGSPD	—	FLWEERR	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	FLWEERR	Flash P/E Protection Error Flag	0: An error has not occurred. 1: An error has occurred.	R
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b8	PRGSPD	Program Suspend Status Flag	0: The flash sequencer is in a state other than those corresponding to the value 1. 1: The flash sequencer is in the program suspend processing state or the program-suspended state.	R
b9	ERSSPD	Erase Suspend Status Flag	0: The flash sequencer is in a state other than those corresponding to the value 1. 1: The flash sequencer is in the erase suspend processing state or the erase-suspended state.	R
b10	DBFULL	Data Buffer Full Flag	0: The data buffer is empty. 1: The data buffer is full.	R
b11	SUSRDY	Suspend Ready Flag	0: The flash sequencer cannot receive P/E suspend commands. 1: The flash sequencer can receive P/E suspend commands.	R
b12	PRGERR	Program Error Flag	0: Program operation has been completed successfully. 1: An error has occurred during program operation.	R
b13	ERSERR	Erase Error Flag	0: Erase operation has been completed successfully. 1: An error has occurred during erase operation.	R
b14	ILGLERR	Illegal Command Error Flag	0: The flash sequencer has not detected an illegal FACI command or illegal flash memory access. 1: The flash sequencer has detected an illegal FACI command or illegal flash memory access.	R
b15	FRDY	Flash Ready Flag	0: Programming, block erase, P/E suspend, P/E resume, forced stop, blank check, or configuration set command processing is in progress. 1: None of the above is in progress.	R
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register indicates the state of the flash sequencer.

FLWEERR Flag (Flash P/E Protection Error Flag)

This flag indicates a violation of the flash memory program/erase protection setting in the FWEPROR register. When this flag is 1, the flash sequencer is in the command-locked state.

[Setting Condition]

- Refer to Table 48.10, Error Protection Type

[Clearing Condition]

- When the flash sequencer starts processing of a forced stop command

PRGSPD Flag (Program Suspend Status Flag)

This flag indicates that the flash sequencer is in the processing of program suspend or has transitioned to the program-suspended state.

[Setting Condition]

- When the flash sequencer starts processing in response to a program suspend command

[Clearing Conditions]

- When the flash sequencer has received a P/E resume command
  (after write access to the FACI command-issuing area is completed)
- When the flash sequencer starts processing of a forced stop command

ERSSPD Flag (Erase Suspend Status Flag)

This flag indicates that the flash sequencer is the processing of erase suspend or has transitioned to the erase-suspended state.

[Setting Condition]

- When the flash sequencer starts processing in response to an erase suspend command

[Clearing Conditions]

- Reception of the P/E resume command by the flash sequencer
  (after write access to the FACI command-issuing area is completed)
- When the flash sequencer starts processing of a forced stop command

DBFULL Flag (Data Buffer Full Flag)

This flag indicates the state of the data buffer when a program command is issued. The FACI incorporates a buffer for write data (data buffer). When data for writing to the flash memory are issued to the FACI command-issuing area while the data buffer is full, the FACI inserts a wait cycle in the peripheral bus 6.

[Setting Condition]

- When the data buffer becomes full while a program command is being issued

[Clearing Condition]

- When the data buffer becomes empty

SUSRDY Flag (Suspend Ready Flag)

This flag indicates whether the flash sequencer can receive a P/E suspend command.

[Setting Condition]

- When the flash sequencer is ready to accept the P/E suspend command after the program or erase sequence has started

[Clearing Conditions]

- When the flash sequencer has accepted the P/E suspend command or forced stop command (after write access to the FACI command-issuing area is completed)
- When the flash sequencer enters the command-locked state during program or erase operation
- When program or erase operation has been completed

PRGERR Flag (Program Error Flag)

This flag indicates the result of programming the flash memory. If this flag is 1, the flash sequencer is in the command-locked state.

[Setting Condition]

- Refer to Table 48.10, Error Protection Type

[Clearing Condition]

- When the flash sequencer starts processing of a status clear or forced stop command

ERSERR Flag (Erase Error Flag)

This flag indicates the result of erasing the flash memory. If this flag is 1, the flash sequencer is in the command-locked state.

[Setting Condition]

- Refer to Table 48.10, Error Protection Type

[Clearing Condition]

- When the flash sequencer starts processing of a status clear or forced stop command

ILGLERR Flag (Illegal Command Error Flag)

This flag indicates that the flash sequencer has detected an illegal FACI command or flash memory access. When this flag is 1, the flash sequencer is in the command-locked state.

[Setting Condition]

- Refer to Table 48.10, Error Protection Type

[Clearing Condition]

- When the flash sequencer starts processing of a status clear or forced stop command while the DFAE and CFAE flag in the FASTAT register is 0

If the flash sequencer completes processing of a status clear or forced stop command while the CFAE or DFAE flag in the FASTAT register is 1, this flag is set to 1. This flag is temporarily set to 0 during processing of a forced stop command, and is re-set to 1 when the CFAE or DFAE flag is detected as 1 upon completion of command processing.

FRDY Flag (Flash Ready Flag)

This flag indicates the command processing state of the flash sequencer.

[Setting Conditions]

- When the flash sequencer completes command processing
- When the flash sequencer receives a P/E suspend command and suspends program or erase sequence to the flash memory
- When the flash sequencer has received a forced stop command and ended command processing

[Clearing Conditions]

- When the flash sequencer receives the FACI command of the setting of the program and configuration and after the first write access is made to the FACI command-issuing area
- When the flash sequencer receives any FACI command other than of the setting of the program and configuration and after the last write access is made to the FACI command issuing area

48.4.8 Flash P/E Mode Entry Register (FENTRYR)

Address(es): FLASH.FENTRYR 007F E084h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								FENTRYD	—	—	—	—	—	—	FENTRYC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	FENTRYC	Code Flash Memory P/E Mode Entry	0: Code flash memory is in read mode. 1: Code flash memory is in P/E mode.	R/W*1, *2
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	FENTRYD	Data Flash Memory P/E Mode Entry	0: Data flash memory is in read mode. 1: Data flash memory is in P/E mode.	R/W*1, *2
b15 to b8	KEY[7:0]	Key Code	Key code	R/W*3

Note 1. Writing to these bits is possible only when the FSTATR.FRDY flag is 1. Writing to these bits while the FSTATR.FRDY flag = 0 is ignored.
Note 2. Writing to this bit is enabled only when this register is written in 16-bit units with the KEY[7:0] bits set to AAh.
Note 3. Written values are not retained by these bits. These bits are read as 0.

This register is used to specify code flash memory P/E mode and data flash memory P/E mode. To specify code flash memory P/E mode or data flash memory P/E mode so that the flash sequencer can receive FACI commands, set either the FENTRYD or FENTRYC bit to 1 to place the flash sequencer in P/E mode.
Note that writing AA81h in this register causes the FSTATR.ILGLERR flag to be set to 1, and the flash sequencer to enter the command-locked state.
The FENTRYR register is initialized when the FSUINITR.SUINIT bit is set to 1. It is also initialized by a reset.

FENTRYC Bit (Code Flash Memory P/E Mode Entry)

This bit specifies the P/E mode for code flash memory.
[Setting Condition]

- When AA01h is written to the FENTRYR register while the FSTATR.FRDY flag is 1 and the FENTRYR register is 0000h

[Clearing Conditions]

- When the FENTRYR register is written in 8-bit units while the FSTATR.FRDY flag is 1
- When the FENTRYR register is written in 16-bit units with the KEY[7:0] bits set to a value other than AAh while the FSTATR.FRDY flag is 1
- When AA00h is written to the FENTRYR register while the FSTATR.FRDY flag is 1
- When the FENTRYR register is written while the FSTATR.FRDY flag is 1 and the FENTRYR register is not 0000h

FENTRYD Bit (Data Flash Memory P/E Mode Entry)

This bit specifies the P/E mode for data flash memory.
[Setting Condition]

- When AA80h is written to the FENTRYR register while the FSTATR.FRDY flag is 1 and the FENTRYR register is 0000h

[Clearing Conditions]

- When the FENTRYR register is written in 8-bit units while the FSTATR.FRDY flag is 1
- When the FENTRYR register is written in 16-bit units with the KEY[7:0] bits set to a value other than AAh while the FSTATR.FRDY flag is 1

- When AA00h is written to the FENTRYR register while the FSTATR.FRDY flag is 1
- When the FENTRYR register is written while the FSTATR.FRDY flag is 1 and the FENTRYR register is not 0000h

KEY[7:0] Bits (Key Code)

These bits control permission and prohibition of writing to the FENTRYD and FENTRYC bits.

48.4.9 Flash Sequencer Set-Up Initialization Register (FSUINITR)

Address(es): FLASH.FSUINITR 007F E08Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								—	—	—	—	—	—	—	SUINIT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SUINIT	Set-Up Initialization	0: The FEADDR, FCPSR, FSADDR, FENTRYR, and FBCCNT flash sequencer set-up registers keep their current values. 1: The FEADDR, FCPSR, FSADDR, FENTRYR, and FBCCNT flash sequencer set-up registers are initialized.	R/W*1, *2
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	Key Code	Key code	R/W*3

Note 1. Writing to these bits is possible only when the FSTATR.FRDY flag is 1. Writing to these bits while the FSTATR.FRDY flag = 0 is ignored.
Note 2. Writing to these bits is possible only when this register is written in 16-bit units with the KEY[7:0] bits set to 2Dh.
Note 3. Written values are not retained by these bits. These bits are read as 0.

This register is used for initialization of the flash sequencer set-up.

SUINIT Bit (Set-Up Initialization)

This bit initializes the following flash sequencer set-up registers.

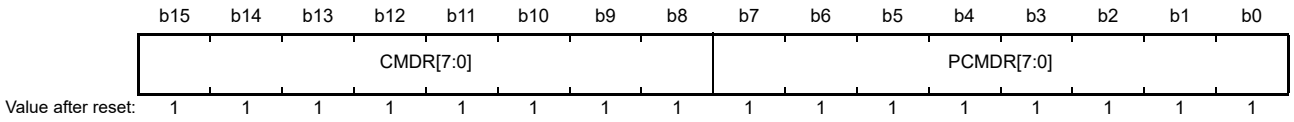
- FEADDR
- FCPSR
- FSADDR
- FENTRYR
- FBCCNT

KEY[7:0] Bits (Key Code)

These bits control permission and prohibition of writing to the SUINIT bit.

48.4.10 FACI Command Register (FCMDR)

Address(es): FLASH.FCMDR 007F E0A0h



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	PCMDR[7:0]	Precommand Flag	The command immediately before the latest command is stored.	R
b15 to b8	CMDR[7:0]	Command Flag	The latest command is stored.	R

This register records the two most recent commands accepted by the FACI.

PCMDR[7:0] Flags (Precommand Flag)

These flags indicate the command received immediately before the last command received by the FACI.

CMDR[7:0] Flags (Command Flag)

These flags indicate the latest command received by the FACI.

Table 48.4 States of FCMDR after Receiving Commands

Command	CMDR	PCMDR
Program	E8h	Previous command
Block erase	D0h	20h
P/E suspend	B0h	Previous command
P/E resume	D0h	Previous command
Status clear	50h	Previous command
Forced stop	B3h	Previous command
Blank check	D0h	71h
Configuration set	40h	Previous command

48.4.11 Flash P/E Status Register (FPESTAT)

Address(es): FLASH.FPESTAT 007F E0C0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	PEERRST[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	PEERRST[7:0]	P/E Error Status Flag	00h: No error 02h: Program error 12h: Erase error	R
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register indicates the result of programming or erasing the flash memory.

PEERRST[7:0] Flags (P/E Error Status Flag)

These flags indicate the source of an error that occurred during a program or erase operation to the code flash memory or data flash memory. The value of these flags is only valid if the ERSERR or PRGERR flag in the FSTATR register is 1 when the FSTATR.FRDY flag becomes 1. When the ERSERR and PRGERR flags are 0, these flags retain their value to indicate the source of the last error to have occurred.

48.4.12 Data Flash Blank Check Control Register (FBCCNT)

Address(es): FLASH.FBCCNT 007F E0D0h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	BCDIR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BCDIR	Blank Check Direction	0: Blank checking is executed from lower addresses to higher addresses (incremental mode). 1: Blank checking is executed from higher addresses to lower addresses (decremental mode).	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register specifies the addressing mode in processing of a blank check command.
The register is initialized when the FSUINITR.SUINIT bit is set to 1. It is also initialized by a reset.

BCDIR Bit (Blank Check Direction)

This bit specifies the addressing mode for blank checking.

48.4.13 Data Flash Blank Check Status Register (FBCSTAT)

Address(es): FLASH.FBCSTAT 007F E0D4h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	BCST
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BCST	Blank Check Status Flag	0: The target area is in the non-programmed state (i.e. is blank; the area has been erased but has not yet been re-programmed). 1: The target area has been programmed with 0s or 1s.	R
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register stores the results of checking in response to a blank check command.

BCST Flag (Blank Check Status Flag)

This flag indicates the results of checking in response to a blank check command.
At the point where the FSTATR.FRDY flag is set to 1, the valid data is stored in the BCST flag.

48.4.14 Data Flash Programming Start Address Register (FPSADDR)

Address(es): FLASH.FPSADDR 007F E0D8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	PSADR[18:16]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PSADR[15:0]															
Value after reset:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b18 to b0	PSADR[18:0]	Programmed Area Start Address	The address of the first programmed area	R
b31 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register indicates the address of the first programmed area to be found in processing of a blank check command.

PSADR[18:0] Bits (Programmed Area Start Address)

These bits indicate the address of the first programmed area to be found in processing of a blank check command. The address is an offset from the address where the data flash memory starts. The setting of these bits is valid only while the FBCSTAT.BCST flag is 1 and when the FSTATR.FRDY flag becomes 1. When the FBCSTAT.BCST flag is 0, the PSADR[18:0] bits holds the address detected by the previous check.

48.4.15 Flash Access Window Monitor Register (FAWMON)

Address(es): FLASH.FAWMON 007F E0DCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	BTFLG	—	—	FAWE[12:0]												
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FSPR	—	—	FAWS[12:0]												
Value after reset:	The value set by the user*1															

Bit	Symbol	Bit Name	Description	R/W
b12 to b0	FAWS[12:0]	Flash Access Window Start Address	Flash access window start address	R
b14, b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	FSPR	Access Window Protection Flag	0: With protection (P/E disabled) 1: Without protection (P/E enabled)	R
b28 to b16	FAWE[12:0]	Flash Access Window End Address	Flash access window end address	R
b30, b29	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31	BTFLG	Start-Up Area Select Flag*2	0: Startup area 0 is in the range from FFFF 8000h to FFFF BFFFh, startup area 1 is in the range from FFFF C000h to FFFF FFFFh. 1: Startup area 1 is in the range from FFFF 8000h to FFFF BFFFh, startup area 0 is in the range from FFFF C000h to FFFF FFFFh.	R

Note 1. The value of the blank product is FFFF FFFFh. This register is set to a specified value after programming of the flash memory with the user program.
Note 2. When the FSUACR.SAS[1:0] bits are changed to 1xb, the startup area is dependent on the setting of the FSUACR.SAS[1:0] bits regardless of the setting of the FAW.BTFLG bit.

This register indicates the values of the write protection flag and start-up area select flag for setting the flash access window start/end address, and the access window. When a reset or configuration set command is executed, the FACI transfers data from the option-setting memory to this register and the setting of the option-setting memory is enabled.

FAWS[12:0] Bits (Flash Access Window Start Address)

These bits are used to verify the access window start address setting value.

FSPR Flag (Access Window Protection Flag)

This flag indicates whether or not protection for a configuration set command for the access window setting or for writing to the FSUACR register is available.

FAWE[12:0] Bits (Flash Access Window End Address)

These bits are used to verify the access window end address setting value. The value of these bits indicates the first address of the next P/E-enabled block as configured in the access window.

BTFLG Flag (Start-Up Area Select Flag)

This flag indicates whether the start-up area is switched by using start-up program protection.

48.4.16 Flash Sequencer Processing Switching Register (FCPSR)

Address(es): FLASH.FCPSR 007F E0E0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ESUSPMD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ESUSPMD	Erase Suspend Mode	0: Suspend priority mode 1: Erase priority mode	R/W
b15 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is for selecting the erase suspend mode.
The register is initialized when the FSUINITR.SUINIT bit is set to 1. It is also initialized by a reset.

ESUSPMD Bit (Erase Suspend Mode)

This bit is for selecting the erase suspend mode when a P/E suspend command is issued while the flash sequencer is executing erase sequence (refer to section 48.6.7.5, P/E Suspend Command). This bit should be set before issuing a block erase command.

48.4.17 Flash Sequencer Processing Clock Frequency Notification Register (FPCKAR)

Address(es): FLASH.FPCKAR 007F E0E4h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								PCKA[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	PCKA[7:0]	Flash Sequencer Processing Clock Frequency Notification	These bits are used to set the frequency of the FlashIF clock (FCLK) and notify the flash sequencer of the frequency used	R/W*1, *2
b15 to b8	KEY[7:0]	Key Code	Key code	R/W*3

Note 1. Writing to these bits is possible only when the FSTATR.FRDY flag is 1. Writing to these bits while the FSTATR.FRDY flag = 0 is ignored.
Note 2. Writing to these bits is enabled only when this register is written in 16-bit units with the KEY[7:0] bits set to 1Eh.
Note 3. Written values are not retained by these bits. These bits are read as 0.

This register specifies the frequency of the FlashIF clock (FCLK) generated in the clock generator and notifies the flash sequencer of the frequency used. The flash sequencer determines the FACI command processing time based on the frequency notified by the FPCKAR register. The initial value is set to the maximum operating frequency of the FCLK.

PCKA[7:0] Bits (Flash Sequencer Processing Clock Frequency Notification)

These bits are used to specify the frequency of the FCLK generated in the clock generator and to notify the flash sequencer of the frequency used. Set the desired frequency in these bits before issuing an FACI command. Specifically, convert the frequency represented in MHz into a binary number and set it in these bits.

Example: When frequency is 35.9 MHz (PCKA[7:0] = 24h)
Round up the first decimal place of 35.9 MHz to a whole number (= 36) and convert it into a binary number.

If the value set in these bits is smaller than the frequency of the FCLK, the rewriting characteristics of the flash memory cannot be guaranteed. Conversely, if the value set in these bits is greater than the frequency of the FCLK, the rewriting characteristics of the flash memory can be guaranteed although the FACI command processing time such as time for rewriting will increase (the FACI command processing time becomes the shortest when the frequency of the FCLK is the same as the value set in the bits).

KEY[7:0] Bits (Key Code)

These bits control permission and prohibition of writing to the PCKA[7:0] bits.

48.4.18 Start-Up Area Control Register (FSUACR)

Address(es): FLASH.FSUACR 007F E0E8h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								—	—	—	—	—	—	SAS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	SAS[1:0]	Start-Up Area Select	b1 b0 0 x: A start-up area is selected based on the setting of the FAW.BTFLG bit. 1 0: The area from FFFF C000h to FFFF FFFFh is selected as the address range for startup area 0, and the area from FFFF 8000h to FFFF BFFFh is selected as the address range for startup area 1, regardless of the setting of the FAW.BTFLG bit. 1 1: The area from FFFF C000h to FFFF FFFFh is selected as the address range for startup area 1, and the area from FFFF 8000h to FFFF BFFFh is selected as the address range for startup area 0, regardless of the setting of the FAW.BTFLG bit.	R/W*1, *2
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	Key Code	Key codes	R/W*3

x: Don't care

Note 1. Writable only when the FAW.FSPR bit is 1. When the FAW.FSPR bit is 0, writing to these bits are ignored.
Note 2. Writing to these bits is possible only when this register is written in 16-bit units with the KEY[7:0] bits set to 66h.
Note 3. Written values are not retained by these bits. These bits are read as 0.

This register is used to switch startup areas 0 and 1 by startup program protection.
Do not use this register in dual mode (the MDE.BANKMD[2:0] bits are 000b). In dual mode, starting up proceeds from startup area 0.

SAS[1:0]Bits (Start-Up Area Select)

These bits are used to switch startup areas 0 and 1.

KEY[7:0] Bits (Key Code)

These bits enables or disable overwriting to the SAS[1:0] bits.

48.4.19 Unique ID Register n (UIDRn) (n = 0 to 2)

Address(es): FLASH.UIDR0 007F B174h, FLASH.UIDR1 007F B1E4h, FLASH.UIDR2 007F B1E8h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16

Value after reset: Unique value for each chip

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0

Value after reset: Unique value for each chip

The UIDRn is a read-only register that stores a 12-byte ID code (unique ID) for identifying the individual MCU. The UIDRn register should be read in 32-bit units.

48.5 Overview of Functions

48.5.1 Program and Erase Methods

By using a flash-memory programmer to program the flash memory of this MCU, the device can be rewritten regardless of whether this is before or after it is mounted on the target system.
Furthermore, security functions to prohibit rewriting or reading of the user program written to the flash memory are incorporated, and this can prevent falsification and illicit reading of the programs by third parties. Blocks 8 and 9 (blocks 8, 9, 30, and 31 in dual mode) of the code flash memory can also be protected against reading by using the TM function.
Programming by the user program (self-programming) is available to suit applications where the application on the target system may require updating after manufacturing or shipment. Protection features for the safe rewriting of the flash memory are also incorporated. Furthermore, interrupt processing during self-programming is supported, so programming can proceed at the same time as processing for external communications, etc., and this is the case in various situations.
Table 48.5 lists the overview of the methods of programming and the corresponding operating modes.

Table 48.5 Methods of Programming

Method of Programming	Functional Overview	Operating Mode
Programming by a flash-memory programmer	A serial programmer is capable of on-board programming of the flash memory after the device is mounted on the target system. The TM function can be in enabled or disabled at this time.	Boot mode
Self-programming	The execution of the user program that is written to code flash memory in advance by serial programming executing is also capable of programming the flash memory. The TM function can be in the enabled state at this time. The background operation capability makes it possible to fetch instructions or otherwise read data from code flash memory while the data flash memory is being programmed. Therefore, the data flash memory can be rewritten by executing a program for writing contained by the code flash memory. Furthermore, background operation can also be used for reading and writing to code flash memory alone, but only when the address range of code flash memory that is the target for programming and the address range of code flash memory that is the target for reading satisfy particular conditions (refer to Table 48.24). When this is the case, at the time of self-programming, a program for programming that is in the code flash memory can be executed to rewrite the code flash memory. In cases where background operation is not possible, instructions in the code flash memory cannot be fetched and data cannot be accessed while code flash memory is being re-written by self-programming. In such cases, a program for programming must be transferred to the internal RAM in advance and executed.	Single-chip mode

The programmable/erasable area of the flash memory and the boot program after a reset are different according to each mode. The differences between modes are listed in Table 48.6.

Table 48.6 Differences between Modes

Item	Single-chip Mode	Boot Mode (SCI or FINE Interface)
Programmable and erasable area	• Code flash memory • Data flash memory • Option-setting memory (programming only)	• Code flash memory • Data flash memory • Option-setting memory
Boot program at a reset	Program on the code flash memory	Boot program

Table 48.7 lists the functions of the flash memory. Serial programmer commands realize each function of serial programming, while reading of the flash memory by an FACI command or the user program realizes each function of self-programming.
For security function settings, refer to section 7.2.1, Serial Programmer Command Control Register (SPCC), in section 7, Option-Setting Memory (OFSM).

Table 48.7 List of Basic Functions

Function	Functional Overview	Support Status	
		Serial Programming	Self-programming
Blank check	This is used to check a specified block to ensure that writing to it has not already proceeded. Results of reading from data flash memory to which nothing has been written after erasure are not guaranteed, so use blank checking to confirm that programming to memory has not proceeded after erasure.	Supported	Supported
Block erase	Erases the memory contents in the specified block.	Supported	Supported
Program	Write to the specified address.	Supported	Supported
Verify/Checksum	Compares the data read from flash memory and the data transferred from a flash memory programmer.	Supported	Not supported (read by user program is possible)
Read	Read data programmed to flash memory.	Supported	Supported
Setting of control or ID codes	Sets the OSIS register.	Supported	Supported
Setting the security functions	Sets the SPCC register and enables the following functions. • Prohibition of on-chip debugger connection • Serial programmer ID code protection • Prohibition of serial programmer connection	Supported	Supported with conditions (Only switching the configuration from disabled to enabled is possible)
Area protection and start-up program protection functions	Configures the area protection and start-up program protection functions.	Supported	Supported
Configuration clear	Erases the configuration setting area. Disables the TM function.	Supported	Not supported
Dual bank function	Switches different modes (linear or dual).	Supported	Supported
Setting the TM function	Sets the TM function.	Supported	Supported with conditions (Only switching the configuration from disabled to enabled is possible)

48.5.2 Security Functions

The flash memory supports various security functions.
The security function includes on-chip debugger ID code protection, prohibition of on-chip debugger connection, serial programmer ID code protection, and prohibition of serial programmer connection.
In serial programming, serial programmer ID code protection and prohibition of serial programmer connection can be used. On-chip debugger ID code protection and prohibition of on-chip debugger connection can be used when using the on-chip debugger.
Table 48.8 lists the security functions supported by the flash memory, and Table 48.9 lists the operations with security settings.

Table 48.8 Lists of Security Functions

Function	Description
On-chip debugger ID code protection	Connection to the on-chip debugger can be controlled by judging the ID code.
Prohibition of on-chip debugger connection	Connection to the on-chip debugger is prohibited regardless of the ID code setting.
Serial programmer ID code protection	Connection of a serial programmer can be controlled by judging the control code or ID code.
Prohibition of serial programmer connection	Connection of a serial programmer for serial programming is prohibited. Since execution of the configuration clear command is also prohibited when the connection of a serial programmer is prohibited, changing a security setting from “disabled” to “enabled” is not possible.

Table 48.9 Operations with Security Settings

	Programming/Erasing/Reading with Respective Security Settings		Notes for Security Setting	
Function	Serial Programming	Self-Programming	Serial Programming	Self-Programming
Prohibition of on-chip debugger connection	Block erase commands: Supported Program commands: Supported Read commands: Supported	Block erase commands: Supported Program commands: Supported Read commands: Supported	The configuration clear command can initialize the setting for prohibition.	Since the configuration clear command is not supported, initialization of the setting for prohibition is not possible.
Serial programmer ID code protection	(When the ID codes do not match) Block erase commands: Not supported Program commands: Not supported Read commands: Not supported (When the ID codes match) Block erase commands: Supported Program commands: Supported Read commands: Supported	(Judgment of the ID codes is not performed.) Block erase commands: Supported Program commands: Supported Read commands: Supported	The configuration clear command can initialize the setting for prohibition.	(Judgment of the ID codes is not performed.)
Prohibition of serial programmer connection	Block erase commands: Not supported Program commands: Not supported Read commands: Not supported	Block erase commands: Supported Program commands: Supported Read commands: Supported	Since execution of the configuration clear command is prohibited, initialization of the setting for prohibition is not possible.	Since the configuration clear command is not supported, initialization of the setting for prohibition is not possible.

48.5.2.1 On-Chip Debugger ID Code Protection

This function is used to prohibit connection to the on-chip debugger (OCD).
The code sent from the emulator is compared with the ID code set in the OCD/serial programmer ID setting register (OSIS) to determine whether they match. If they match, connection to the OCD is allowed. If they do not match, connection to the OCD is prohibited.
For details on the OSIS register, refer to section 7.2.2, OCD/Serial Programmer ID Setting Register (OSIS).

48.5.2.2 Serial Programmer ID Code Protection

This function is used to prohibit connection with the serial programmer.
The code sent from the serial programmer is compared with the ID code set in the OCD/serial programmer ID setting register (OSIS) to determine whether they match. If they match, connection with the serial programmer is allowed. If they do not match, the serial programmer cannot be connected. If the control code is 45h and the codes do not match three times in a row, all data in the flash memory are erased. However, if the FAW.FSPR bit is 0, the flash memory is not erased.
For details on the OSIS register, refer to section 7.2.2, OCD/Serial Programmer ID Setting Register (OSIS). For details on the FAW register, refer to section 7.2.9, Flash Access Window Setting Register (FAW).

48.5.3 Protection Function

48.5.3.1 Software Protection

Software protection disables programming and erasing of the code flash memory through the settings of control registers. If an attempt is made to issue an FACI command against software protection, the flash sequencer enters the command-locked state.

(1) Protection through FWEPROR Register

Programming cannot proceed in any mode unless the FWEPROR.FLWE[1:0] bits are set to 01b.

(2) Protection through FENTRYR Register

When the FENTRYR register is set to 0000h, the flash sequencer enters read mode. In read mode, FACI commands cannot be accepted. If an attempt is made to issue an FACI command in read mode, the flash sequencer enters the command-locked state.

48.5.3.2 Error Protection

Error protection detects erroneous issuance of FACI commands, unauthorized access, and flash sequencer malfunction. FACI command acceptance is disabled (command-locked state) in response to the detection of these errors. The flash memory cannot be programmed or erased while the flash sequencer is in the command-locked state. For release from the command-locked state, issue a status clear or forced stop command while the CFAE and DFAE flags in the FASTAT register are 0. The status clear command can only be used while the FSTATR.FRDY flag is 1. The forced stop command can be used regardless of the value of the FRDY flag.
Generation of a flash access error (FIFERR) interrupt detects malfunction. An FIFERR interrupt is generated under any of the following conditions.

- When violation in access to the data flash memory occurred (the FASTAT.DFAE flag is 1) while the FAEINT.DFAEIE bit is 1
- When the flash sequencer enters the command lock state (when the FASTAT.CMDLK flag is 1) while the FAEINT.CMDLKIE bit is 1
- When violation in access to the code flash memory occurred (when the FASTAT.CFAE flag is 1) while the FAEINT.CFAEIE bit is 1

When the flash sequencer enters the command-locked state in response to a command other than the P/E suspend command during program or erase operation, the flash sequencer continues the program or erase operation. In this state, the P/E suspend command cannot be used to suspend the program or erase operation. If a command is issued in the command-locked state, the FSTATR.ILGLERR flag becomes 1 and the other flags retain the values set due to previous error detection.
Table 48.10 shows error protection types and status bit values after error detection.

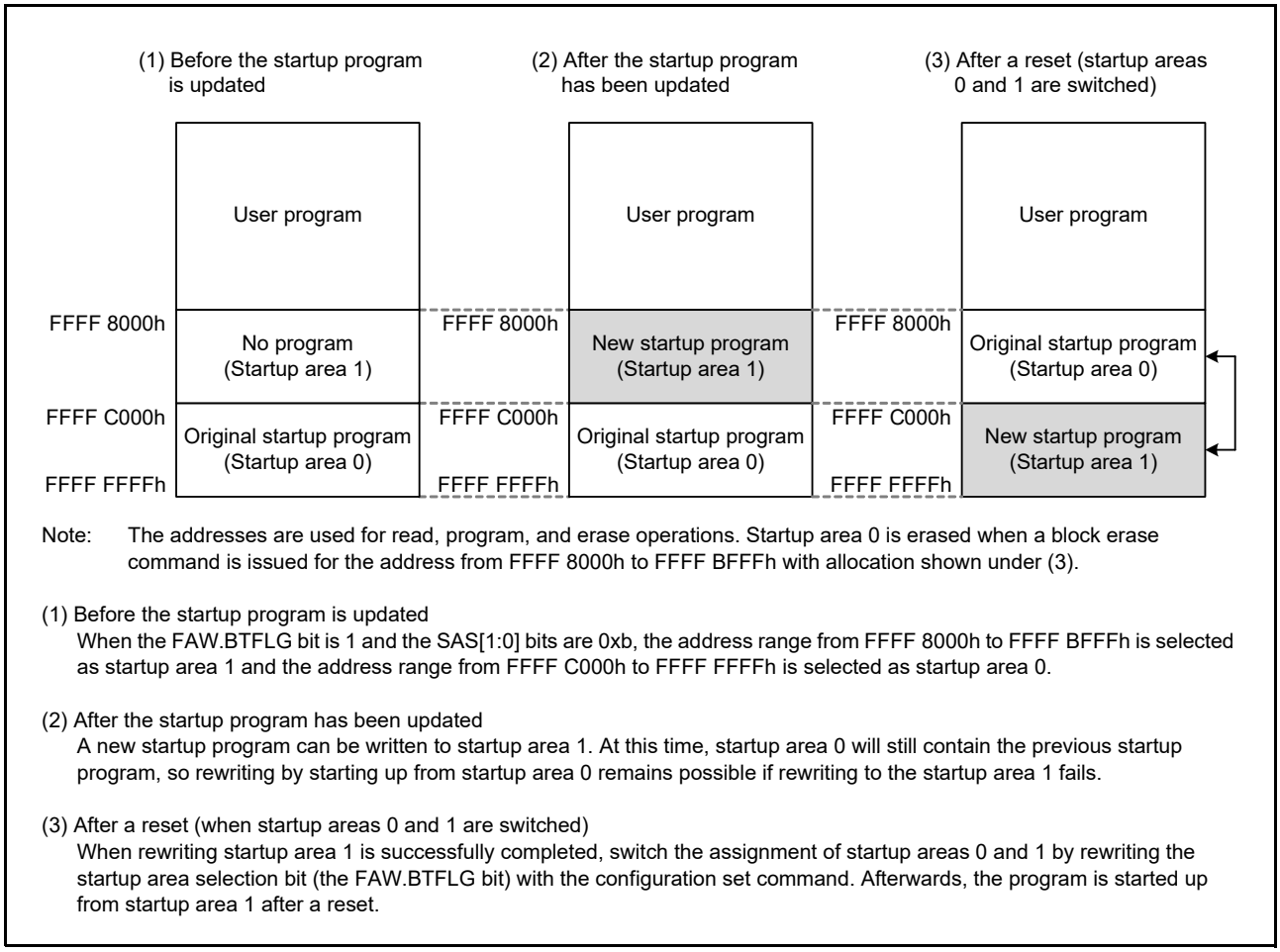
Table 48.10 Error Protection Type

Error Type	Description	ILGLERR	ERSERR	PRGERR	FLWEERR	CFAE	DFAE
FENTRYR setting error	AA81h was written to the FENTRYR register.	1	0	0	0	0	0
	The FENTRYR register value at suspension disagrees with that at resumption.	1	0	0	0	0	0
Illegal command error	An undefined code was written on the first access of an FACI command.	1	0	0	0	0	0
	A value other than D0h was written on the last access of an FACI command with multiple access cycles.	1	0	0	0	0	0
	The value “N” (refer to Table 48.16) specified in the second access of the FACI command in the program or configuration set command is illegal.	1	0	0	0	0	0
	A blank check command was issued with one of the following settings. • The FBCCNT.BCDIR bit is 0, and the value in the FSADDR register > that in the FEADDR register • The FBCCNT.BCDIR bit is 1, and the value in the FEADDR register > that in the FSADDR register • The value of bits b18 to b0 of the FEADDR register is set in the range from 0 4000h to 7 FFFFh	1	0	0	0	0	0
	An FACI command not acceptable in each mode was issued (refer to Table 48.13).	1	0	0	0	0	0
	A program or block erase command was issued to an area protected by area protection.	1	0	0	0	0	0
	An FACI command was issued when command acceptance conditions are not satisfied (refer to Table 48.14)	1	0/1	0/1	0/1	0/1	0/1
Erase error	An error occurs during erase operation.	0	1	0	0	0	0
Program error	An error occurs during program operation.	0	0	1	0	0	0
Code flash memory access violation	A program command or block erase command was issued under the following settings in code flash memory P/E mode. • The value of bits b23 to b0 of the FSADDR register is set in the range from 00 0000h to F7 FFFFh*1 or from 00 0000h to FB FFFFh*2	1	0	0	0	1	0
Data flash memory access violation	A program command or block erase command was issued under the following settings in data flash memory P/E mode. • The value of bits b18 to b0 of the FSADDR register is set in the range from 0 4000h to 7 FFFFh	1	0	0	0	0	1
	A configuration set command was issued under the following settings in data flash memory P/E mode • The value of bits b18 to b0 of the FSADDR register is set in the range from 0 0000h to 0 003Fh or from 0 0100h to 7 FFFFh	1	0	0	0	0	1
Security error	A configuration set command was issued to the FAW register while the FAW.FSPR bit was 0.	1	0	0	0	0	0
Others	The FACI command-issuing area was accessed in read mode	1	0	0	0	0	0
	The FACI command-issuing area was read in code flash memory P/E mode or data flash memory P/E mode	1	0	0	0	0	0
Flash P/E protection error	A flash memory program/erase protection error for the setting of the FWEPROR register was detected during command processing by the flash sequencer.	0	0/1	0/1	1	0	0

Note 1. For products with 64 Kbytes of RAM
Note 2. For products with 48 Kbytes of RAM

48.5.4 Start-Up Program Protection

Protection of the startup program is for protection of the program to be started after a reset (the startup program). This function provides a way to safely update the startup program when rewriting is suspended during a reset.
The startup area is 8 Kbytes in size and is assigned to the code flash memory. This function uses the values of the FAW.BTFLG bit and the FSUACR.SAS[1:0] bits to change the area where the startup program is stored in block units (see Figure 48.5 to Figure 48.8).
In protection of the startup program, the state of the selection of the startup area can be fixed by the access window protection bit (FAW.FSPR). However, the FAW.FSPR bit never be restored to 1 once the flag is set to 0. Exercise extra caution when handling the FAW.FSPR bit.
In addition, this protection cannot be used when dual mode is selected by the bank mode switching function (when the MDE.BANKMD[2:0] bits are 000b).



Note: The addresses are used for read, program, and erase operations. Startup area 0 is erased when a block erase command is issued for the address from FFFF 8000h to FFFF BFFFh with allocation shown under (3).

(1) Before the startup program is updated
When the FAW.BTFLG bit is 1 and the SAS[1:0] bits are 0xb, the address range from FFFF 8000h to FFFF BFFFh is selected as startup area 1 and the address range from FFFF C000h to FFFF FFFFh is selected as startup area 0.

(2) After the startup program has been updated
A new startup program can be written to startup area 1. At this time, startup area 0 will still contain the previous startup program, so rewriting by starting up from startup area 0 remains possible if rewriting to the startup area 1 fails.

(3) After a reset (when startup areas 0 and 1 are switched)
When rewriting startup area 1 is successfully completed, switch the assignment of startup areas 0 and 1 by rewriting the startup area selection bit (the FAW.BTFLG bit) with the configuration set command. Afterwards, the program is started up from startup area 1 after a reset.

Figure 48.5 Concept of Protection of the Startup Program

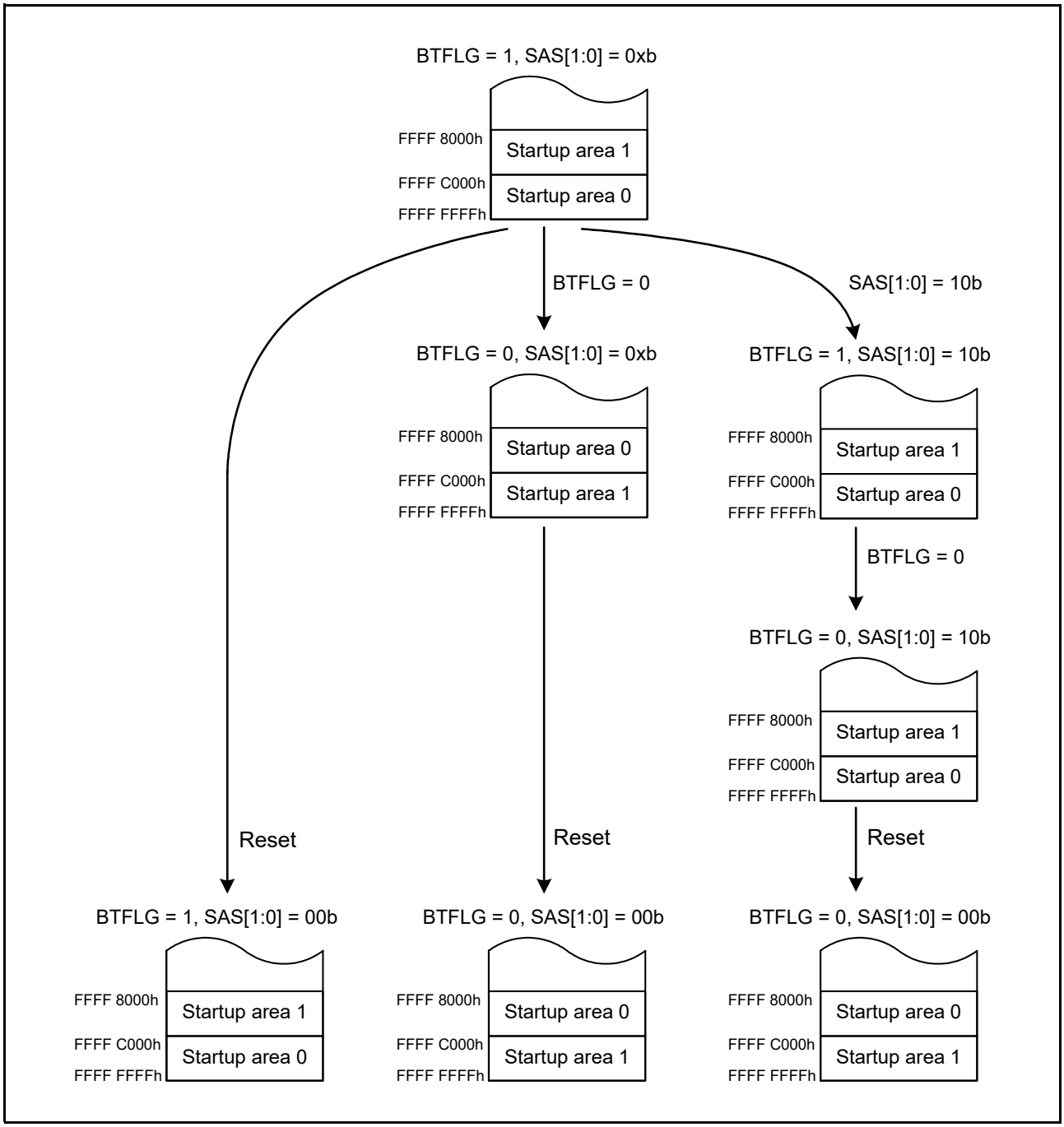


Figure 48.6 Example 1 of Transitions for Startup Program Protection Settings

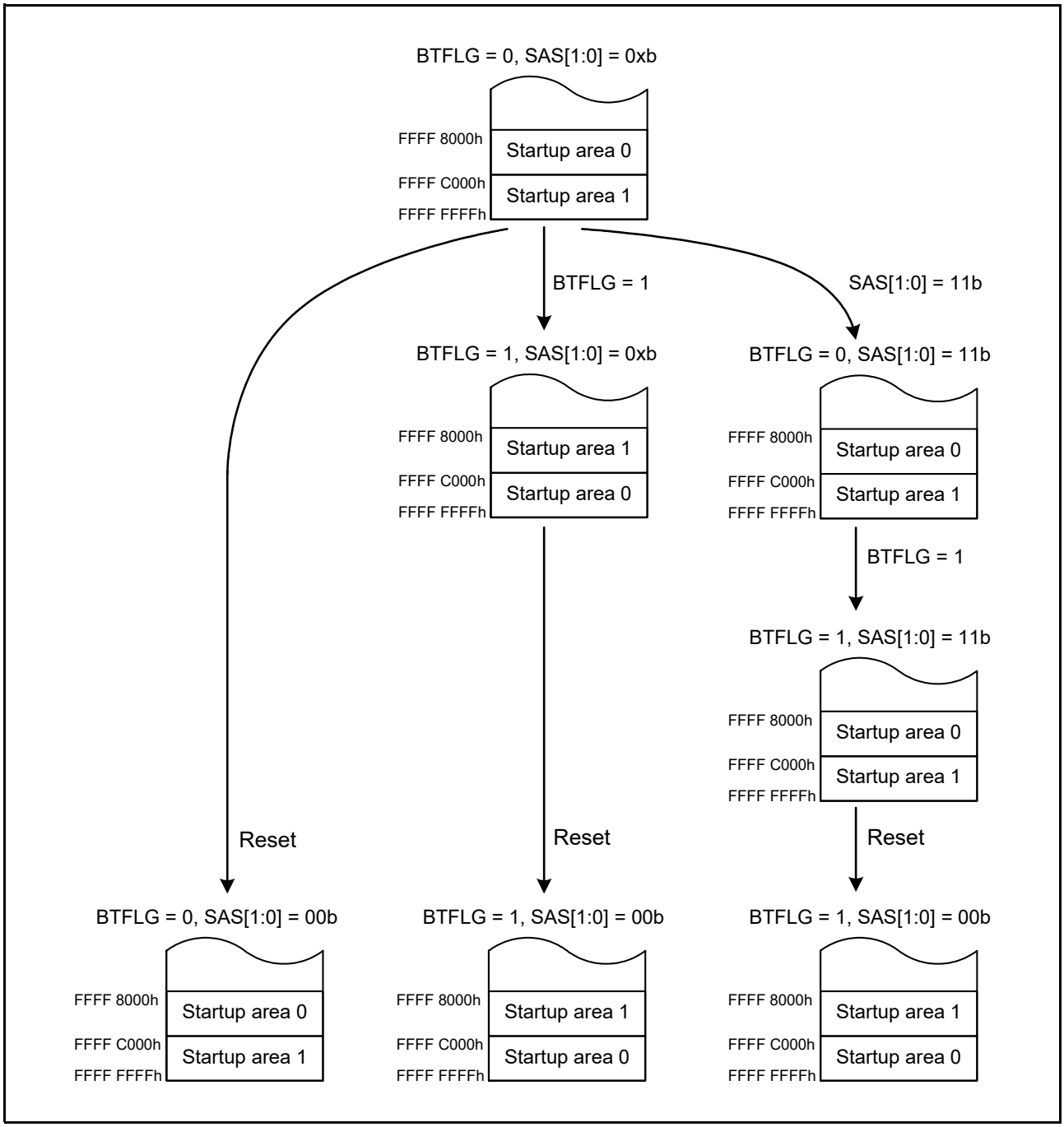


Figure 48.7 Example 2 of Transitions for Startup Program Protection Settings

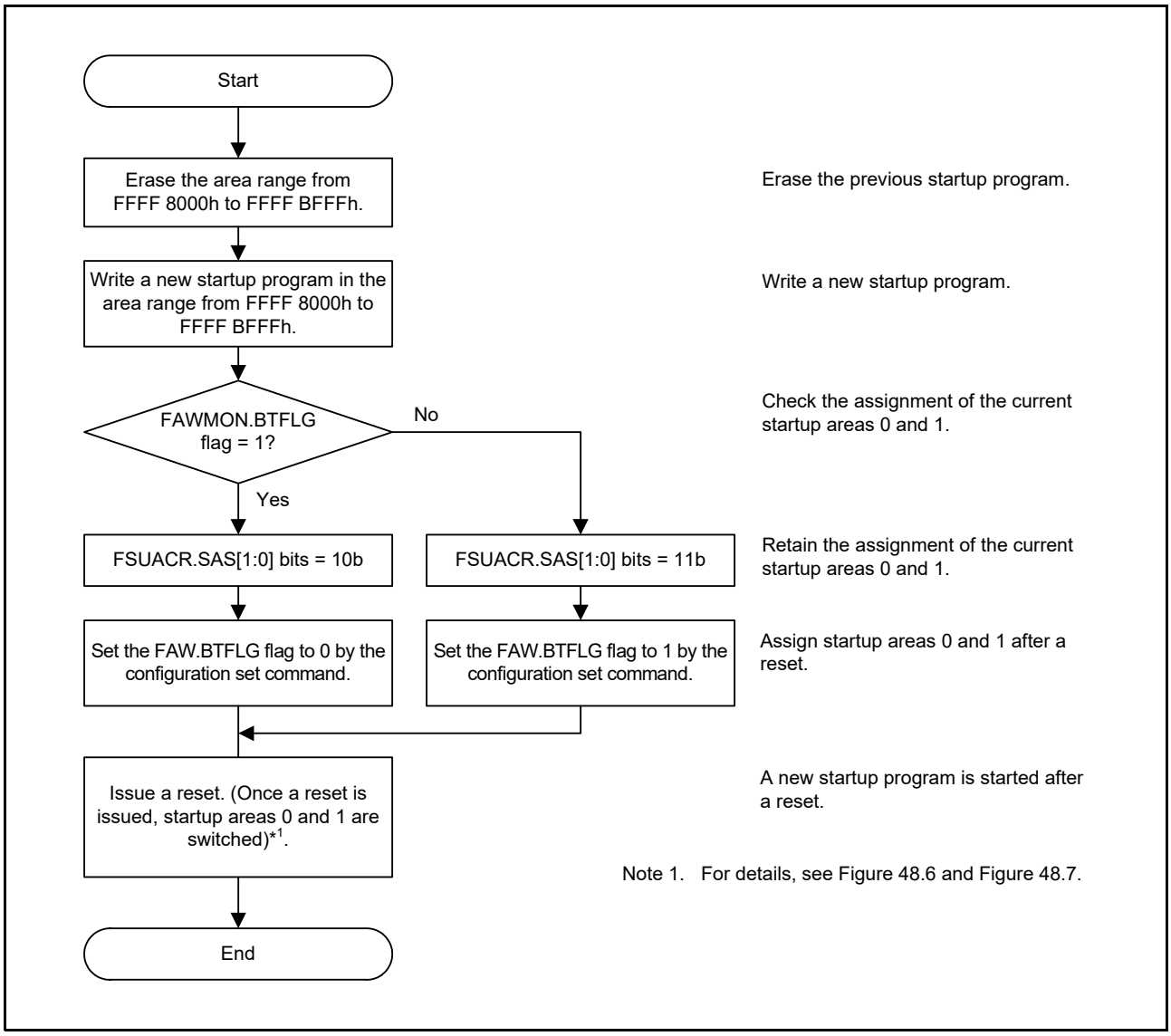


Figure 48.8 Example of Flowchart for Selecting Startup Area

48.5.5 Protection by Area Protection

Issuing an FACI command that programs or erases the area set outside the access window leads to the command-locked state. The access window is valid only in the code flash memory. The access window is valid in self-programming mode or serial programming mode.

The access window can be set in block units.

Figure 48.9 shows the Area Protection Overview.

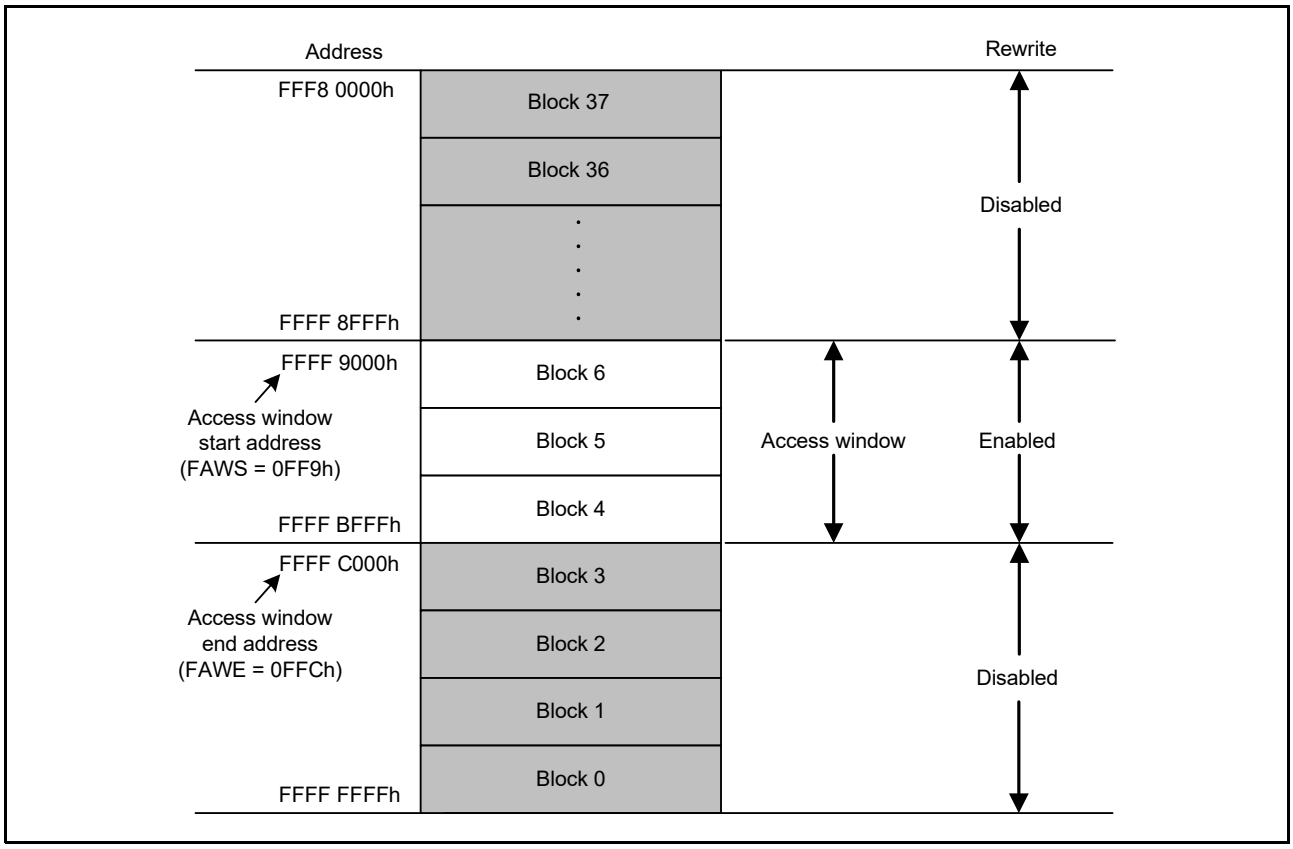


Figure 48.9 Area Protection Overview (When Blocks 4 to 6 are Set as the Access Window in Products with 512 Kbytes of Code Flash Memory)

48.5.6 Dual Bank Function

This protection uses the functions of bank mode switching and startup bank selection to update a program while a user program is running and to provide a safe method of updating in cases where program operation is suspended during a reset.

48.5.6.1 Switching Bank Modes

The bank mode switching function selects either linear mode in which the code flash memory is used as one area, or dual mode in which the code flash memory is divided into two bank areas. Figure 48.10 shows an example of flow of switching bank modes. A reset after setting the MDE.BANKMD[2:0] bits in the option setting memory determines the mode of the bank mode switching function. Selecting dual mode enables the startup bank selection function.
When dual mode is selected by bank mode switching function (the MDE.BANKMD[2:0] bits are 000b), start-up program protection function cannot be used.

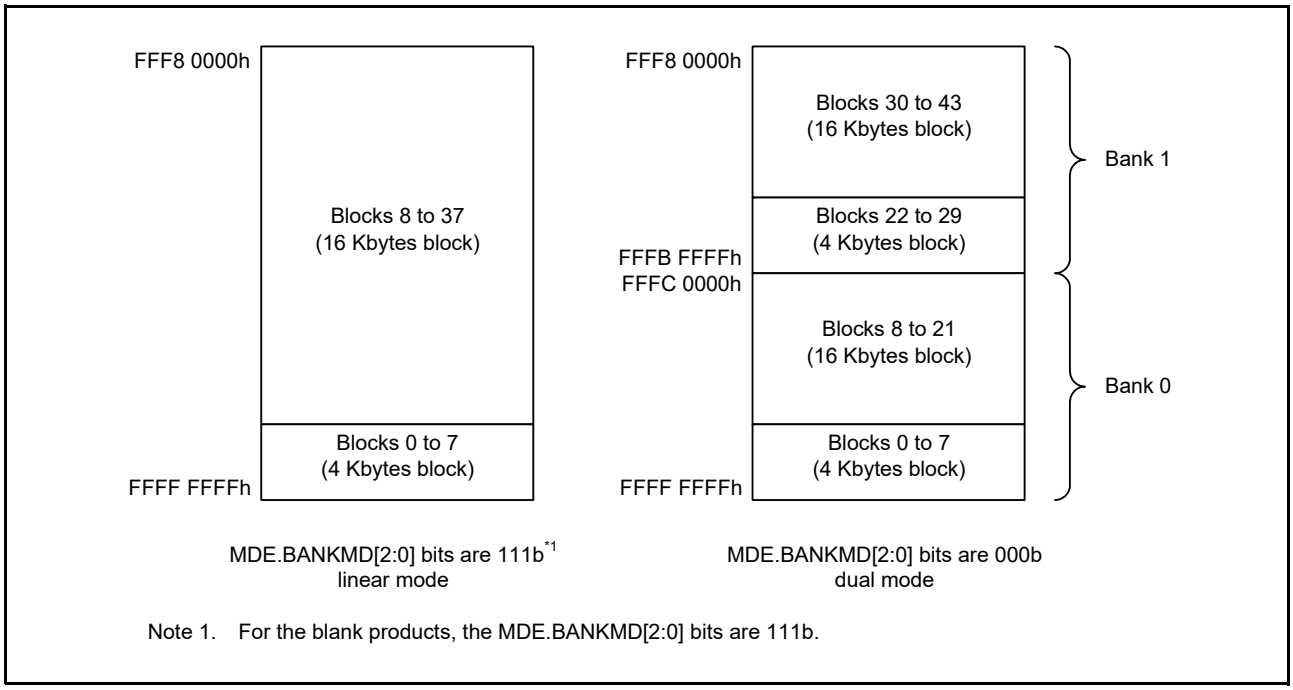


Figure 48.10 Example of Bank Mode Switching (For Products with 512 Kbytes of Code Flash Memory)

48.5.6.2 Selecting the Startup Bank

Startup bank selection provides a way to safely update the program by selecting a bank area to be started in dual mode (when the MDE.BANKMD[2:0] bits are 000b) when program operation is suspended during a reset. Figure 48.11 is a schematic view of startup bank selection and Figure 48.12 shows an example of the flow of startup bank selection. A reset after setting the value of the BANKSEL.BANKSWP[2:0] bits in the option-setting memory changes the addresses of banks 0 and 1 and booting up a program proceeds from the updated area. When the address is switched by using startup bank selection, the P/E target for the FACI commands is also switched. This function is invalid in linear mode.

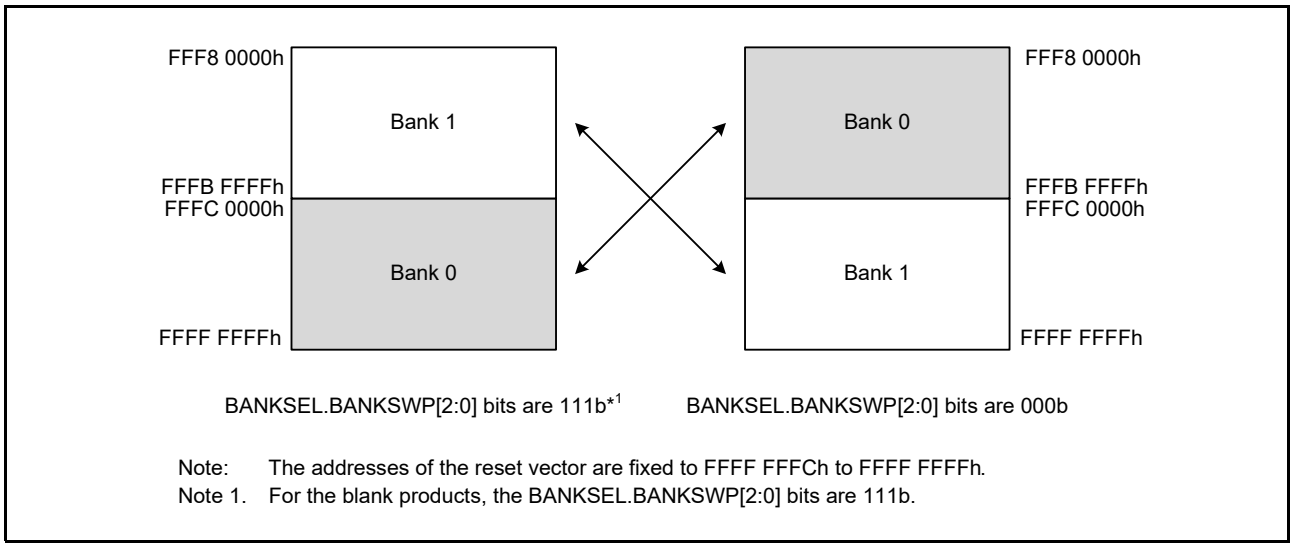


Figure 48.11 Example of Startup Bank Selection (For Products with 512 Kbytes of Code Flash Memory)

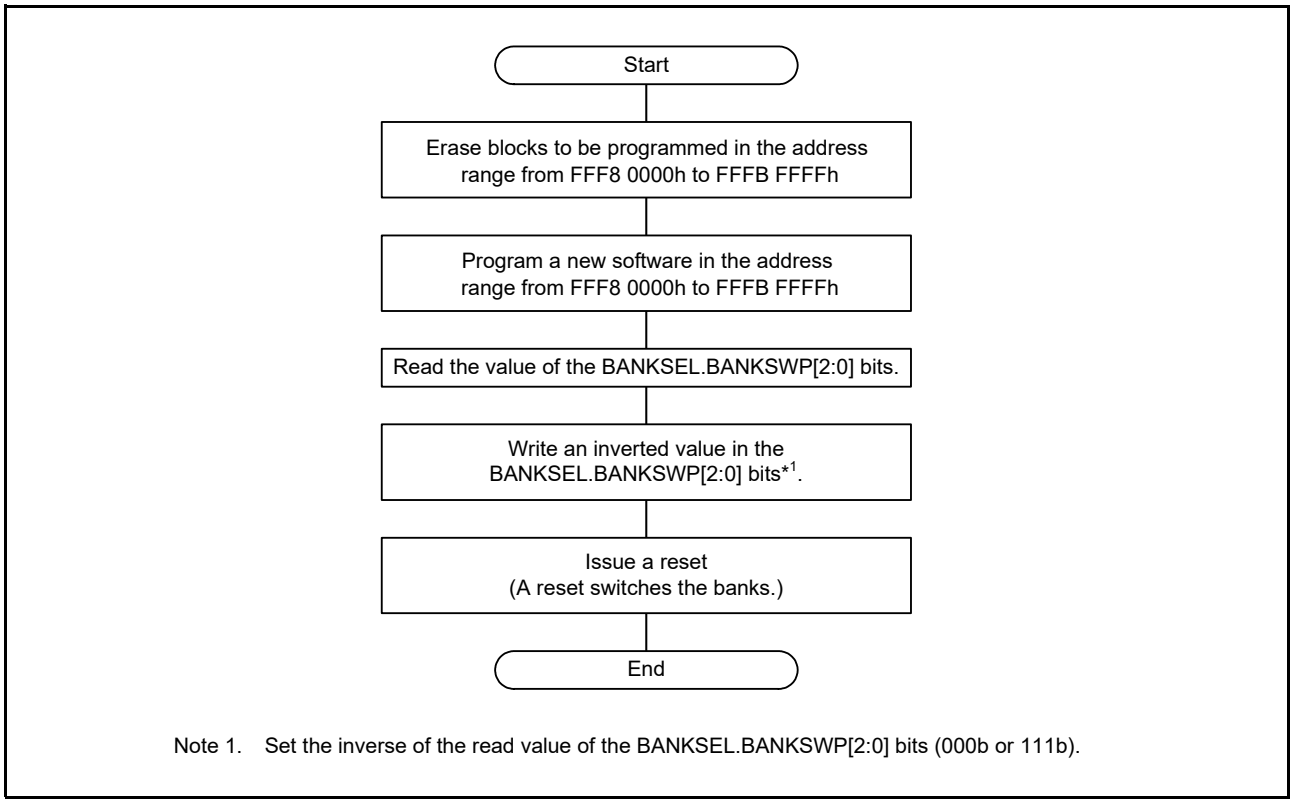


Figure 48.12 Example of Startup Bank Selection Flow (For Products with 512 Kbytes of Code Flash Memory)

48.5.7 Suspend Function

Reading from the flash memory without using the BGO function is not possible during program or erase operation. When a P/E suspend command is issued to suspend the program or erase operation to the flash memory, reading from the flash memory is enabled. Regarding P/E suspend commands, there are one program suspend mode and two erase suspend modes (suspend priority mode and erase priority mode). To resume suspended program or erase operation, the P/E resume command is available.

48.5.8 Trusted Memory

This MCU has a trusted memory (hereafter referred to as TM) function to prevent reading of blocks 8 and 9 (blocks 8, 9, 30, and 31 in dual mode) in the code flash memory by outsider software. When set as trusted memory, these areas are suitable for storing software that handles processing for encryption algorithms, device control software that is associated with proprietary or confidential know-how, purchased middleware, and so on.
Table 48.11 lists the specifications of the TM function, Table 48.12 lists access restrictions within the TM target area when the TM function is enabled, and Figure 48.13 shows the cases where the CPU is able to operate in relation to the TM target area.

Table 48.11 TM Specifications

Item	Description
TM target area	Linear mode: Blocks 8 and 9 in the code flash memory (32 Kbytes in total) Dual mode: Blocks 8 and 9 in the code flash memory (32 Kbytes in total) and blocks 30 and 31 (32 Kbytes in total)

Table 48.11 TM Specifications

Item	Description
Access restrictions when TM is enabled	Refer to Table 48.12, Restrictions on Access to the TM Target Area while the TM Function is Enabled.
How to run program code when the TM function is enabled	When the TM function is enabled, starting to run program code in an area being handled as TM is only possible with a branch instruction from program code outside the areas being handled as TM.
Interrupt processing during the execution of program code in an area being handled as TM while the TM function is enabled	Both the acceptance of requests for interrupt processing and return from interrupt processing are possible.
Security function	Enabling the TM function restricts access to program code in the areas for handling as TM to instruction fetching only
Protection functions	• Restrictions on data access to the TM target area when the TM function is enabled*1 • Once enabled, the TM function prevents its own disabling until the areas being handled as TM target area are erased. • Once enabled, the TM function prevents further writing to the areas being handled as TM target area

Note 1. Access to data in operations that include the borders of the TM target areas is also not allowed.

Table 48.12 Restrictions on Access to the TM Target Area while the TM Function is Enabled

Type of Access	CPU	DMAC/DTC
Instruction fetching	Yes	—
Access to data*1	No	No

Note: When using the on-chip debugger (OCD), the same restrictions apply as for the DMAC in the table above. For the OCD operation to the TM target area, refer to the manual of the emulator to be used.
Note 1. Place data outside the TM target areas.

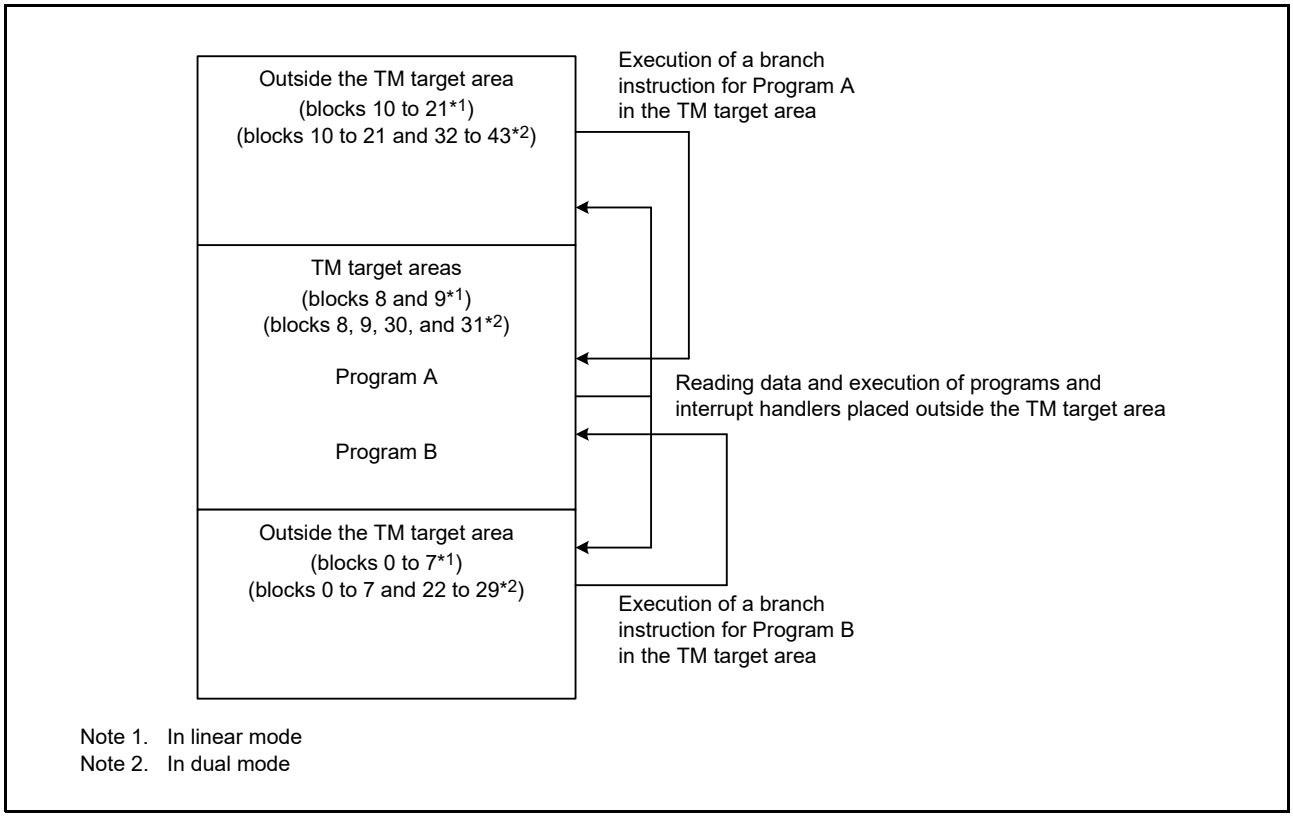


Figure 48.13 Cases where CPU is Allowed to Operate in Relation to the TM Target Area when the TM Function is Enabled

48.5.8.1 Allocating Program Code to the TM Target Area

When the TM function is enabled, implement countermeasures in the form of software for the TM target area as required as further measures to prevent the running of programs for access to consecutive addresses in the TM target areas from areas outside the TM target areas.

48.5.8.2 How to Enable the TM Function

(1) By Self-Programming

After programming to blocks 8 and 9 (blocks 8, 9, 30, and 31 in dual mode) of the code flash memory, i.e. the target areas for TM, use the configuration set command of the FACI to enable the TM function.
Figure 48.14 is a flowchart of the procedure for enabling the TM function by self-programming.

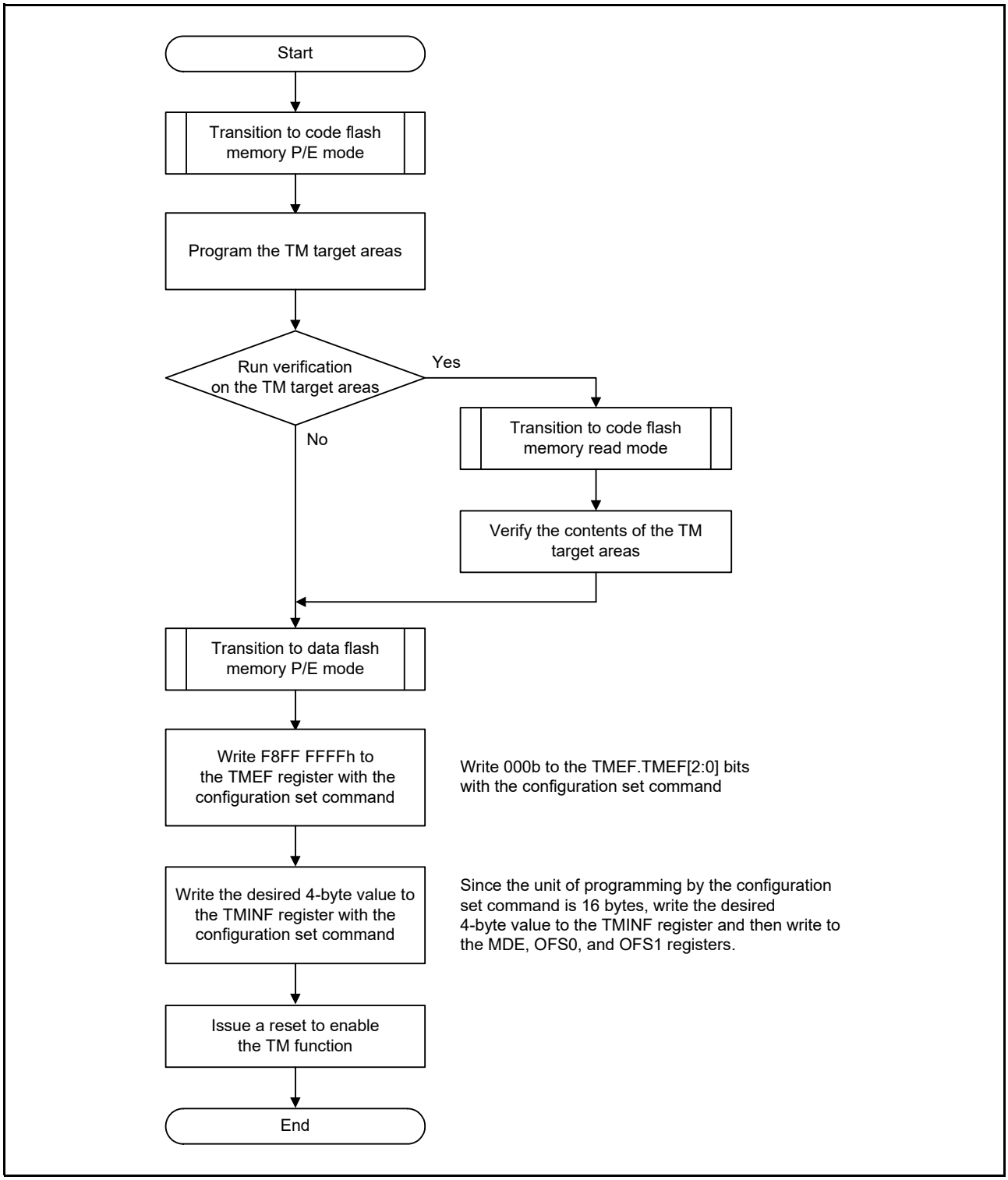


Figure 48.14 Flowchart for Enabling the TM Function by Self-Programming

(2) By Using Boot Mode

In boot mode, use the configuration program command among the boot commands to enable the TM function after programming to blocks 8 and 9 (blocks 8, 9, 30, and 31 in dual mode) of the code flash memory.
For the configuration program command among the boot commands, refer to section 48.8.22, Configuration Program Command.
Figure 48.15 is a flowchart of enabling the TM function in boot mode.

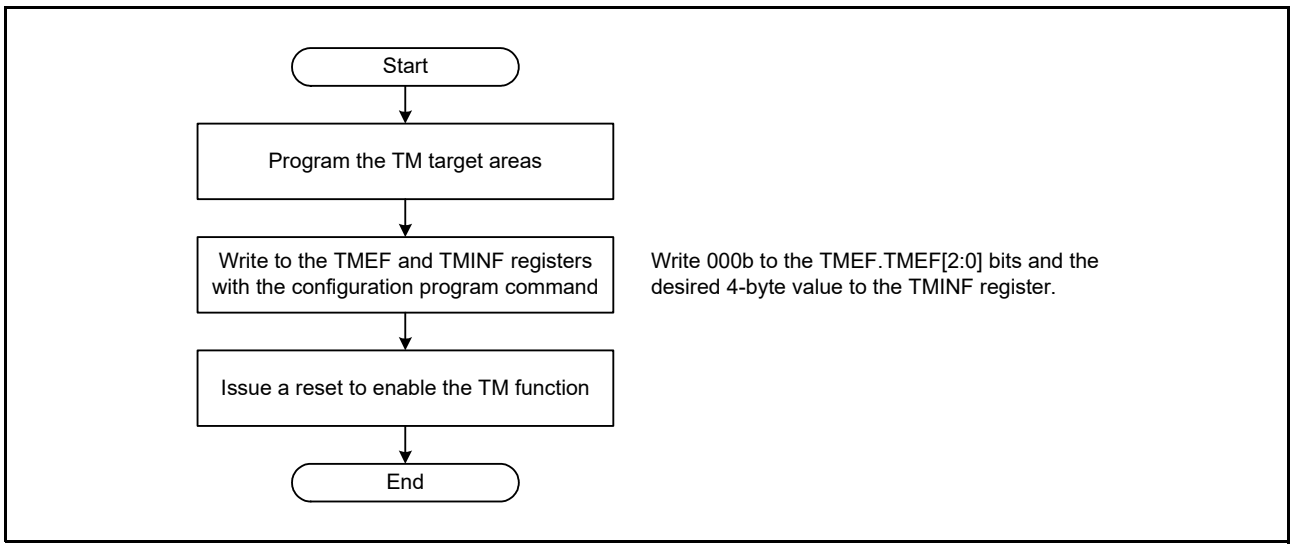
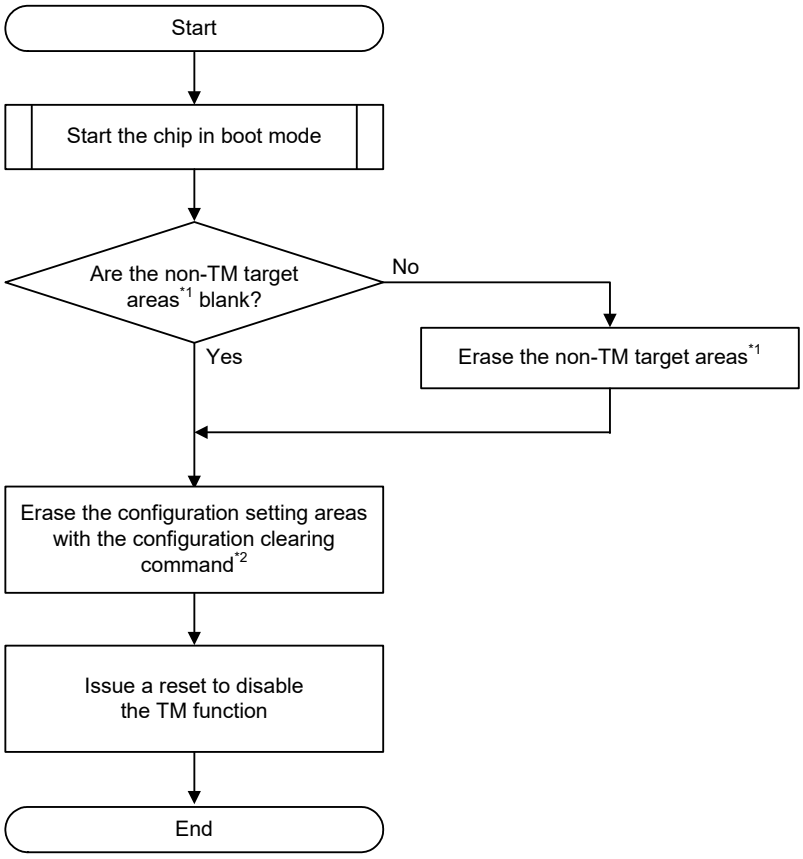


Figure 48.15 Flowchart for Enabling the TM Function by Using Boot Mode

48.5.8.3 How to Disable the TM Function

The TM function cannot be disabled unless the TM target areas are first erased with the configuration clear command. Do not use the configuration clear command for this purpose unless the TM function is to be disabled.
Figure 48.16 is a flowchart of disabling the TM function in boot mode.



Note 1. Non-TM target areas are the areas other than blocks 8 and 9 (blocks 8, 9, 30, and 31 in dual mode) of the code flash memory and data flash memory.
Note 2. The configuration clear command erases the configuration setting area including the TMEF and TMINF registers and the TM target areas.

Figure 48.16 Flowchart for Disabling the TM Function in Boot Mode

48.5.8.4 Notes on Enabling the TM Function

(1) Protection against Access to the TM Target Areas

While the TM function is enabled, the only type of access to the TM target areas is instruction fetching by the CPU, so do not allocate data to the TM target areas.
If the CPU, DMAC, DTC, or OCD attempt access to data in the TM target areas while the TM function is enabled, values are always read as 0 instead of the actual values.

(2) Further Programming to the TM Target Areas

Further programming to the TM target areas is not allowed as long as the TM function is enabled.
If you need to program further data, follow the procedure described in section 48.5.8.3, How to Disable the TM Function; and follow the procedure described in section 48.5.8.2, How to Enable the TM Function, to program blocks 8 and 9 (blocks 8, 9, 30, and 31 in dual mode) of the code flash memory and to re-enable the TM function.

(3) Executing the Configuration Clear Command

Follow the procedure in section 48.5.8.3, How to Disable the TM Function, before issuing the configuration clear command.

(4) When the MPU Setting is for Access to the TM Target Areas

When the TM function is enabled, even if the MPU is set to allow access to the TM target areas, the TM function takes priority.

(5) FACI Block Erase Command for the TM Target Areas

There are no special restrictions on block erase of the TM target areas with the FACI block erase command. Accordingly, since each area corresponds to an erase block, the areas can be erased by issuing the block erase command.

(6) Conditions for Correct Operation of the TM Function

The TM function operates normally under the conditions prescribed in section 49, Electrical Characteristics.

48.6 Flash Sequencer

48.6.1 Operating Modes of the Flash Sequencer

The flash sequencer has three operating modes as shown in Figure 48.17. Transitions between modes are initiated by changing the value of the FENTRYR register.
When the value of the FENTRYR register is 0000h, the flash sequencer is in read mode. In this mode, it does not receive FACI commands. Code flash memory and data flash memory are readable.
When the value of the FENTRYR register is 0001h, the flash sequencer is in code flash memory P/E mode where the code flash memory can be programmed or erased by FACI commands. Data flash memory is readable. Under the conditions where BGO is allowed, code flash memory is readable.
When the value of the FENTRYR register is 0080h, the flash sequencer is in data flash memory P/E mode where the data flash memory can be programmed or erased by FACI commands. In this mode, the data flash memory is not readable. However, the code flash memory is readable.

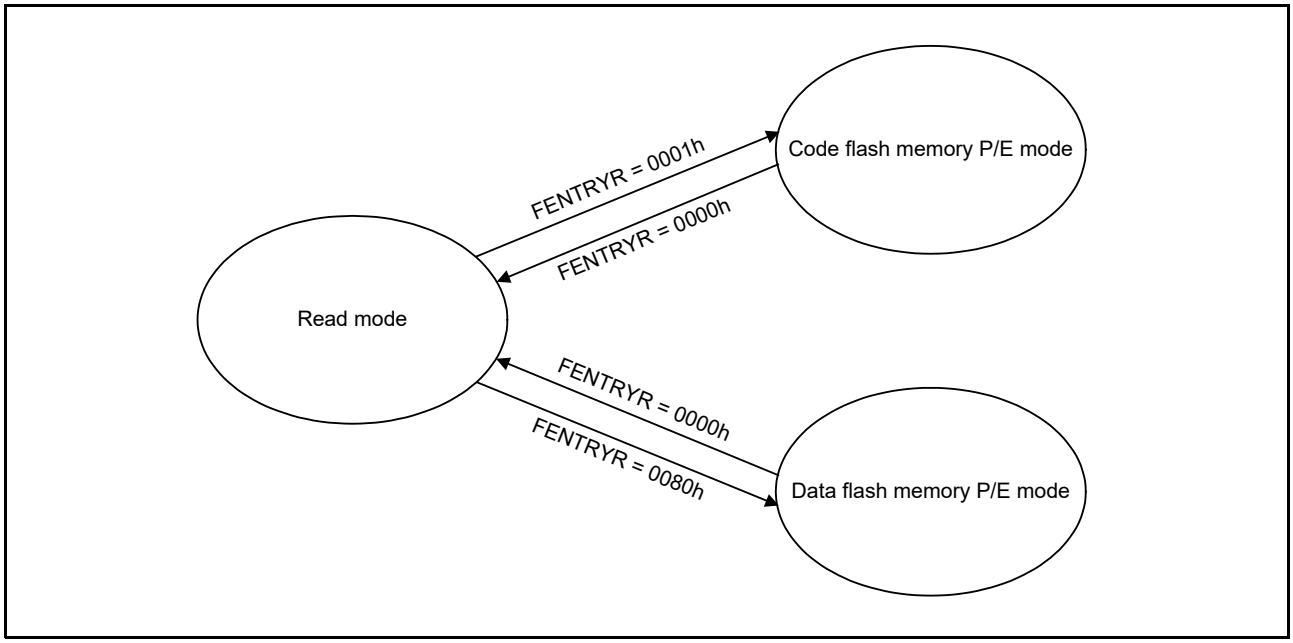


Figure 48.17 Modes of the Flash Sequencer

48.6.2 Read Mode

Read mode is for high-speed reading of the code flash memory or data flash memory.

(1) Code Flash Memory

Special settings are not required to read code flash memory in single-chip mode. Data can simply be read out through access to addresses in the code flash memory.

When reading code flash memory that has been erased but not yet been programming again (i.e. that is in the non-programmed state), all bits are read as 1.

(2) Data Flash Memory

Special settings are not required to read data flash memory in single-chip mode. Data can simply be read out through access to addresses in the data flash memory.

Values read from data flash memory that has been erased but not yet been programming again (i.e. that is in the non-programmed state) are undefined. Use a blank check command when you need to confirm that an area is in the non-programmed state.

48.6.3 Code Flash Memory P/E Mode

In this mode, the code flash memory is in P/E mode, and the data flash memory is in read mode. The sequencer enters this mode when setting the FENTRYR.FENTRYD bit to 0 and the FENTRYR.FENTRYC bit to 1.

48.6.4 Data Flash Memory P/E Mode

In this mode, the code flash memory is in read mode, and the data flash memory is in P/E mode. The sequencer enters this mode when setting the FENTRYR.FENTRYD bit to 1 and the FENTRYR.FENTRYC bit to 0.

48.6.5 Transitions of Operating Modes

Each FACI command can be accepted in a specific mode or state of the flash sequencer. FACI commands should be issued after the transition of the flash sequencer to the code flash memory P/E mode or data flash memory P/E mode and checking of the state of the flash sequencer. Use the FSTATR and FASTAT registers to check the state of the flash sequencer. The value of the FASTAT.CMDLK flag is the logical OR of values of the ILGLERR, ERSERR, PRGERR, and FLWEERR flags in the FSTATR register, and the CFAE and DFAE flags in the FASTAT register. Therefore, the occurrence of errors can be checked by reading the value of the FASTAT.CMDLK flag.
Table 48.13 lists the available commands in each operating mode.

Table 48.13 Operating Mode and Available Commands

Operating Mode	FENTRYR Register Value	Available Commands
Read mode	0000h	None
Code flash memory P/E mode	0001h	Program Block erase P/E suspend P/E resume Status clear Forced stop
Data flash memory P/E mode	0080h	Program Block erase P/E suspend P/E resume Status clear Forced stop Blank check Configuration set

Table 48.14 shows the state of the flash sequencer and acceptable FACI commands. An appropriate mode is assumed to be set before the commands are executed.

Table 48.14 Acceptable FACI Commands and the State of the Flash Sequencer

	Processing of Program or Erase*1	Processing of Configuration Set	Suspending Program or Erase	Blank checking	During Program Suspend	During Erase Suspend	Program during Erase Suspend	Command-Locked State (FRDY = 1)	Command-Locked State (FRDY = 0)	Processing of Forced Stop Command	Other State
FRDY flag	0	0	0	0	1	1	0	1	0	0	1
SUSRDY flag	1	0	0	0	0	0	0	0	0	0	0
ERSSPD flag	0	0	0/1	0/1	0	1	1	0/1	0/1	0	0
PRGSPD flag	0	0	0/1	0/1	1	0	0	0/1	0/1	0	0
CMDLK flag	0	0	0	0	0	0	0	1	1	0	0
Program	×	×	×	×	×	✓*2	×	×	×	×	✓
Block erase	×	×	×	×	×	×	×	×	×	×	✓
P/E suspend	✓	×	×	×	×	×	×	—	×	×	—
P/E resume	×	×	×	×	✓	✓	×	×	×	×	×
Status clear	×	×	×	×	✓	✓	×	✓	×	×	✓
Forced stop	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Blank check	×	×	×	×	✓*3	✓*3	×	×	×	×	✓*3
Configuration set	×	×	×	×	×	×	×	×	×	×	✓*3

✓: Acceptable
×: Not acceptable (the sequencer in the command-locked state)
—: Ignored
Note 1. Cases that the program or erase operation is completed before the P/E suspend command is accepted are also included.
Note 2. Program command can be issued to a block other than the erase-suspended block.
Note 3. Acceptable only in data flash memory P/E mode

48.6.5.1 Transition to Code Flash Memory P/E Mode

To use the FACI commands for the code flash memory, a transition to code flash memory P/E mode is required. To enter code flash memory P/E mode, set the FENTRYR.FENTRYC bit to 1.

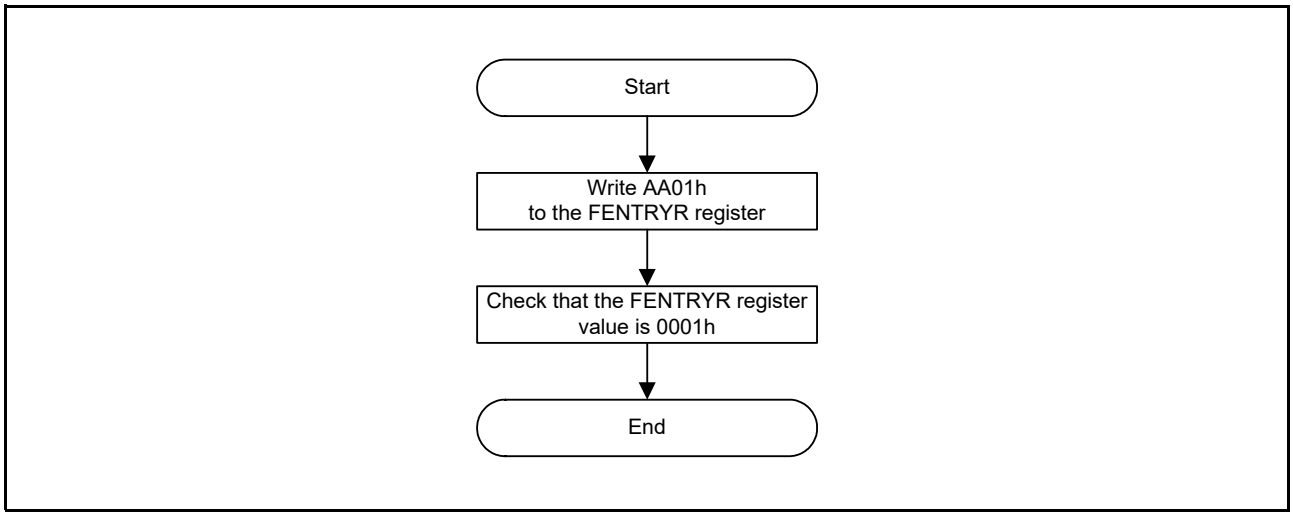


Figure 48.18 Procedure for Transition to Code Flash Memory P/E Mode

48.6.5.2 Transition to Data Flash Memory P/E Mode

To use the FACI commands for the data flash memory, a transition to data flash memory P/E mode is required. To enter data flash memory P/E mode, set the FENTRYR.FENTRYD bit to 1.

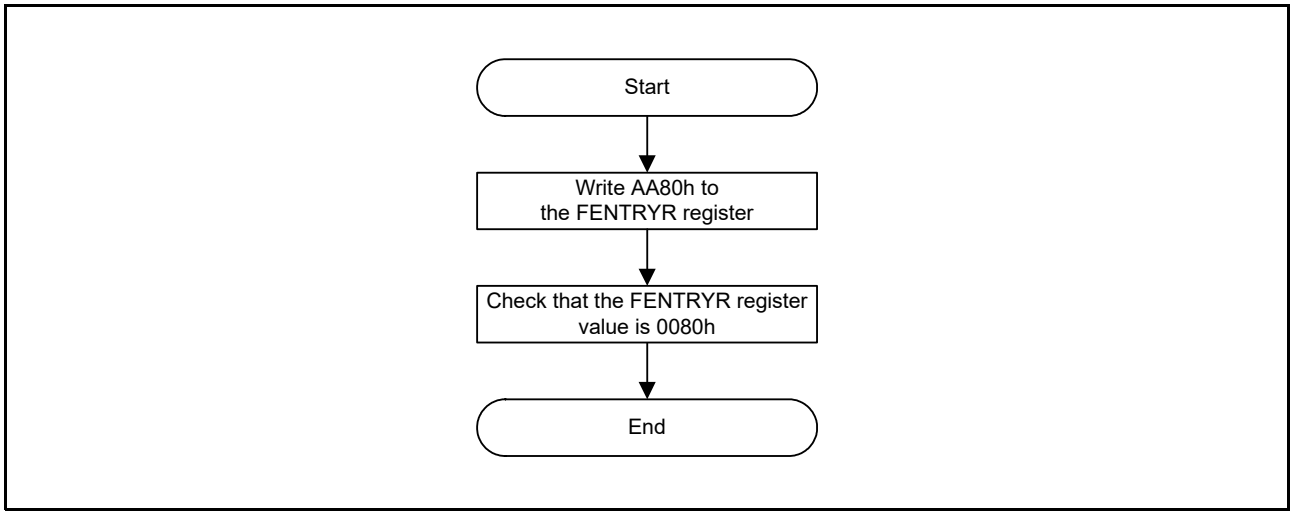
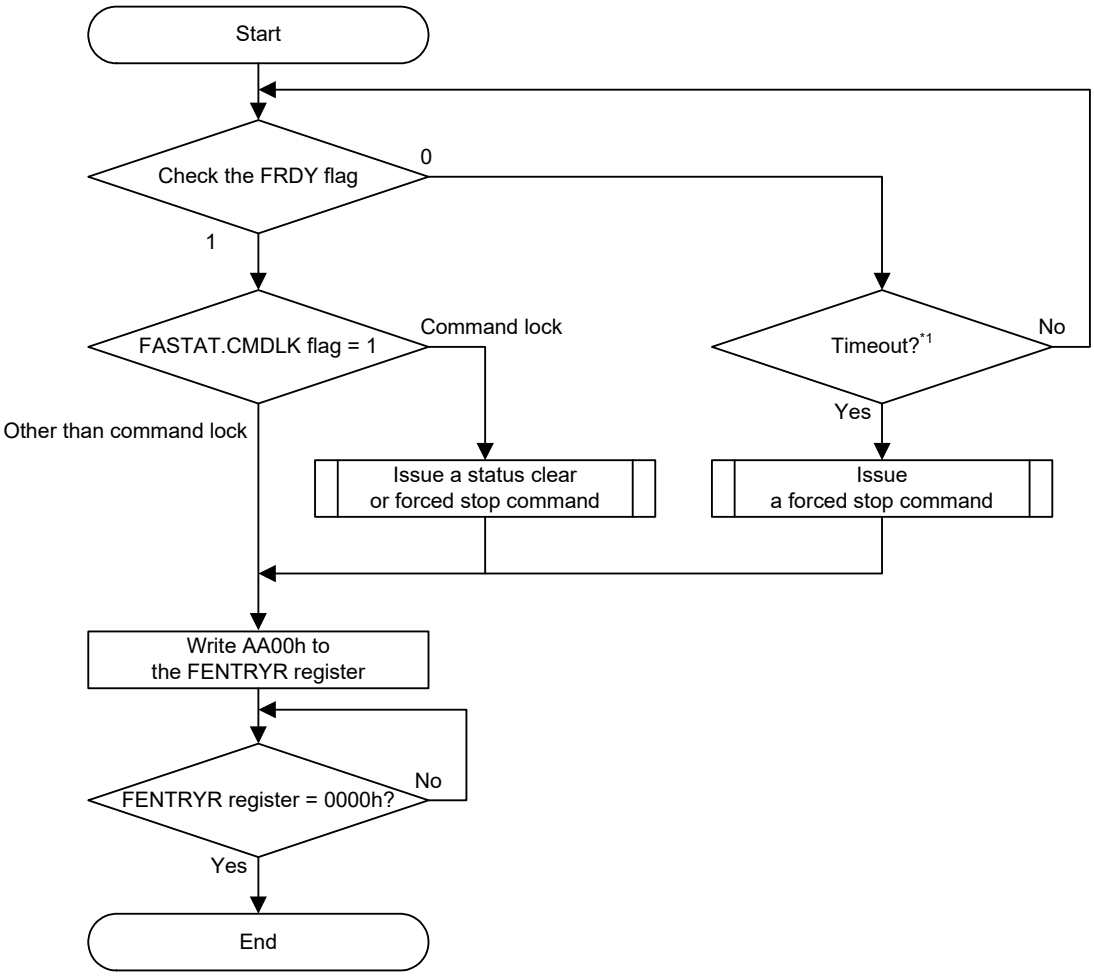


Figure 48.19 Procedure for Transition to Data Flash Memory P/E Mode

48.6.5.3 Transition to Read Mode

To read the flash memory without using the BGO function, a transition to read mode is required. To return to read mode, set the FENTRYR register to 0000h. The transition to read mode should be made after processing by the flash sequencer is completed and while operation is in other than in the command-locked state. In addition, operation is started in read mode after release from the reset state.



Note 1. Judgment of the timeout is based on 1.1 times the maximum time for processing of the FACI command that is in progress (refer to section 49, Electrical Characteristics).

Figure 48.20 Procedure for Transition to Read Mode

48.6.6 List of FACI Commands

Table 48.15 List of FACI Commands

FACI Command	Description
Program	This is used to program the code flash memory and data flash memory. The program unit is 128 bytes for the code flash memory and 4 bytes for the data flash memory.
Block erase	This is used to erase the code flash memory and data flash memory. The erase unit is one block. (code flash memory: 4 K or 16 Kbytes, data flash memory: 64 bytes)
P/E suspend	This suspends program or erase operation.
P/E resume	This resumes suspended program or erase operation.
Status clear	This initializes the ILGLERR, ERSERR, and PRGERR flags in the FSTATR register, and the CMDLK flag in the FASTAT register and releases the flash sequencer from the command-locked state.
Forced stop	This forcibly stops processing of FACI commands and initializes the FSTATR register.
Blank check	This is used to check if data flash memory is blank. Units of blank checking: 4 bytes to 16 Kbytes (specified in 4-byte units).
Configuration set	This is used to set the configuration setting area. Units of setting: 16 bytes.

The FACI commands are issued by writing to the FACI command-issuing area (refer to Table 48.2). When write access as shown in Table 48.16 proceeds in the specified state, the flash sequencer executes the processing corresponding to the given command (refer to section 48.6.5, Transitions of Operating Modes).

Table 48.16 FACI Command Formats

FACI Commands	Number of Write Access	Data to be Written to the FACI Command-Issuing Area			
		1st Access	2nd Access	3rd to (N+2)th Access	(N+3)th Access
Program (code flash memory) 128-byte programming, N = 64	67	E8h	40h (= N)	WD_1 to WD_{64}	D0h
Program (data flash memory) 4-byte programming: N = 2	5	E8h	02h (= N)	WD_1 and WD_2	D0h
Block erase	2	20h	D0h	—	—
P/E suspend	1	B0h	—	—	—
P/E resume	1	D0h	—	—	—
Status clear	1	50h	—	—	—
Forced stop	1	B3h	—	—	—
Blank check	2	71h	D0h	—	—
Configuration set N = 8	11	40h	08h (= N)	WD_1 to WD_8	D0h

Note: WD_N (N = 1, 2,…): Nth 16-bit data to be programmed.

The flash sequencer clears the FSTATR.FRDY flag to 0 at the start of processing of a command other than the status clear command and sets this bit to 1 upon completion of command processing.
When the setting of the FRDYIE.FRDYIE bit is 1 and when the FSTATR.FRDY flag is set to 1, a flash ready (FRDY) interrupt is generated.

48.6.7 Usage of FACI Commands

This section gives an overview of the usage of FACI commands.

48.6.7.1 Overview Flow when FACI Command is Used

Figure 48.21 shows an overview flow when the FACI command is used.
When using the BGO function, the jump to the internal RAM (other than code flash memory) is not required because an FACI command can be issued for the code or data flash memory by using the rewriting program in the code flash memory.
When the FCLK is changed, changing the FPCKAR register shortens time for processing the FACI command. For details, refer to section 48.4.17, Flash Sequencer Processing Clock Frequency Notification Register (FPCKAR).

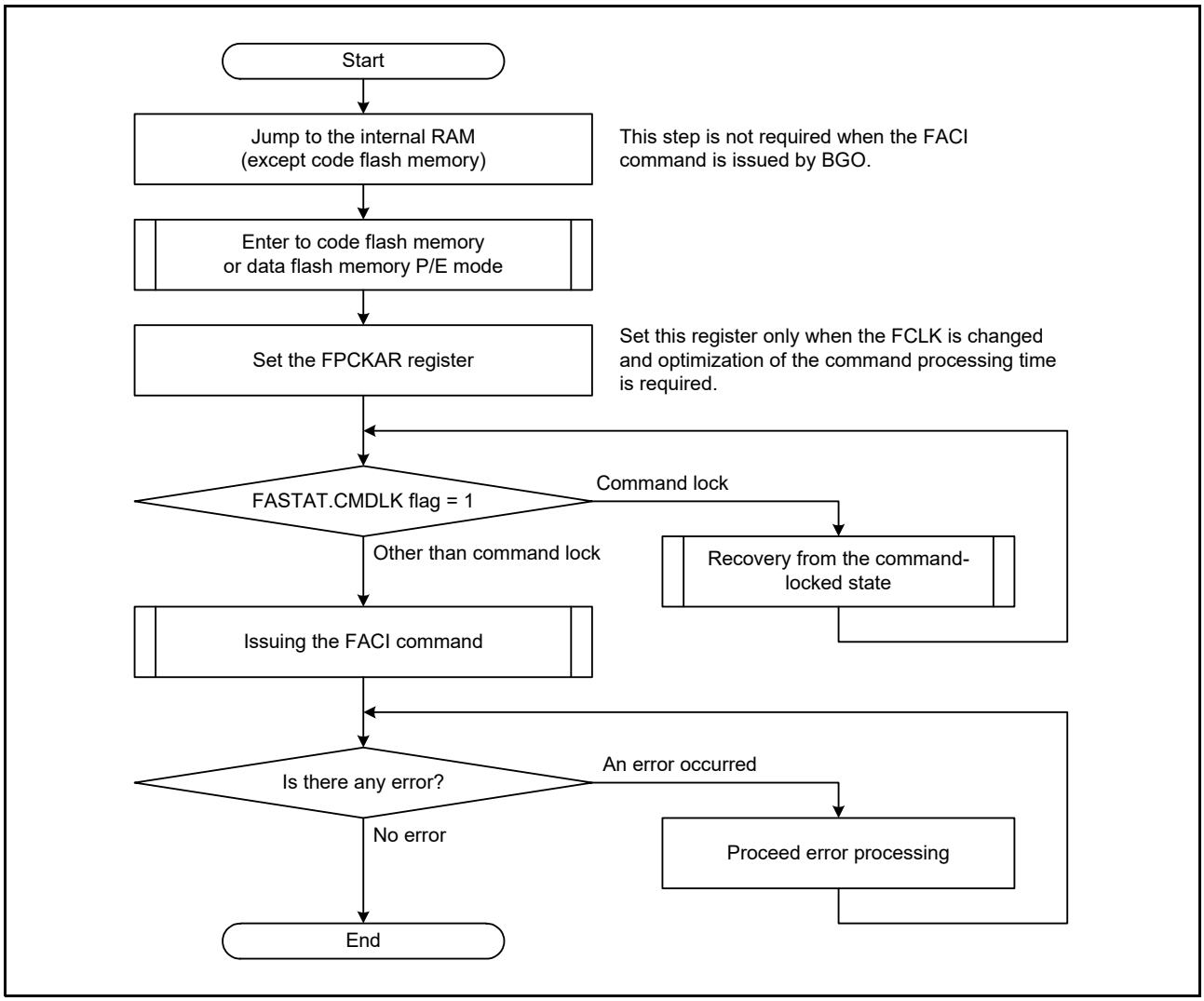
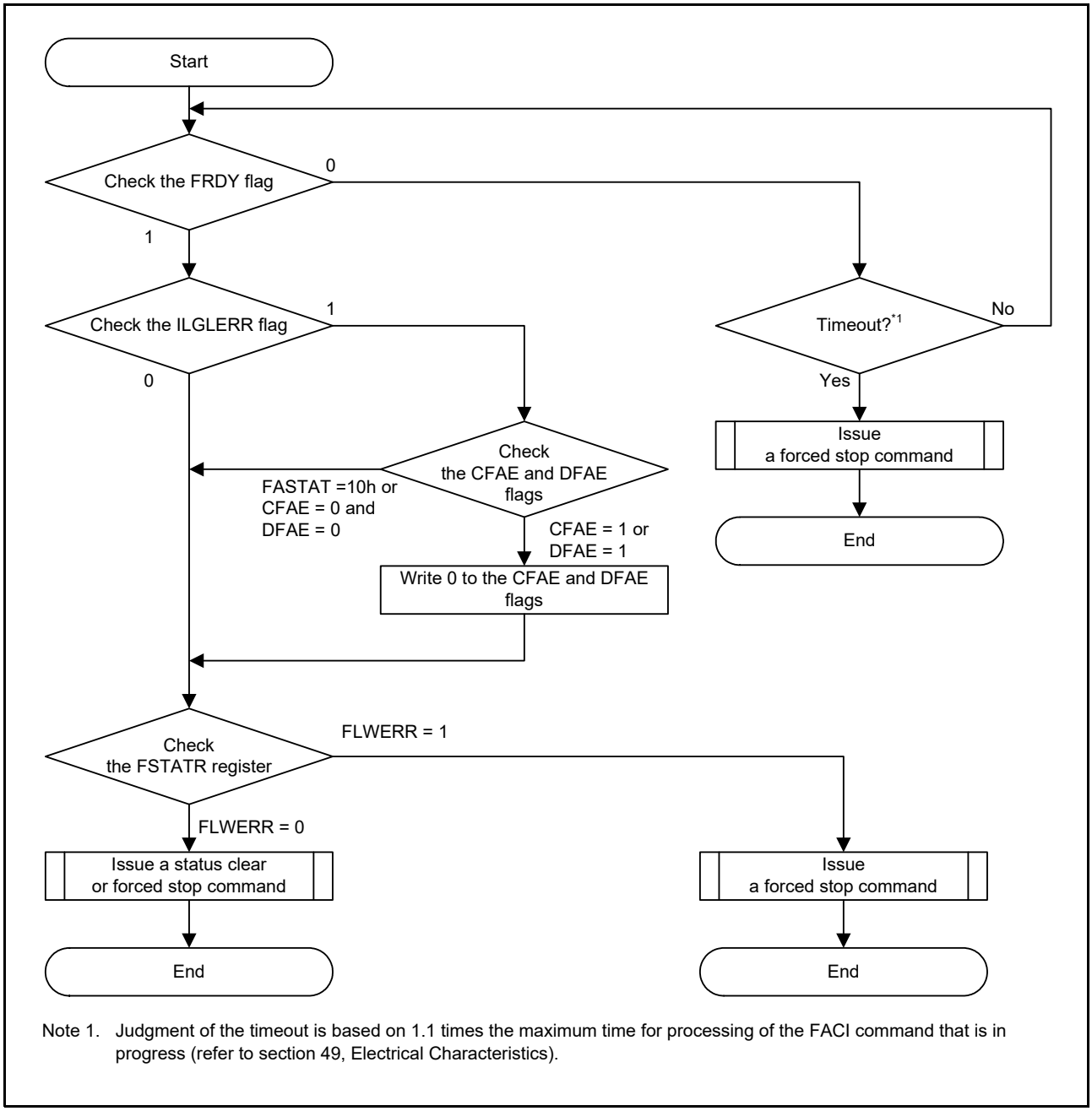


Figure 48.21 Overview Flow when FACI Command is Used

48.6.7.2 Recovery from the Command-Locked State

When the flash sequencer enters the command-locked state, FACI commands cannot be accepted. To release the sequencer from the command-locked state, use the status clear command, or forced stop command, or FASTAT register. When the command-locked state is detected by checking for an error before issuing the P/E suspend command, the FSTATR.FRDY flag may hold 0 as the command processing has not been completed. If processing is not completed within the maximum program/erase time specified in section 49, Electrical Characteristics, this can be considered a timeout, and the flash sequencer should be stopped by the forced stop command.
When the FSTATR.ILGLERR flag is 1, check the FASTAT value. If the CFAE or DFAE flag in the FASTAT register is 1, set the CFAE or DFAE flag in the FASTAT register to 0 and then issue the status clear and forced stop commands.
The FSTATR.FLWEERR flag is not changed from 1 to 0 by the status clear command. When this flag is set to 1, use the forced stop command for release from the command-locked state. The other bits that indicate the command-locked state can be changed from 1 to 0 by the status clear or forced stop command.

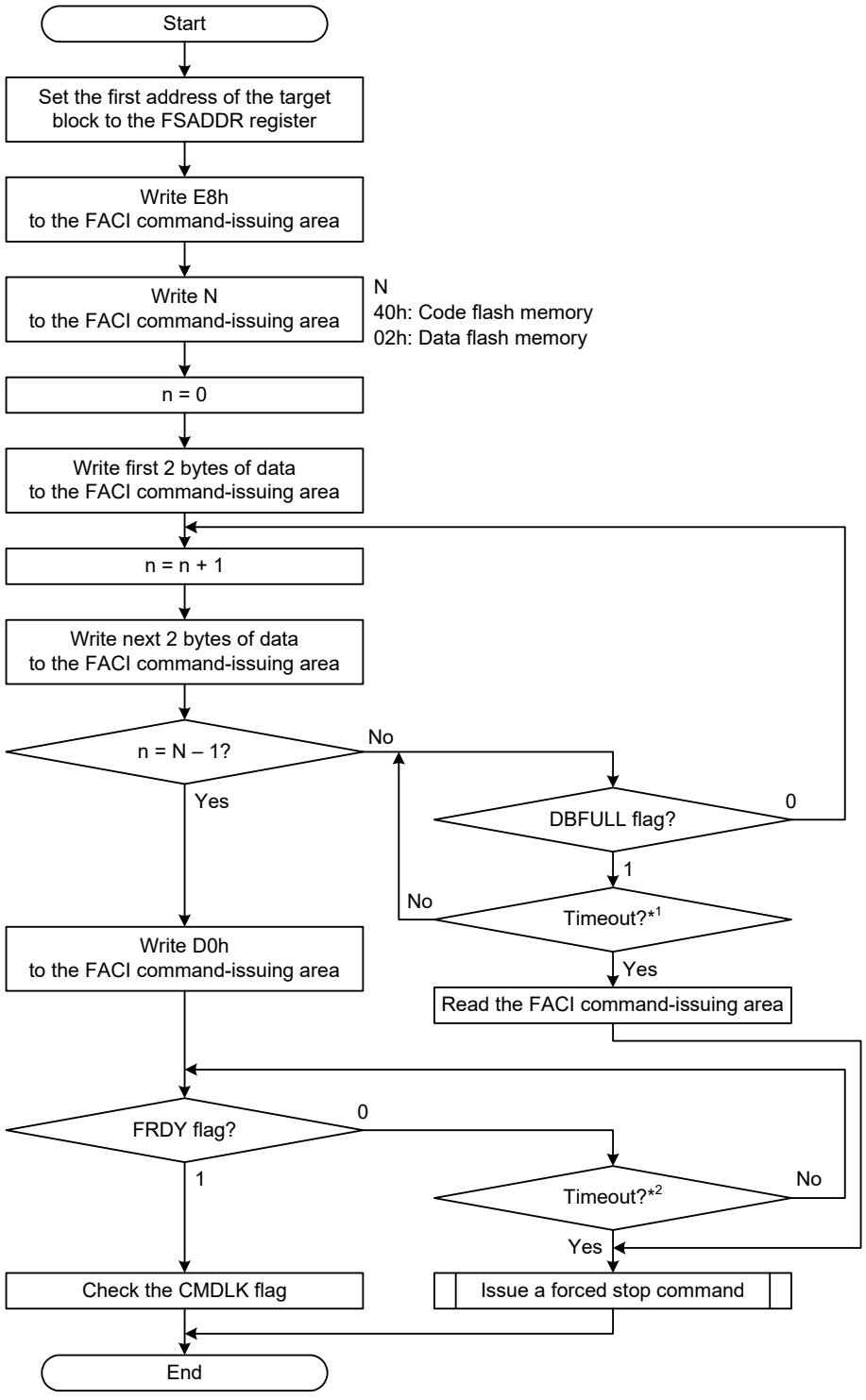


Note 1. Judgment of the timeout is based on 1.1 times the maximum time for processing of the FACI command that is in progress (refer to section 49, Electrical Characteristics).

Figure 48.22 Recovery from the Command-Locked State

48.6.7.3 Program Command

A program command is used for writing to the code flash memory and data flash memory.
Before issuing a program command, set the first address of the target block in the FSADDR register.
Writing D0h to the FACI command-issuing area at the final access of the FACI command-issuing starts the program command processing. Completion of command processing can be checked by reading the FSTATR.FRDY flag. If the target area of program command processing contains the area not for writing, write FFFFh to the corresponding area.
Issuing a program command consecutively while the FACI internal data buffer is full leads to a wait on the peripheral bus 6 and this may affect on the bus accesses of the other peripheral IP modules. To avoid the generation of such a wait, issue an FACI command while the FSTATR.DBFULL flag is 0.
Writing to the data flash memory will not lead to the data buffer becoming full.

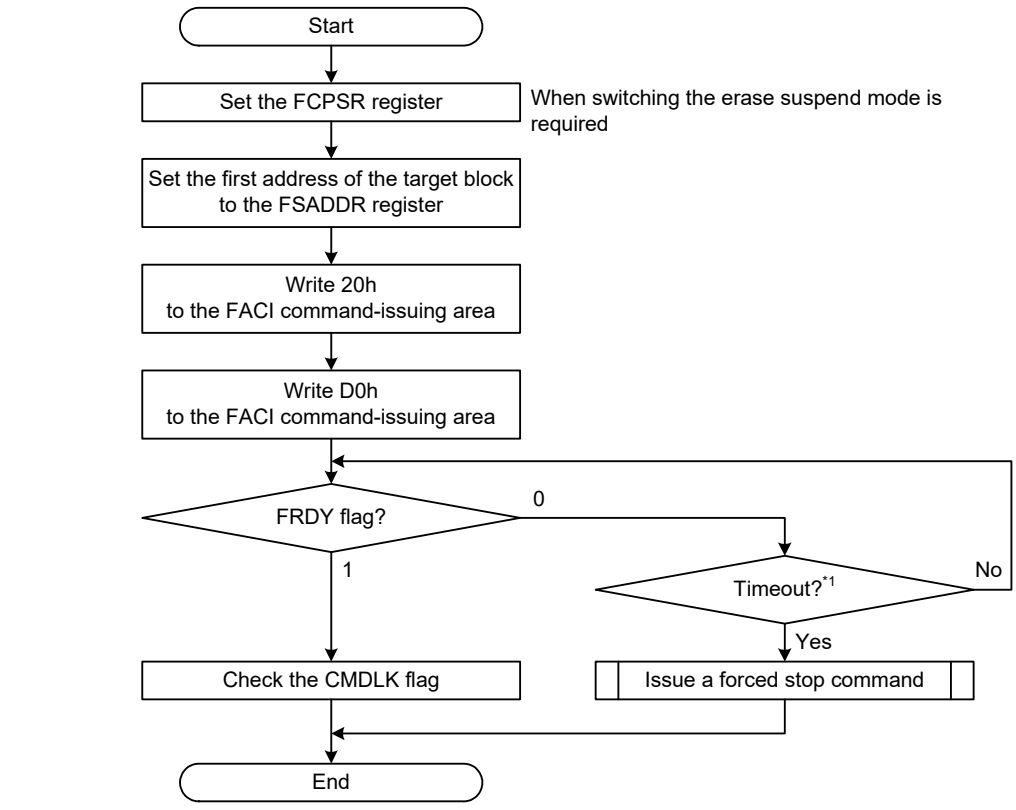


Note 1. Judgment of the timeout is based on 1.4 µs elapsing (when FCLK ≥ 20 MHz)
Note 2. Judgment of the timeout is based on 1.1 times 128-byte programming time (max) for code flash memory.
For data flash memory, judgment is based on 1.1 times 4-byte programming time (max).
For the 128-byte programming time and 4-byte programming time, refer to section 49, Electrical Characteristics.

Figure 48.23 Usage of the Program Command

48.6.7.4 Block Erase Command

A block erase command is used to erase the code flash memory, lock bit, and data flash memory in single-block units. Before issuing a block erase command, set the first address of the target block in the FSADDR register. Writing 20h and D0h to the FACI command-issuing area starts processing of a block erase command. Completion of command processing can be checked by reading the FSTATR.FRDY flag.
The FCPSR register must be set before issuing the block erase command. The setting of the FCPSR register must be changed to switch the suspending method (suspend priority mode/erase priority mode) by the P/E suspend command.



Note 1. Judgment of the timeout is based on 1.1 times 16-Kbyte erase time (max) for code flash memory. For data flash memory, judgment is based on 1.1 times 64-byte erase time (max). For the 16-Kbyte erase time and 64-byte erase time, refer to section 49, Electrical Characteristics.

Figure 48.24 Usage of the Block Erase Command

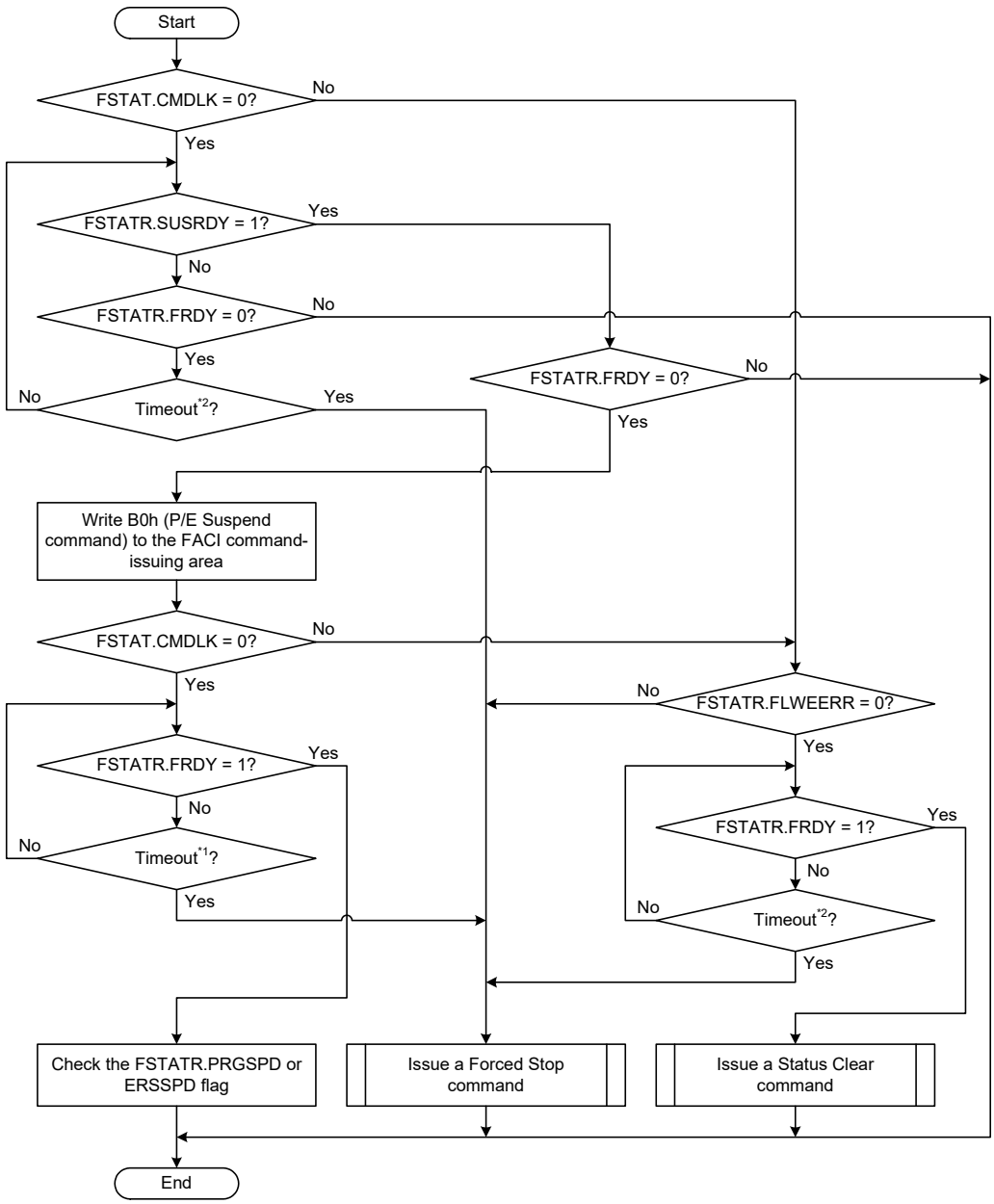
48.6.7.5 P/E Suspend Command

The P/E suspend command is used to suspend an in-progress program or erase operation. Before issuing a P/E suspend command, check that the FASTAT.CMDLK flag is 0, and the execution of program/erase is normally performed. To confirm that the P/E suspend command can be received, also check that the FSTATR.SUSRDY flag is 1. After issuing a P/E suspend command, read the FASTAT.CMDLK flag to confirm that its value is not 1 (the flash sequencer is not in the command-locked state).

If an error occurs during program or erase processing, the FASTAT.CMDLK flag is set to 1. When P/E processing is completed between the FSTATR.SUSRDY flag having been confirmed to be 1 and acceptance of the P/E suspend command, the P/E suspend command is ignored and the flash sequencer does not enter the suspended state (the FSTATR.FRDY flag is 1 and the ERSSPD and PRGSPD flags in the FSTATR register are 0).

When a P/E suspend command is received and then the program/erase suspend processing finishes normally, the flash sequencer enters the suspended state, the FSTATR.FRDY flag is set to 1, and the ERSSPD or PRGSPD flag in the FSTATR register is 1. After issuing a P/E suspend command, check that the ERSSPD or PRGSPD flag in the FSTATR register is 1 and the suspended state is entered, and then decide the subsequent flow. If a P/E resume command is issued in the subsequent flow although the suspended state is not entered, an illegal command error occurs and the flash sequencer moves to the command-locked state (refer to section 48.5.3.2, Error Protection).

If the erase-suspended state is entered, program to blocks other than the one currently suspended can be performed. In addition, the program- and erase-suspended states can move to read mode by clearing the FENTRYR register.



Note 1. For both the code flash and data flash memory, judgment of the timeout is based on 1.1 times the maximum value of the erase suspend latency.

Note 2. For both the code flash and data flash memory, judgment of the timeout is based on 1.1 times the maximum value of the erase time.

For the erase suspend latency and the erase time, refer to section 49, Electrical Characteristics.

Figure 48.25 Usage of the P/E Suspend Command

(1) Suspension during Program Operation

When a P/E suspend command is issued during program operation to flash memory, the flash sequencer suspends the program operation. Figure 48.26 shows the program suspend operation.

When the flash sequencer accepts a program or P/E resume command, it sets the FSTATR.FRDY flag to 0 and starts program sequence. When the flash sequencer is ready to accept the P/E suspend command after the program sequence has started, it sets the FSTATR.SUSRDY flag to 1. When a P/E suspend command is issued, the flash sequencer accepts the command and sets the FSTATR.SUSRDY flag to 0. When the flash sequencer receives a P/E suspend command while a program pulse is being applied, it continues to apply the pulse. After the specified pulse application time, the flash sequencer finishes applying program pulse, starts the suspend processing, and sets the FSTATR.PRGSPD flag to 1. When the suspend processing is completed, the flash sequencer sets the FSTATR.FRDY flag to 1 and enters the program-suspended state. When the flash sequencer receives a P/E resume command in the program-suspended state, it sets the FSTATR.FRDY and PRGSPD flags to 0 and resumes the program sequence.

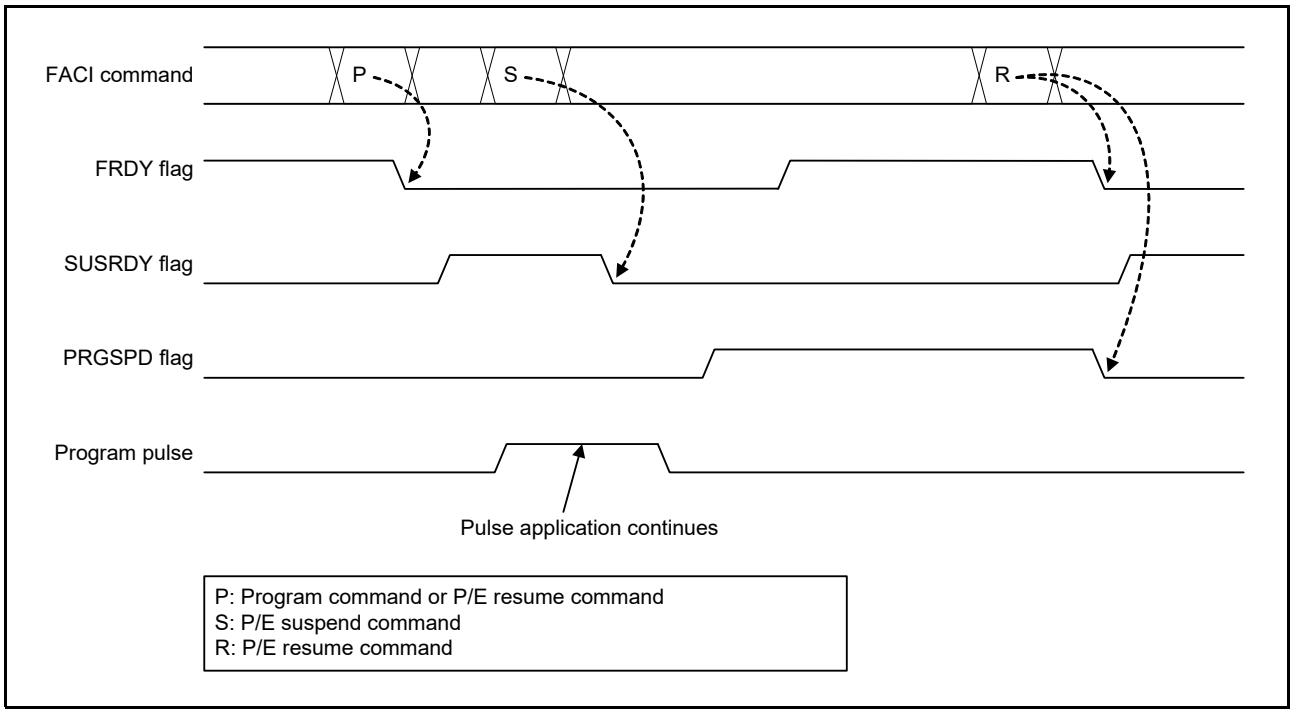


Figure 48.26 Program Suspend Operation

(2) Suspension during Erase Operation (Suspend Priority Mode)

This MCU supports the suspend priority mode as a suspend method during erase operation. Figure 48.27 shows the erase suspend operation in suspend priority mode (the FCPSR.ESUSPMD bit is set to 0).

When the flash sequencer accepts a block erase or P/E resume command, it sets the FSTATR.FRDY flag to 0 and starts erase sequence. When the flash sequencer is ready to accept the P/E suspend command after the erase sequence has started, it sets the FSTATR.SUSRDY flag to 1. When a P/E suspend command is issued, the flash sequencer accepts the command and sets the FSTATR.SUSRDY flag to 0. When the flash sequencer receives a P/E suspend command during erase sequence, the flash sequencer starts the suspend processing even while an erase pulse is being applied and sets the FSTATR.ERSSPD flag to 1. When the suspend processing is completed, the flash sequencer sets the FSTATR.FRDY flag to 1 and enters the erase-suspended state. When the flash sequencer receives a P/E resume command in the erase-suspended state, it sets the FSTATR.FRDY and ERSSPD flags to 0 and resumes the erase sequence. The behavior of the FSTATR.FRDY, SUSRDY, and ERSSPD flags when the erase sequence is suspended or resumed is the same regardless of the erase suspend mode.

The setting of the erase suspend mode affects the control method of erase pulses. In suspend priority mode, when a P/E suspend command is received while an erase pulse A, which has never been interrupted, is being applied, the flash sequencer stops applying erase pulse A and enters the erase-suspended state. When the flash sequencer receives a P/E suspend command again while the erase sequence is resumed by a P/E resume command and erase pulse A is being re-applied, it continues to apply erase pulse A. After the specified pulse application time, the flash sequencer finishes applying erase pulse A and enters the erase-suspended state. When the flash sequencer accepts a P/E resume command again and starts applying a new erase pulse B, and then accepts a P/E suspend command again, it stops applying erase pulse B.

In suspend priority mode, the application of an erase pulse is interrupted once per pulse to give priority to the suspend processing, thus reducing the suspend latency.

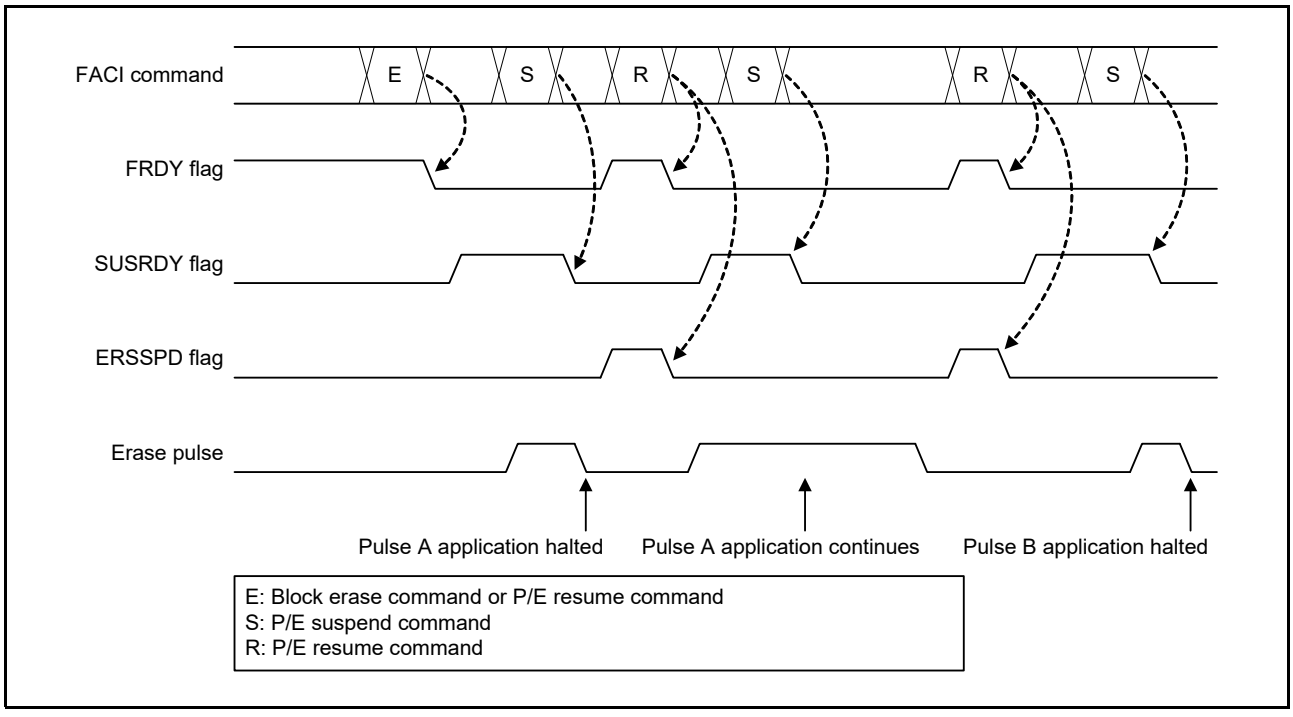


Figure 48.27 Erase Suspend Operation (Suspend Priority Mode)

(3) Suspension during Erase Operation (Erase Priority Mode)

This MCU supports the erase priority mode as a suspend method during erase operation.

Figure 48.28 shows the erase suspend operation in erase priority mode (the FCPSR.ESUSPMD bit is set to 1). The control method of erase pulses in erase priority mode is the same as that of program pulses for the program suspend processing.

When the flash sequencer receives a P/E suspend command while an erase pulse is being applied, the flash sequencer continues to apply the pulse. In this mode, the erase pulse is not re-applied when a P/E resume command is issued, thus reducing the overall time required for the erase operation compared to the suspend priority mode.

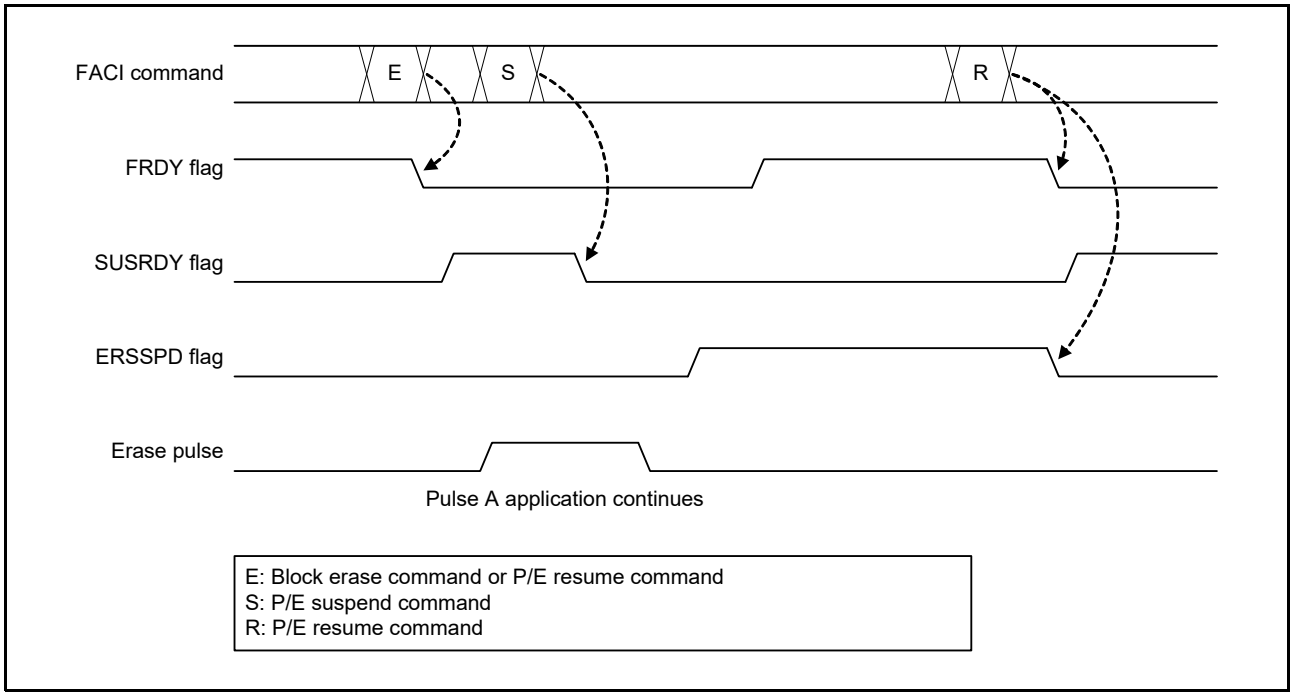
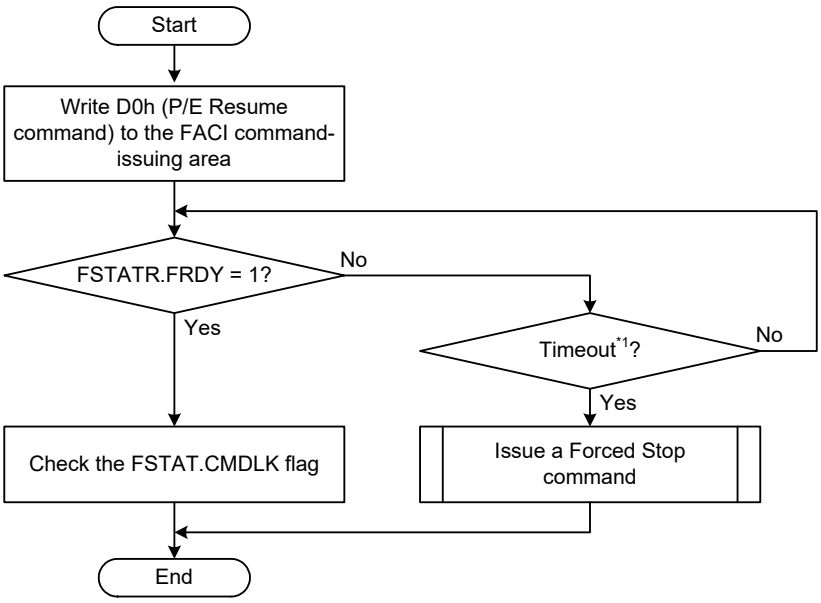


Figure 48.28 Erase Suspend Operation (Erase Priority Mode)

48.6.7.6 P/E Resume Command

To resume the suspended program or erase operation, use the P/E resume command. If the FENTRYR register setting is changed during suspend, reset the FENTRYR register value to the value immediately before the P/E suspend command was issued, and then issue a P/E resume command. Completion of processing of the resumed command can be confirmed by the FSTATR.FRDY flag.



Note 1. Judgment of the timeout is based on 1.1 times the maximum time for halted processing of the FACI command (refer to section 49, Electrical Characteristics).

Figure 48.29 Usage of the P/E Resume Command

48.6.7.7 Status Clear Command

When one of the ILGLERR, ERSERR, PRGERR, and FLWEERR flag bits in the FSTATR register is set to 1, the flash sequencer enters the command-locked state. When one of the CFAE and DFAE flags in the FASTAT register is set to 1, the flash sequencer also enters the command lock state. In the command-locked state, the flash sequencer can accept only status clearing command or forced end command.

The status clear command is used to clear the command-locked state (refer to section 48.6.7.2, Recovery from the Command-Locked State). To clear the CMDLK flags in the FASTAT register, and the ILGLERR, ERSERR, and PRGERR flags in the FSTATR register in the command-locked state, the status clear command is available.

The FSTATR.FLWEERR flag cannot only be cleared by the status clearing command, but can only be cleared by the forced end command.

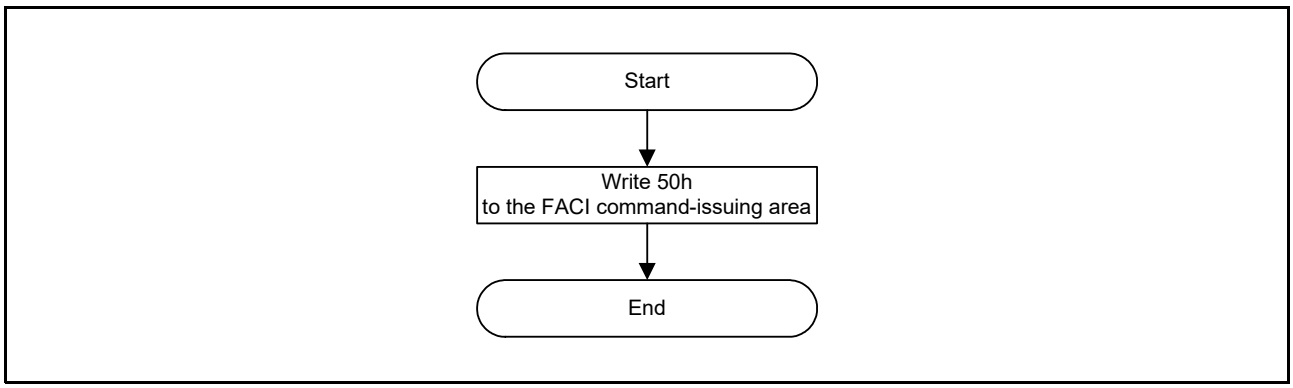
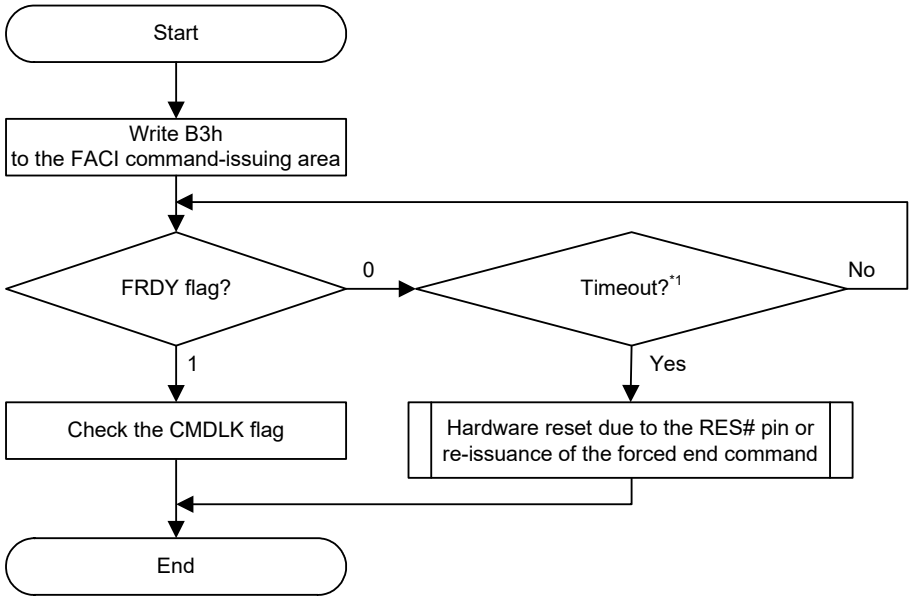


Figure 48.30 Usage of the Status Clear Command

48.6.7.8 Forced Stop Command

The forced stop command forcibly terminates command processing by the flash sequencer. This command can interrupt the command processing faster than the P/E suspend command, but the values of the area being programmed or erased are not guaranteed. Also, the interrupted operation cannot be resumed. A program or erase operation terminated by a forced stop command is also defined as one program/erase cycle.
Executing the forced stop command initializes the entire flash sequencer and part of the FACI. It also initializes the FASTAT.CMDLK flag and the FSTATR register. Therefore, this command can also be used in the recovery procedure from the command-locked state and the time-out process of the flash sequencer (refer to section 48.6.7.2, Recovery from the Command-Locked State).



Note 1. Judgment of the timeout is based on 1.1 times the maximum time for the forced stop command (refer to section 49, Electrical Characteristics).

Figure 48.31 Usage of the Forced Stop Command

How to Use the Forced Stop Command when a Command is being issued

When processing is terminated by the forced stop command while a timeout is generated based on the DBFULL bit judgment of the program command, writing to the FACI command issuing area may be handled as the data written by the program command. In this case, read the FACI command issuing area and generate a command lock intentionally, then issue the forced stop command according to the method of returning from the command lock state. A command lock can be generated although the access size for reading the FACI command issuing area is in 8-, 16-, or 32-bit units.

48.6.7.9 Blank Check Command

Values read from data flash memory that has been erased but not yet been programming again (i.e. that is in the nonprogrammed state) are undefined. Use the blank check command when you need to confirm that an area is in the nonprogrammed state.

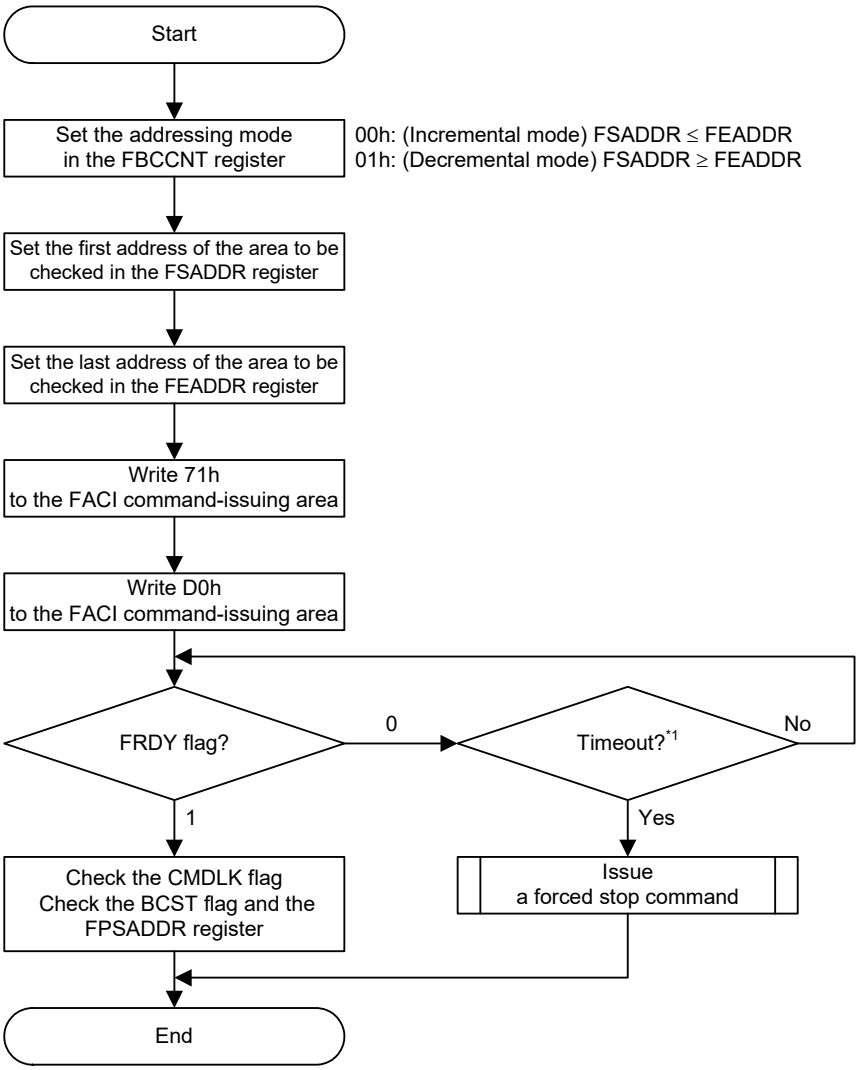
Before issuing a blank check command, set addressing mode, start and end addresses of the target area for blank checking to the FBCCNT, FSADDR, and FEADDR registers.

When the FBCCNT.BCDIR bit is 1, the value specified in the FSADDR register must be set at least the value specified in the FEADDR register.

When the FBCCNT.BCDIR bit is 0, the value specified in the FSADDR register must be the value specified in the FEADDR register or less.

When the settings of the FBCCNT.BCDIR bit, FSADDR register, and FEADDR register are inconsistent, the flash sequencer enters the command-locked state. The size of the target area for blank checking is in the range from 4 bytes to 16 Kbytes and is set in units of 4 bytes.

Write 71h and D0h to the FACI command-issuing area to start blank checking. Completion of processing can be confirmed by the FSTATR.FRDY flag. At the end of processing, the result of blank checking is stored in the FBCSTAT.BCST flag. If the target area for blank checking includes areas where programming has been completed, the flash sequencer stores the address of the programmed data that it first detected in the FPSADDR register.

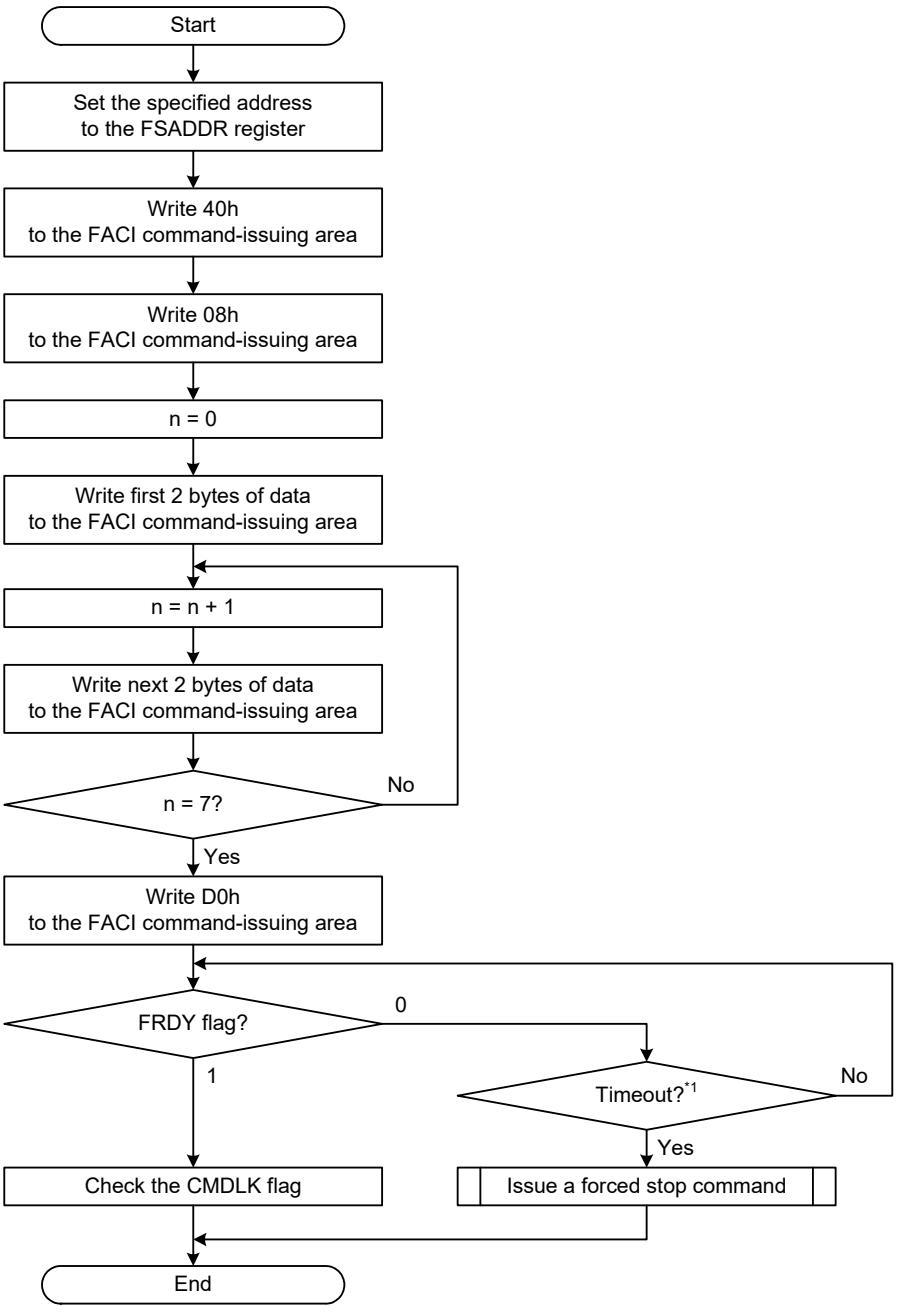


Note 1. Judgment of the timeout is based on 1.1 times the blank checking time (max.) for the entire target area. For the blank checking time, refer to section 49, Electrical Characteristics.

Figure 48.32 Usage of the Blank Check Command

48.6.7.10 Configuration Set Command

The configuration set command is used to set the configuration setting area.
Before issuing a configuration set command, set the specified address (shown in Table 48.17) in the FSADDR register.
Writing D0h to the FACI command-issuing area in the final access for issuing the FACI command starts processing of the configuration set command.



Note 1. Timeout is judged to be after 70ms (when 20 MHz ≤ FCLK ≤ 60 MHz), or 140 ms (when FCLK = 4 MHz).

Figure 48.33 Usage of the Configuration Set Command

The correspondence between the possible target data for configuration setting and the address value set in the FSADDR register is shown in Table 48.17.

Table 48.17 Address Used by Configuration Set Command

Address	FSADDR Register Value	Setting Data	Operation of additional writing		Timing when the Setting is Enabled
			FAW.FSPR bit is 1	FAW.FSPR bit is 0	
0012 0040h	0000 0040h	• Serial programmer command control register (SPCC) • TM enable flag register (TMEF)	Writable*1 (from 1 to 0 only)	Writable*1 (from 1 to 0 only)	After a reset or when command is executed*2
0012 0050h	0000 0050h	• OCD/serial programmer ID setting register (OSIS)	Writable	Writable	After a reset
0012 0060h	0000 0060h	• TM identification data register (TMINF) • Endian select register (MDE) • Option function select register 0 (OFS0) • Option function select register 1 (OFS1)	Writable	Writable	After a reset
0012 0090h	0000 0090h	• Bank select register (BANKSEL)	Writable	Writable	After a reset
0012 00A0h	0000 00A0h	• Flash access window setting register (FAW)*3	Writable	Not writable*3	After a reset or when command is executed

Note 1. Once these bits are set to 0, the bits cannot be restored to 1 by using the configuration setting command.
Note 2. The setting in the serial programmer command control register (SPCC) is enabled after a reset. The setting of the TM enable flag register (TMEF) is enabled after a reset or when command is executed.
Note 3. The FAW.FSPR bit cannot be restored to 1 once it is set to 0. Therefore, setting the access window and start-up area select bits again becomes impossible. (when the configuration setting command is issued to the address of 0012 00A0h, the flash sequencer enters the command-locked state.) Exercise extra caution when handling the FAW.FSPR bit.

48.7 Boot Mode

There are two serial programming modes; the boot mode (SCI interface) with SCI and the boot mode (FINE interface) with FINE. Table 48.18 lists the I/O pins used in boot mode.
I/O pins that are not used in boot mode are in the same state as that after a reset.

Table 48.18 I/O Pins Used in Boot Mode

Pin Name	I/O	Mode to be Used	Use
PN6/MD	Input	Boot mode (SCI interface)	Selection of operating mode
PD5/RXD1	Input		For host communication (to receive data through SCI)
PD3/TXD1	Output		For host communication (to transmit data through SCI)
PN6/MD/FINED	I/O	Boot mode (FINE interface)	Selection of operating mode, FINE data I/O

48.7.1 Boot Mode (SCI Interface)

In boot mode (SCI interface), the host sends control commands and data for programming, and the code flash memory and data flash memory can be programmed or erased. An on-chip SCI handles transfer between the host and this MCU in asynchronous mode. Tools for transmission of control commands and the data for programming must be prepared in the host.
When this MCU is activated in boot mode (SCI interface), the program (boot program) in a dedicated area within the MCU is executed. The boot program automatically adjusts the bit rate of the SCI and controls program/erase operation by receiving control commands from the host.
Figure 48.34 shows the system configuration for operations in boot mode (SCI interface).

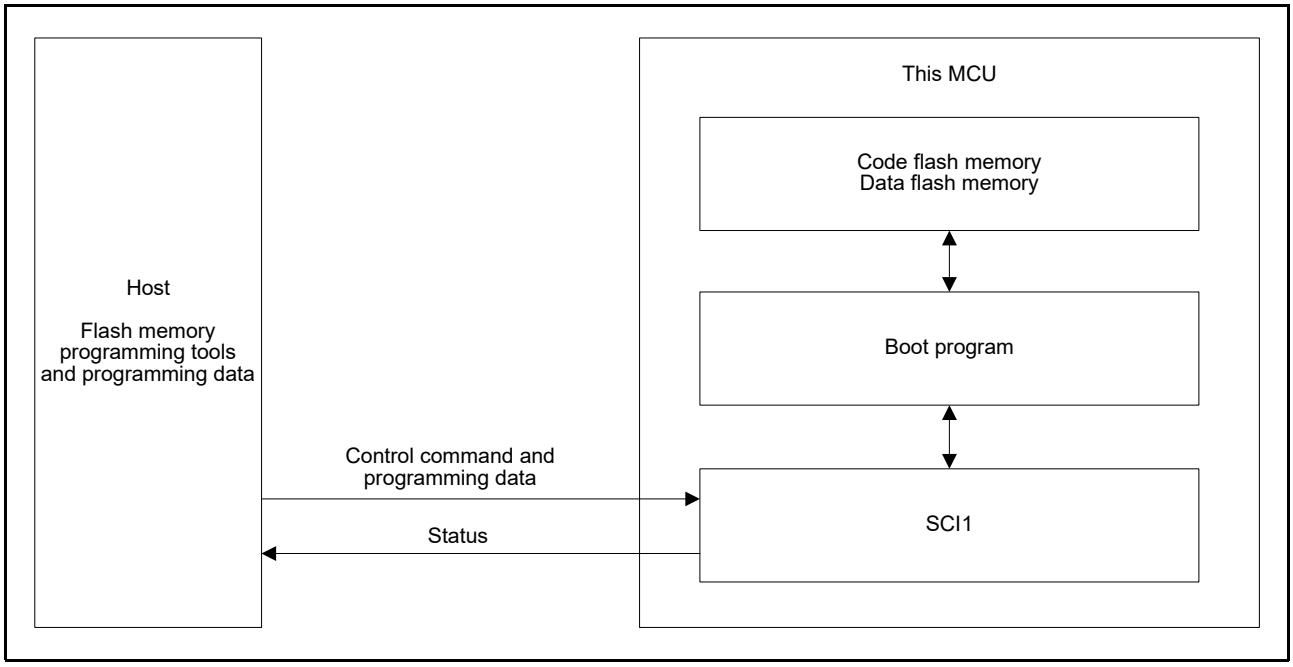


Figure 48.34 System Configuration for Operations in Boot Mode (SCI Interface)

48.7.2 Boot Mode (FINE Interface)

The flash memory can be programmed and erased using the FINE in boot mode (FINE interface). The code flash memory and data flash memory can be rewritten.

48.7.2.1 Operating Conditions in Boot Mode (FINE Interface)

FINE is used to communicate with the serial programmer in boot mode (FINE Interface).
Figure 48.35 shows an Example of Pin Connections in Boot Mode (FINE Interface). Table 48.19 lists Pin Handling in Boot Mode (FINE Interface).
The example of pin connections shown in Figure 48.35 is a simplified circuit. Operations are not guaranteed in all systems.

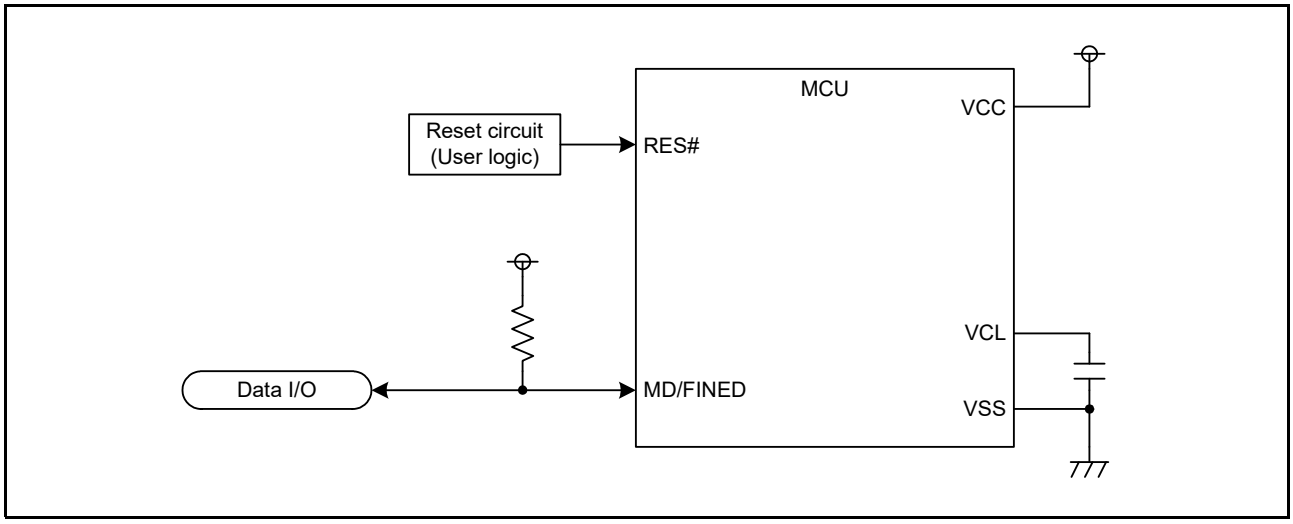


Figure 48.35 Example of Pin Connections in Boot Mode (FINE Interface)

Table 48.19 Pin Handling in Boot Mode (FINE Interface)

Pin Name	Name	I/O	Function
VCC, VSS	Power supply input	Input	Input 2.7 V or higher to the VCC pin. Input 0 V to the VSS pin.
VCL	Decoupling capacitor connect pin	—	Connect to the VSS pin via a 0.47-μF multilayer ceramic capacitor for stabilizing the internal voltage.
MD/FINED	Operating mode control/data I/O	I/O	Connect the VCC pin via a resistor (pull up).
RES#	Reset input	Input	Reset pin. Connect to the reset circuit.

48.8 Boot Mode Communications Protocol

This section describes the communications protocol for use in boot mode. When developing a serial programmer, use this communications protocol to control it.

48.8.1 How to Start the Chip Up in Boot Mode

(1) How to Start the Chip Up in Boot Mode (SCI Interface)

The chip starts up in boot mode (SCI interface) if the MD pin is at the low level on release from the reset state (i.e. when the level on the RES# pin changes from low to high). A waiting time of at least 400 ms is required while the RES# pin is held at the high level after the chip starts up in boot mode (SCI interface) until communications with the MCU can proceed.

Figure 48.36 shows the states of pins up to communications in boot mode (SCI interface) becoming possible.

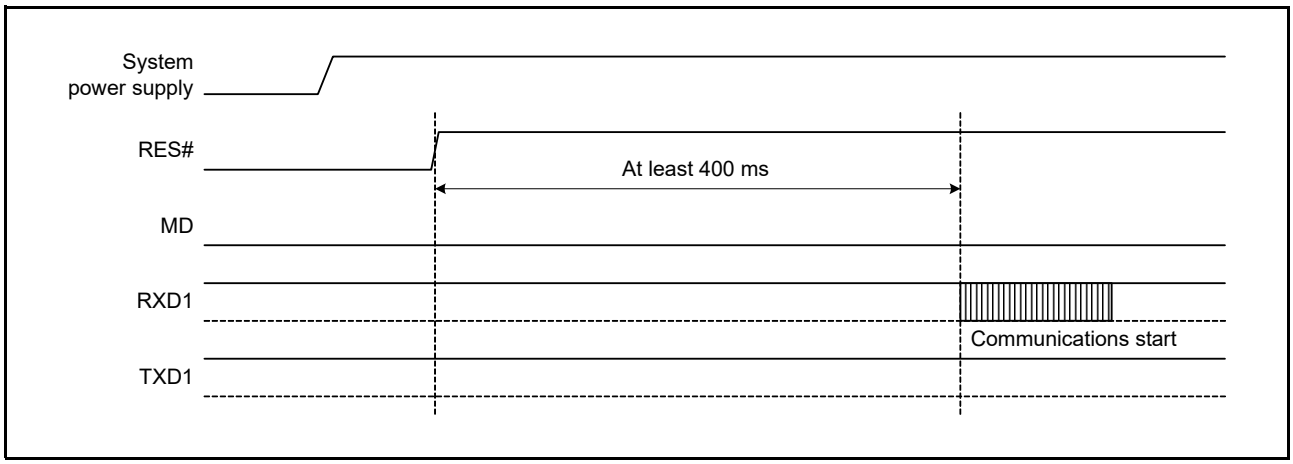


Figure 48.36 States of Pins Up to Communications in Boot Mode (SCI Interface) Becoming Possible

48.8.2 State Transitions in Boot Mode

48.8.2.1 State Transitions in Boot Mode (SCI Interface)

Figure 48.37 shows a flowchart for transition to boot mode (SCI interface).

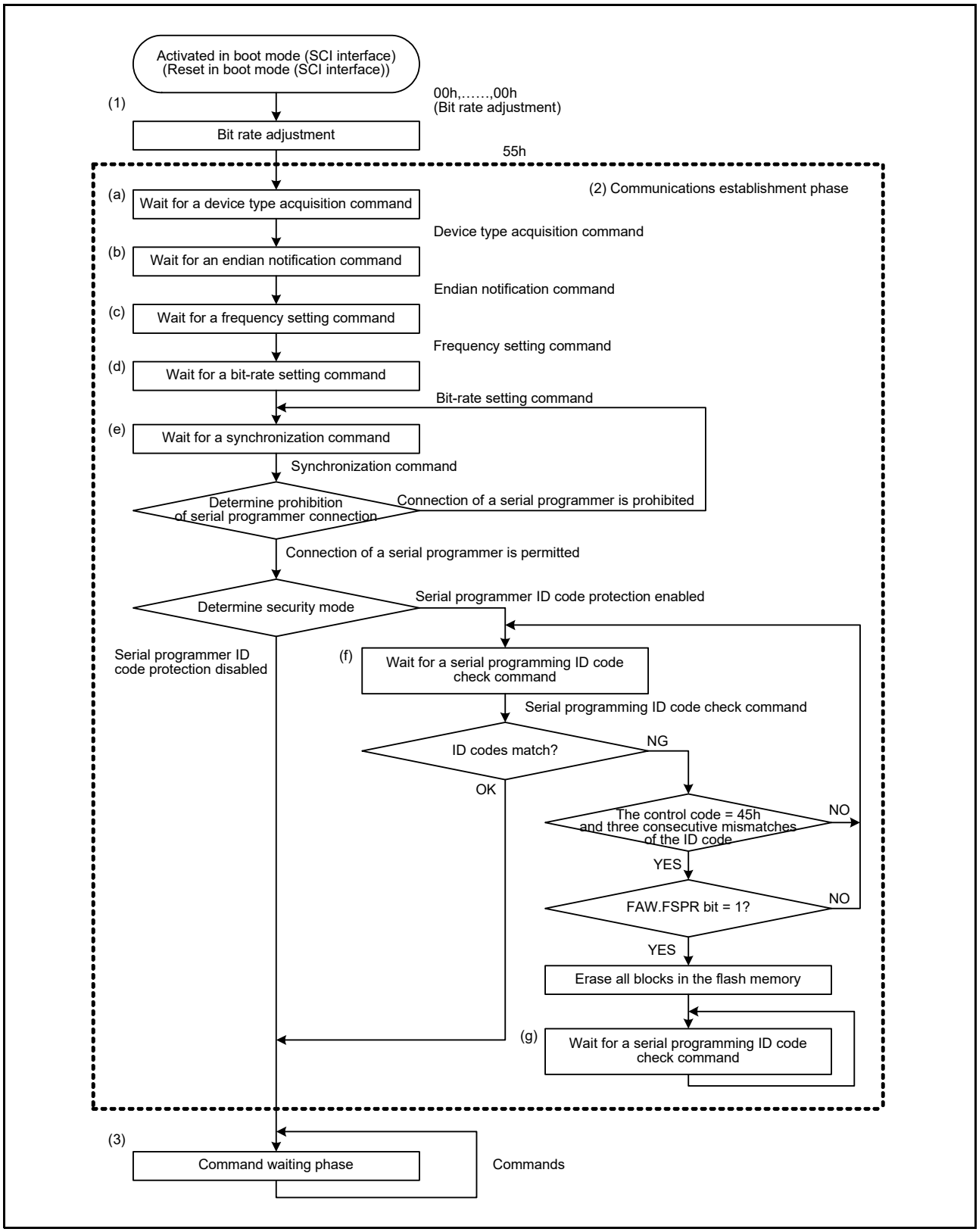


Figure 48.37 Flowchart for Transition to Boot Mode (SCI Interface)

(1) Matching the bit rates

When this MCU is activated in boot mode, the bit rate of the SCI is automatically adjusted to match that of the host. On completion of this automatic bit rate adjustment, this MCU transmits the value 00h to the host. On subsequent correct reception of the value 55h sent from the host, this MCU enters the communications establishment phase. For details on matching of the bit rates, refer to section 48.8.3, Automatic Adjustment of the Bit Rate.

(2) Communications establishment phase

The device, endian, frequency, and bit rate are selected in this phase. If serial programmer ID code protection is enabled, ID code authentication is performed. For the commands to use in the communications establishment phase, refer to section 48.8.5, Communications Establishment Phase.

(a) Waiting for a device type acquisition command

In this state, the MCU is waiting for a device type acquisition command to be sent from the host. When it receives a device type acquisition command, the state moves to waiting for an endian notification command. For details of the device type acquisition command, refer to section 48.8.9, Device Type Acquisition Command.

(b) Waiting for an endian notification command

In this state, the MCU is waiting for an endian notification command to be sent from the host. When it receives an endian notification command, the state moves to waiting for a frequency setting command. For details of the endian notification command, refer to section 48.8.10, Endian Notification Command.

(c) Waiting for a frequency setting command

In this state, the MCU is waiting for a frequency setting command to be sent from the host. When it receives a frequency setting command, the state moves to waiting for a bit-rate setting command. For details of the frequency setting command, refer to section 48.8.11, Frequency Setting Command.

(d) Waiting for a bit-rate setting command

In this state, the MCU is waiting for a bit-rate setting command to be sent from the host. When it receives a bit-rate setting command, the state moves to waiting for a synchronization command. For details of the bit-rate setting command, refer to section 48.8.12, Bit-Rate Setting Command.

(e) Waiting for a synchronization command

In this state, the MCU is waiting for a synchronization command to be sent from the host. When it receives a synchronization command, it determines whether the serial programmer ID code protection is enabled or disabled. When the ID code protection is disabled, the MCU enters the command waiting phase. When the ID code protection is enabled, it enters the state of waiting for a serial programming ID code check command. If the MCU has been set to prohibit the connection of a serial programmer, this MCU transmits an error code to indicate that connecting a serial programmer is prohibited and remains in the state of waiting for a synchronization command. For details of the synchronization command, refer to section 48.8.13, Synchronization Command.

(f) Waiting for a serial programming ID code check command

In this state, the MCU is waiting for a serial programming ID code check command to be sent from the host. The control code and ID codes sent are compared with the control code and ID codes in the OSIS register, and the command waiting phase is entered if they match. If they do not match, the next transition is back to the state of waiting for a serial programming ID code check command. If the control code is 45h and the codes do not match three times in a row, all data in the flash memory are erased*1.
For details of the ID code check command, refer to section 48.8.15, Serial Programming ID Code Check Command.

Note 1. When the FAW.FSPR bit is 0, the data in the flash memory are not erased.

(g) Waiting for a serial programming ID code check command (after erase)

After all blocks in the flash memory are erased, reboot the MCU in boot mode.

(3) Phase of waiting for commands

In this state, program and erase operations proceed according to commands from the host. For details of the commands that can be issued in the command waiting phase, refer to section 48.8.6, Command Waiting Phase.

48.8.3 Automatic Adjustment of the Bit Rate

When this MCU is booted up in boot mode (SCI interface), asynchronous transfer by the SCI is used to measure the periods at low level of consecutive bytes with value 00h that are sent from the host. While the period at low level is being measured, set the host's SCI transfer format to 8-bit data, one stop bit, no parity, and a transfer rate of 9,600 bps. This MCU measures the periods at low level in the signal from the host, adjusts the bit rate of its SCI, and then sends the value 00h to the host.

If reception of the value 00h by the host is successful, the host responds by sending the value 55h to this MCU. If successful reception of 00h by the host is not possible, reboot this MCU in boot mode, and then repeat the process of automatically adjusting the bit rate. If reception of the value 55h by this MCU is successful, it responds by sending C2h to the host, and if successful reception of 55h by this MCU is not possible, it responds by sending FFh to the host.

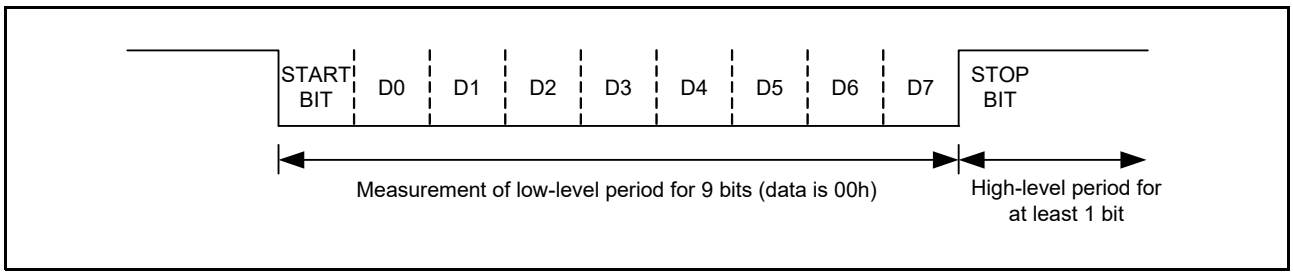


Figure 48.38 Transfer Format Used by SCI in Automatic Adjustment of Bit Rate

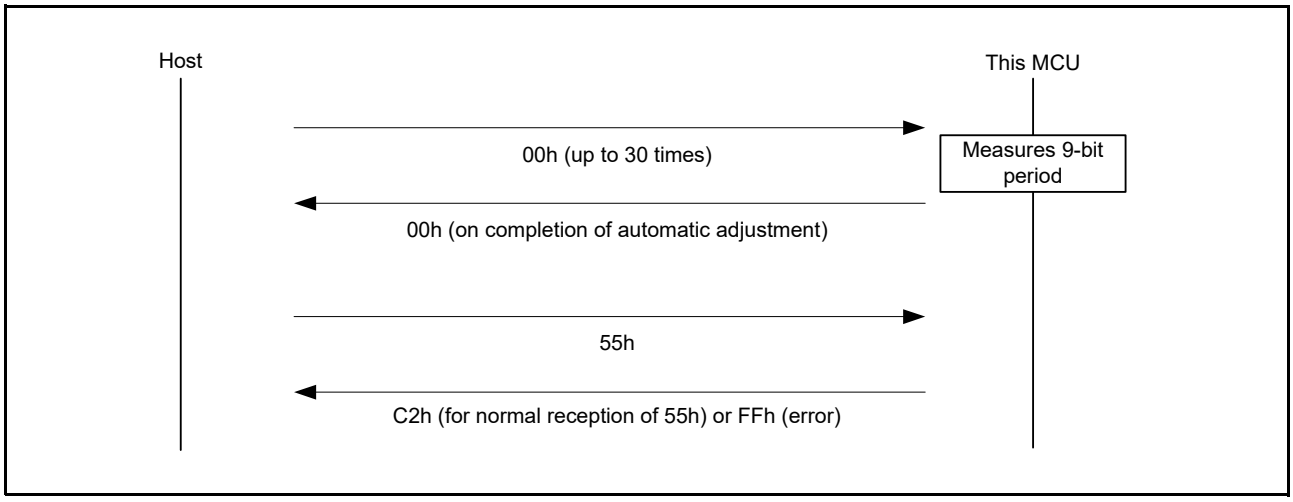


Figure 48.39 Sequence of Transfer between Host and This MCU

For the host's SCI bit rate, ensure that SCI communications proceed under the conditions given in Table 48.20.

Table 48.20 Conditions for Automatic Bit-Rate Adjustment

Bit Rate of SCI in Host
9,600 bps

48.8.4 Packet Format

(1) Command packet

The host sends commands to this MCU in the format below.

SOH	LNH	LNL	COM	Command information (variable length) (up to 255 bytes)	SUM	ETX

Symbol	Code	Description
SOH	01h	Start of a packet (1 byte)
LNH	—	Upper byte of packet length (length of COM and the command information) (1 byte)
LNL	—	Lower byte of packet length (length of COM and the command information) (1 byte)
COM	—	Command code (1 byte)
Command information	—	Command information (up to 255 bytes)
SUM*1	—	Two's complement of the sum of values of LNH, LNL, COM, and the command information (1 byte)
ETX	03h	End of a packet (1 byte)

Note 1. SUM indicates the 1 byte of data that produces 00h as the result of adding LNH, LNL, COM, the command information, and SUM itself.

(2) Status packet and data packet

Data transmission proceeds between the host and this MCU in the format below.

SOD	LNH	LNL	RES	Data (variable length) (up to 1024 bytes)	SUM	ETB	or	ETX

Symbol	Code	Description
SOD	81h	Start of a packet (1 byte)
LNH	—	Upper byte of packet length (length of RES and the data) (1 byte)
LNL	—	Lower byte of packet length (length of RES and the data) (1 byte)
RES	—	Response code (1 byte)
Data	—	Data (up to 1024 bytes)
SUM*1	—	Two's complement of the sum of values of LNH, LNL, RES, and the data (1 byte)
ETB	17h	End of a packet (1 byte)
ETX	03h	End of the last packet (1 byte)

Note 1. SUM indicates the 1 byte of data that produces 00h as the result of adding LNH, LNL, RES, the data, and SUM itself.

48.8.5 Communications Establishment Phase

Table 48.21 lists the commands available in the commands establishment phase.
The synchronization command and ID authentication mode acquisition command can also be used in the command waiting phase.

Table 48.21 Commands Available in the Communications Establishment Phase

Command Name	Function
Device type acquisition	Transmits the oscillation frequency and CPU operating frequency (in Hz) supported by boot mode to the host.
Endian notification	Indicates whether big-endian or little-endian is to be used.
Frequency setting	Sets the values of the oscillation frequency and CPU operating frequency (in Hz).
Bit-rate setting	Changes the bit rate.
Synchronization	This command is used in processing for communications synchronization. It is also used when confirming whether the MCU can accept commands.
ID authentication mode acquisition	This command sends the enabled/disabled state of the serial programmer ID code protection to the host.
Serial programming ID code check	Determines whether the control code and ID code written in the option-setting memory matches the control code and ID code sent by the host.

In the communications establishment phase, send commands from the host in the order of the device type acquisition, endian notification, frequency setting, bit-rate setting, and synchronization commands according to the responses to commands. When serial programmer ID code protection is enabled, send the ID authentication mode acquisition or serial programming ID code check command following the synchronization command.
If commands are issued in an incorrect order or other commands are issued, this MCU returns a response indicating a flow error.

48.8.6 Command Waiting Phase

Table 48.22 lists the commands available in the command waiting phase.
The synchronization command and ID authentication mode acquisition command can also be used in the communications establishment phase.

Table 48.22 Commands Available in the Command Waiting Phase

Command Name	Function
Synchronization	Refer to Table 48.21.
Blank check	Check that a selected area is blank.
Block erase	Erases a selected single block.
Area erase	Erases the specified area.
Program	Programs the selected area.
Read	Reads data from a selected area.
ID authentication mode acquisition	Refer to Table 48.21.
Configuration clear	Erases the configuration setting area and the TM target areas.
Configuration program	Set values to the configuration setting area.
Configuration read	Read values set in the configuration setting area.
Simple addition checksum	Calculates the sum of values in a selected area.
Acquisition of the number of the area information	Obtains the number of area information of the flash memory.
Acquisition of the area information	Obtains the area information of the flash memory.

If the host has sent an undefined command, this MCU returns a response indicating an error in the form of a unsupported command.

48.8.7 Command Transfer Sequence

Though the sequence of transfer differs from command to command, common transfer sequences are used for the commands that only make settings for this MCU and for the commands that obtain information on the settings in this MCU. However, as the contents of the command packet, status packet, and data packet differ for each command, refer to the sections on the individual commands for details.

(1) Common transfer sequence for the commands that only make settings

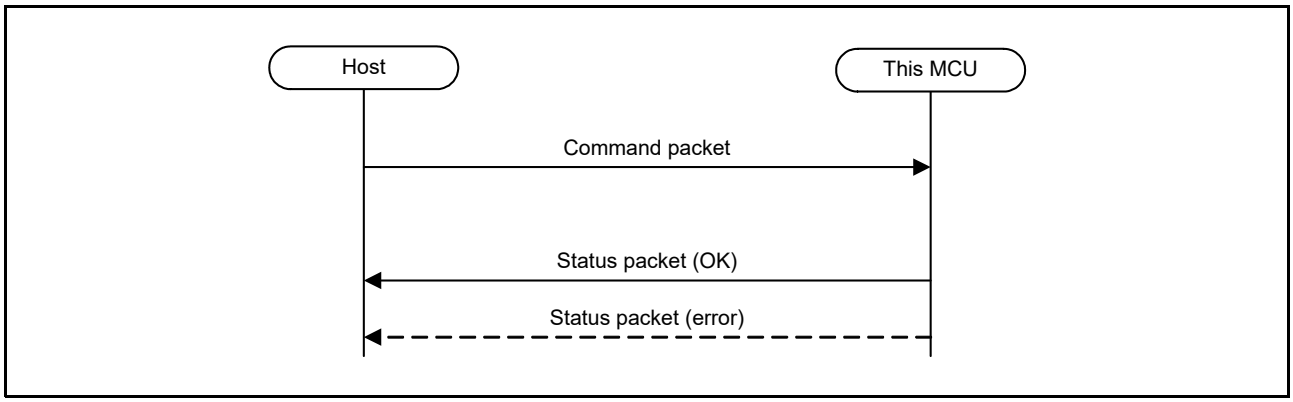


Figure 48.40 Common Transfer Sequence for the Commands that Only Make Settings

(2) Common transfer sequence for the commands that obtain information on settings

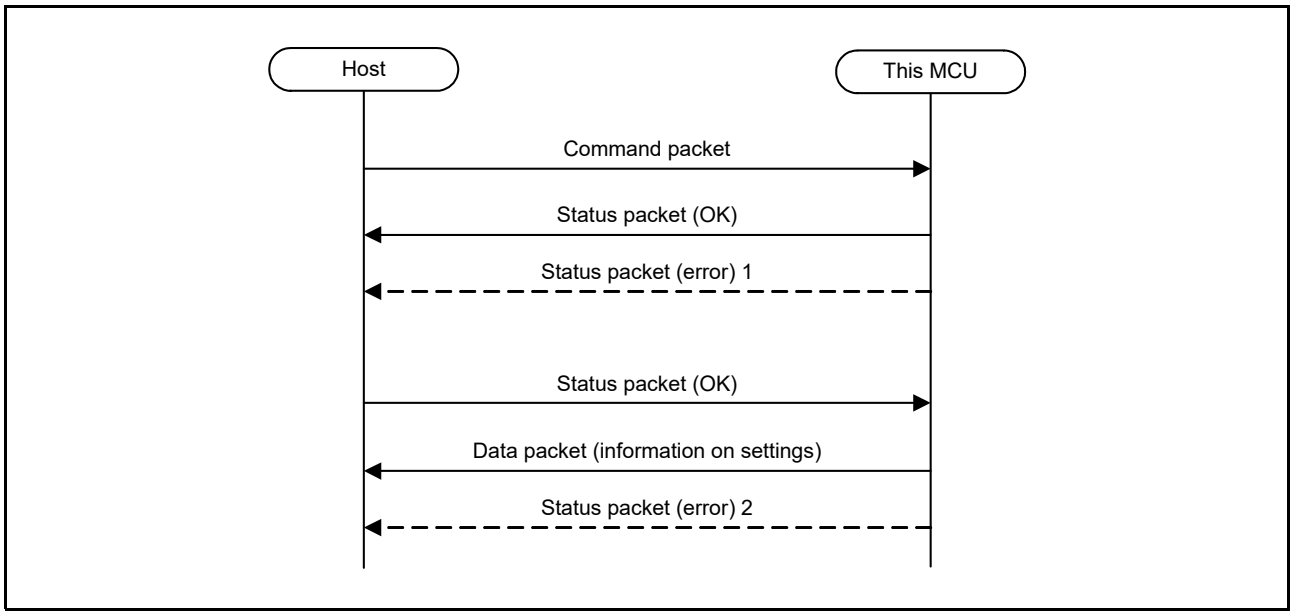


Figure 48.41 Common Transfer Sequence for the Commands that Obtain Information on Settings

Table 48.23 Common Transfer Sequence

Command Name	Common Transfer Sequence Type
Device type acquisition	Command that obtains information on settings
Endian notification	Command that only makes a setting
Frequency setting	Command that obtains information on settings
Bit rate setting	Not in a common transfer sequence
Synchronization	Command that only makes a setting
ID authentication mode acquisition	Command that obtains information on settings
Serial programming ID code check	Command that only makes a setting
Blank check	Command that only makes a setting
Block erase	Command that only makes a setting
Area erase	Command that only makes a setting
Program	Not in a common transfer sequence
Read	Not in a common transfer sequence
Configuration clear	Command that only makes a setting
Configuration program	Not in a common transfer sequence
Configuration read	Command that obtains information on settings
Simple addition checksum	Command that obtains information on settings
Acquisition of the number of the area information	Command that obtains information on settings
Acquisition of the area information	Command that obtains information on settings

For the command transfer sequences that are not in a common transfer sequence, refer to the sections on the individual commands.

48.8.8 Unsupported Commands

When this MCU receives an undefined command packet, it returns an unsupported error (C0h) and enters the command waiting state.

(1) Command packet structure

SOH	LNH	LNL	COM		SUM	ETX

SOH: 01h
LNH: Upper byte of packet length
LNL: Lower byte of packet length
COM: Command code*1
SUM: Sum of values
ETX: 03h

Note 1. Command code other than those specified in Table 48.22.

(2) Status packet structure

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: 80h | COM (command code)
ERR: Error code
C0h (“unsupported” error)
C1h (packet error)
C2h (checksum error)
C3h (flow error)
SUM: Sum of values
ETX: 03h

48.8.9 Device Type Acquisition Command

This command is used to make the MCU send the input frequency and system clock frequency (in Hz) supported by boot mode (SCI interface).
This command can only be accepted in the communications establishment phase.

(1) Command packet structure

SOH	LNH	LNL	COM	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 01h
COM: 38h
SUM: C7h
ETX: 03h

(2) Data packet structure

SOD	LNH	LNL	RES	TYP	OSA	OSI	CPA	CPI	SUM	ETX

The values sent is given below.

Maximum input frequency = 16,000,000 Hz
- OSA (1st byte): 00h
- OSA (2nd byte): F4h
- OSA (3rd byte): 24h
- OSA (4th byte): 00h

Minimum input frequency = 16,000,000 Hz
- OSI (1st byte): 00h
- OSI (2nd byte): F4h
- OSI (3rd byte): 24h
- OSI (4th byte): 00h

Maximum system clock (ICLK) = 120,000,000 Hz
- CPA (1st byte): 07h
- CPA (2nd byte): 27h
- CPA (3rd byte): 0Eh
- CPA (4th byte): 00h

Minimum system clock: (ICLK) = 120,000,000 Hz
- CPI (1st byte): 07h
- CPI (2nd byte): 27h
- CPI (3rd byte): 0Eh
- CPI (4th byte): 00h

SOD: 81h
LNH: 00h
LNL: 19h
RES: 38h (OK)
TYP: Type code (8 bytes)*1
OSA: Maximum input frequency (4 bytes)
OSI: Minimum input frequency (4 bytes)
CPA: Maximum system clock frequency (4 bytes)
CPI: Minimum system clock frequency (4 bytes)
SUM: Sum of values
ETX: 03h

Note 1. Reserved data

(3) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 38h (OK)
SUM: C7h
ETX: 03h

(4) Status packet structure, error occurrence 1

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: B8h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

(5) Status packet structure, error occurrence 2

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: B8h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

48.8.10 Endian Notification Command

This command is used to inform the MCU of the endian (big or little).
Specify either endian as the endian information according to the data to be programmed.
This command can only be accepted in the communications establishment phase.

(1) Command packet structure

SOH	LNH	LNL	COM	END	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 02h
COM: 36h
END: Endian information
00h (big-endian)
01h (little-endian)
SUM: Sum of values
ETX: 03h

(2) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 36h (OK)
SUM: C9h
ETX: 03h

(3) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: B6h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
D7h (endian error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

48.8.11 Frequency Setting Command

This command is used to set the values of oscillation frequency and CPU operating frequency (in Hz).
In boot mode (SCI interface) or boot mode (FINE interface), the HOCO runs at 16 MHz and the ICLK runs at 120 MHz, so set the input frequency to 16 MHz and the system clock frequency to 120 MHz. Furthermore, in boot mode (SCI interface) or boot mode (FINE interface), the FCLK and PCLKB run at 60 MHz, so the MCU sends that the frequency of the peripheral module clock is 60 MHz.
This command can only be accepted in the communications establishment phase.

(1) Command packet structure

SOH	LNH	LNL	COM	OC1	OC2	OC3	OC4	CC1	CC2	CC3	CC4	SUM	ETX

When the input frequency is 16,000,000 Hz and the system clock frequency is 120,000,000 Hz, send the values as below.

OC1: 00h CC1: 07h
OC2: F4h CC2: 27h
OC3: 24h CC3: 0Eh
OC4: 00h CC4: 00h

SOH: 01h
LNH: 00h
LNL: 09h
COM: 32h
OC1: Input frequency
OC2: Input frequency
OC3: Input frequency
OC4: Input frequency
CC1: System clock frequency
CC2: System clock frequency
CC3: System clock frequency
CC4: System clock frequency
SUM: Sum of values
ETX: 03h

(2) Data packet structure

SOD	LNH	LNL	RES	FQ1	FQ2	FQ3	FQ4	PF1	PF2	PF3	PF4	SUM	ETX

The values sent is given below.
System clock frequency = 120,000,000 Hz
Peripheral clock frequency = 60,000,000 Hz

FQ1: 07h PF1: 03h
FQ2: 27h PF2: 93h
FQ3: 0Eh PF3: 87h
FQ4: 00h PF4: 00h

SOD: 81h
LNH: 00h
LNL: 09h
RES: 32h
FQ1: System clock frequency
FQ2: System clock frequency
FQ3: System clock frequency
FQ4: System clock frequency
PF1: Peripheral clock frequency
PF2: Peripheral clock frequency
PF3: Peripheral clock frequency
PF4: Peripheral clock frequency
SUM: Sum of values
ETX: 03h

(3) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 32h (OK)
SUM: CDh
ETX: 03h

(4) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: B2h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
D1h (input frequency error)
D2h (system clock (ICLK) frequency error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

48.8.12 Bit-Rate Setting Command

This command is used to change the bit rate after receiving bit rate data (in bps).
If an error occurs, the bit rate is not changed.
This command can only be accepted in the communications establishment phase.
In boot mode (FINE interface), switching of the bit rate does not proceed. Set the desired value for the bit rate.

(1) Procedure

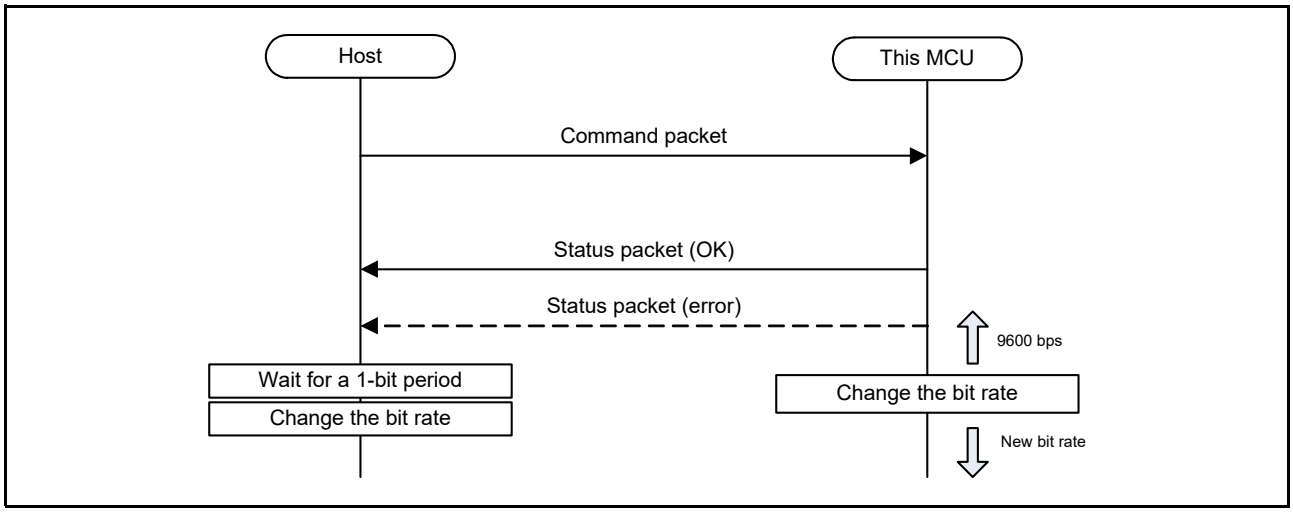


Figure 48.42 Bit-Rate Setting Command Transfer Sequence

(2) Command packet structure

SOH	LNH	LNL	COM	BR1	BR2	BR3	BR4	SUM	ETX

When the bit rate is 2,000,000 bps, send the values as below.

BR1: 00h
BR2: 1Eh
BR3: 84h
BR4: 80h

SOH: 01h
LNH: 00h
LNL: 05h
COM: 34h
BR1: Bit rate
BR2: Bit rate
BR3: Bit rate
BR4: Bit rate
SUM: Sum of values
ETX: 03h

(3) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 34h (OK)
SUM: CBh
ETX: 03h

(4) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: B4h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
D4h (bit rate error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

48.8.13 Synchronization Command

This command is used in processing to synchronize communications.
It is also used when checking whether the MCU is ready to accept commands. If a serial programmer connection is prohibited, a serial programmer connection prohibition error is returned.
This command can be accepted in both the communications establishment and command waiting phases.

(1) Command packet structure

SOH	LNH	LNL	COM	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 01h
COM: 00h
SUM: FFh
ETX: 03h

(2) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 00h (OK)
SUM: FFh
ETX: 03h

(3) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: 80h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
DCh (serial programmer connection prohibition error)
SUM: Sum of values
ETX: 03h

48.8.14 ID Authentication Mode Acquisition Command

This command sends the enabled/disabled state of the serial programmer ID code protection to the host.
This command can be accepted in both the communications establishment and command waiting phases.

(1) Command packet structure

SOH	LNH	LNL	COM	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 01h
COM: 2Ch
SUM: D3h
ETX: 03h

(2) Data packet structure

SOD	LNH	LNL	RES	MOD	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: 2Ch (OK)
MOD: ID authentication information (1 byte)
00h (serial programmer ID code protection is enabled)
FFh (serial programmer ID code protection is disabled)
SUM: Sum of values
ETX: 03h

(3) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 2Ch (OK)
SUM: D3h
ETX: 03h

(4) Status packet structure, error occurrence 1

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: ACh (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

(5) Status packet structure, error occurrence 2

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: ACh (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

48.8.15 Serial Programming ID Code Check Command

This command determines whether the control code and ID codes set in the OSIS register match the control code and ID codes sent by the host, and sends the result to the host.
This command can be accepted in the communications establishment phase. When the serial programmer ID code protection is enabled, the MCU does not enter the command waiting phase unless processing in response to this command ends normally.
If the control code is 45h and the codes do not match three times in a row, all data in the flash memory are erased*1.

Note 1. When the FAW.FSPR bit is 0, data in the flash memory are not erased.

(1) Command packet structure

SOH	LNH	LNL	COM	IDC	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 11h
COM: 30h
IDC: Control code and ID codes (16 bytes)*1
SUM: Sum of values
ETX: 03h

Note 1. Send the values as below.
<ID code>
ID = 0F0E0D0C0B0A09080706050403020100h
(Control code: 00h, ID code 2: 01h, ID code 3: 02h, ... , ID code 16: 0Fh)
<Data for transmission>

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte	…
00h	01h	02h	03h	04h	05h	06h	07h	…

(2) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 30h (OK)
SUM: CFh
ETX: 03h

(3) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: B0h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
DBh (ID code mismatch error)
E1h (erase error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

48.8.16 Blank Check Command

This command checks whether the specified area is blank.
Specify a range in the code flash memory with addresses on 128-byte boundaries or in the data flash memory with addresses on 4-byte boundaries. When the TM function is enabled, attempting a blank check for the area containing the TM target area will result in an error.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	SHH	SHL	SLH	SLL	EHH	EHL	ELH	ELL	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 09h
COM: 10h
SHH: Blank check start address (b24 to b31)
SHL: Blank check start address (b16 to b23)
SLH: Blank check start address (b8 to b15)
SLL: Blank check start address (b0 to b7)
EHH: Blank check end address (b24 to b31)
EHL: Blank check end address (b16 to b23)
ELH: Blank check end address (b8 to b15)
ELL: Blank check end address (b0 to b7)
SUM: Sum of values
ETX: 03h

(2) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 10h (OK)
SUM: EFh
ETX: 03h

(3) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: 90h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
D0h (address error)
E0h (non-blank error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

48.8.17 Block Erase Command

This command is used to erase a specified single block.
Specify the block to erase by the first address of the block.
When the TM function is enabled, the TM target area cannot be erased. To erase the TM target area when the TM function is enabled, use the configuration clear command.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	SHH	SHL	SLH	SLL	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 05h
COM: 12h
SHH: First address of the block to be erased (b24 to b31)
SHL: First address of the block to be erased (b16 to b23)
SLH: First address of the block to be erased (b8 to b15)
SLL: First address of the block to be erased (b0 to b7)
SUM: Sum of values
ETX: 03h

(2) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 12h (OK)
SUM: EDh
ETX: 03h

(3) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: 92h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
D0h (address error)
DAh (protection error)
E1h (erase error)
E7h (flash sequencer error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

48.8.18 Area Erase Command

The area erase command erases the specified area successively, block by block, in ascending order of addresses.
The target area can be selected from the code flash memory or data flash memory.
When the TM function is enabled, only blocks other than the TM target area are erased.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	ARE	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 02h
COM: 50h
ARE: Area
00h (code flash memory)
20h (data flash memory)
SUM: Sum of values
ETX: 03h

(2) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 50h (OK)
SUM: AFh
ETX: 03h

(3) Status packet structure, error occurrence 1

SOD	LNH	LNL	RES	SUM	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: D0h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
D5h (area error)
DAh (protection error)
E1h (erase error)
E7h (flash sequencer error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

48.8.19 Program Command

This command sets up the MCU for data to be programmed to its flash memory and specifies the area where the data are to be programmed.
Specify addresses for a data length that is a multiple of 128 bytes for the code flash memory or a multiple of 4 bytes for the data flash memory. Furthermore, as the address where programming is to start, specify an address on a 128-byte boundary for the code flash memory or on a 4-byte boundary for the data flash memory.
When the TM function is enabled, the TM target area cannot be programmed.
This command can only be accepted in the command waiting phase.

(1) Procedure

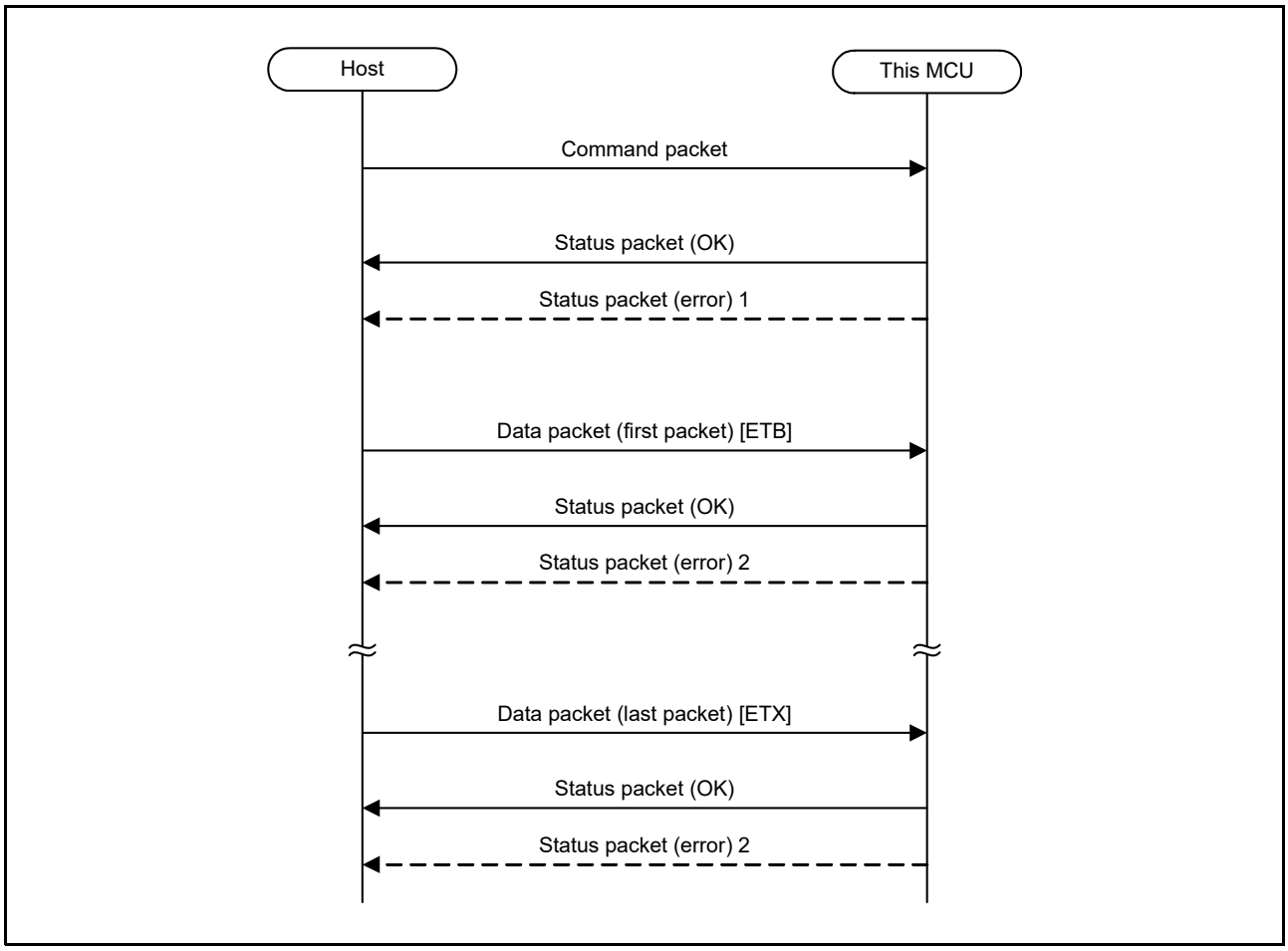


Figure 48.43 Program Command Transfer Sequence

(2) Command packet structure

SOH	LNH	LNL	COM	SHH	SHL	SLH	SLL	EHH	EHL	ELH	ELL	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 09h
COM: 13h
SHH: Program start address (b24 to b31)
SHL: Program start address (b16 to b23)
SLH: Program start address (b8 to b15)
SLL: Program start address (b0 to b7)
EHH: Program end address (b24 to b31)
EHL: Program end address (b16 to b23)
ELH: Program end address (b8 to b15)
ELL: Program end address (b0 to b7)
SUM: Sum of values
ETX: 03h

(3) Data packet structure

SOD	LNH	LNL	RES	Data	SUM	ETB or ETX

SOD: 81h
LNH: Upper byte of “Data length + 1”
LNL: Lower byte of “Data length + 1”
RES: 13h (OK)
Data: Program data
SUM: Sum of values
ETB: 17h
ETX: 03h

(4) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 13h (OK)
SUM: ECh
ETX: 03h

(5) Status packet structure, error occurrence 1

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: 93h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
D0h (address error)
DAh (protection error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

(6) Status packet structure, error occurrence 2

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: 93h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
E2h (program error)
E7h (flash sequencer error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

48.8.20 Read Command

This command is used to read data from the specified area in the flash memory and send it to the host.
The minimum unit for reading is 1 byte. When the TM function is enabled, reading the TM target area returns 00h.
This command can only be accepted in the command waiting phase.

(1) Procedure

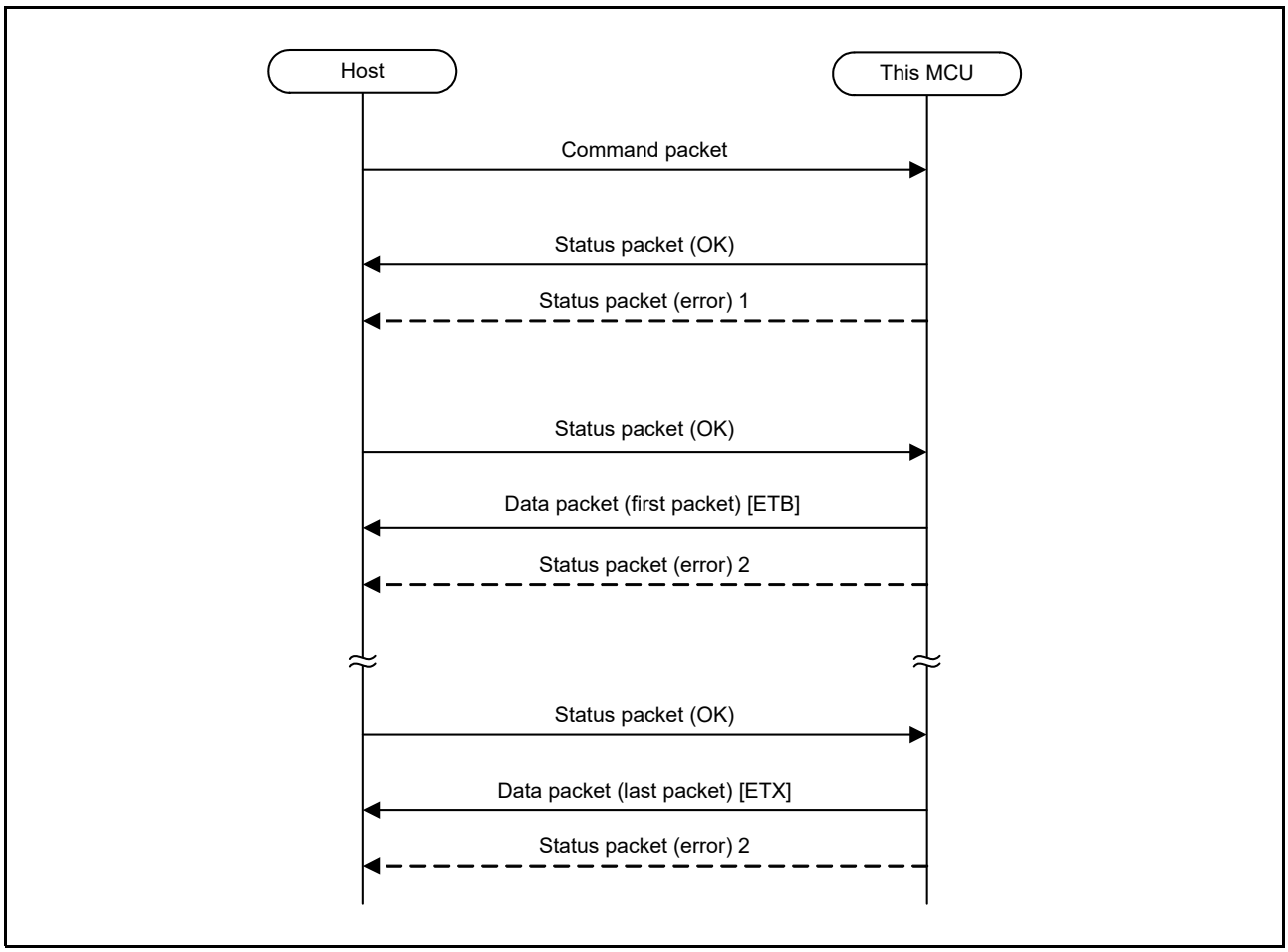


Figure 48.44 Read Command Transfer Sequence

(2) Command packet structure

SOH	LNH	LNL	COM	SHH	SHL	SLH	SLL	EHH	EHL	ELH	ELL	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 09h
COM: 15h
SHH: Read start address (b24 to b31)
SHL: Read start address (b16 to b23)
SLH: Read start address (b8 to b15)
SLL: Read start address (b0 to b7)
EHH: Read end address (b24 to b31)
EHL: Read end address (b16 to b23)
ELH: Read end address (b8 to b15)
ELL: Read end address (b0 to b7)
SUM: Sum of values
ETX: 03h

(3) Data packet structure

SOD	LNH	LNL	RES	Data	SUM	ETB or ETX

SOD: 81h
LNH: Upper byte of “Data length + 1”
LNL: Lower byte of “Data length + 1”
RES: 15h (OK)
Data: Read data
SUM: Sum of values
ETB: 17h
ETX: 03h

(4) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 15h (OK)
SUM: EAh
ETX: 03h

(5) Status packet structure, error occurrence 1

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: 95h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
D0h (address error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

(6) Status packet structure, error occurrence 2

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: 95h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

48.8.21 Configuration Clear Command

This command is used to erase the option-setting memory (configuration setting area) and the TM target area.
This command cannot be issued when the code flash memory and data flash memory are not blank. If an access window has been set, set the TM target area to be located within the access window before sending this command.
When the serial programmer ID code protection is enabled, a reset is required to disable it.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 01h
COM: 1Ch
SUM: E3h
ETX: 03h

(2) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 1Ch (OK)
SUM: E3h
ETX: 03h

(3) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: 9Ch (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
E0h (non-blank error)
E1h (erase error)
E2h (program error)
E7h (flash sequencer error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

48.8.22 Configuration Program Command

This command is used to receive data to be written to the option-setting memory and to program the data in the specified area.
Programming must be specified in the unit acquired by using the area information acquisition command.
This command can only be accepted in the command waiting phase.
Once the FAW.FSPR bit has been enabled, the bit cannot be disabled.
For details of the FAW.FSPR bit, refer to section 7.2.9, Flash Access Window Setting Register (FAW) in section 7, Option-Setting Memory (OFSM).
Rewriting the SPCC.SPE bit while the connection of serial programmer is prohibited leads to normal termination, but programming does not proceed.
In addition, this MCU cannot be connected to a serial programmer at a reset while connection to a serial programmer is disabled.
Writing to the TMEF.TMEF[2:0] bits, TMINF register, and MDE.BANKMD[2:0] bits while the TMEF.TMEF[2:0] bits are 000b in linear mode leads to normal termination, but programming does not proceed.
Writing to the TMEF.TMEF[2:0] bits, TMINF register, and MDE.BANKMD[2:0] bits while the TMEF.TMEF[2:0] bits or the TMEF.TMEFDB[2:0] bits are 000b in dual mode leads to normal termination, but programming does not proceed.

(1) Procedure

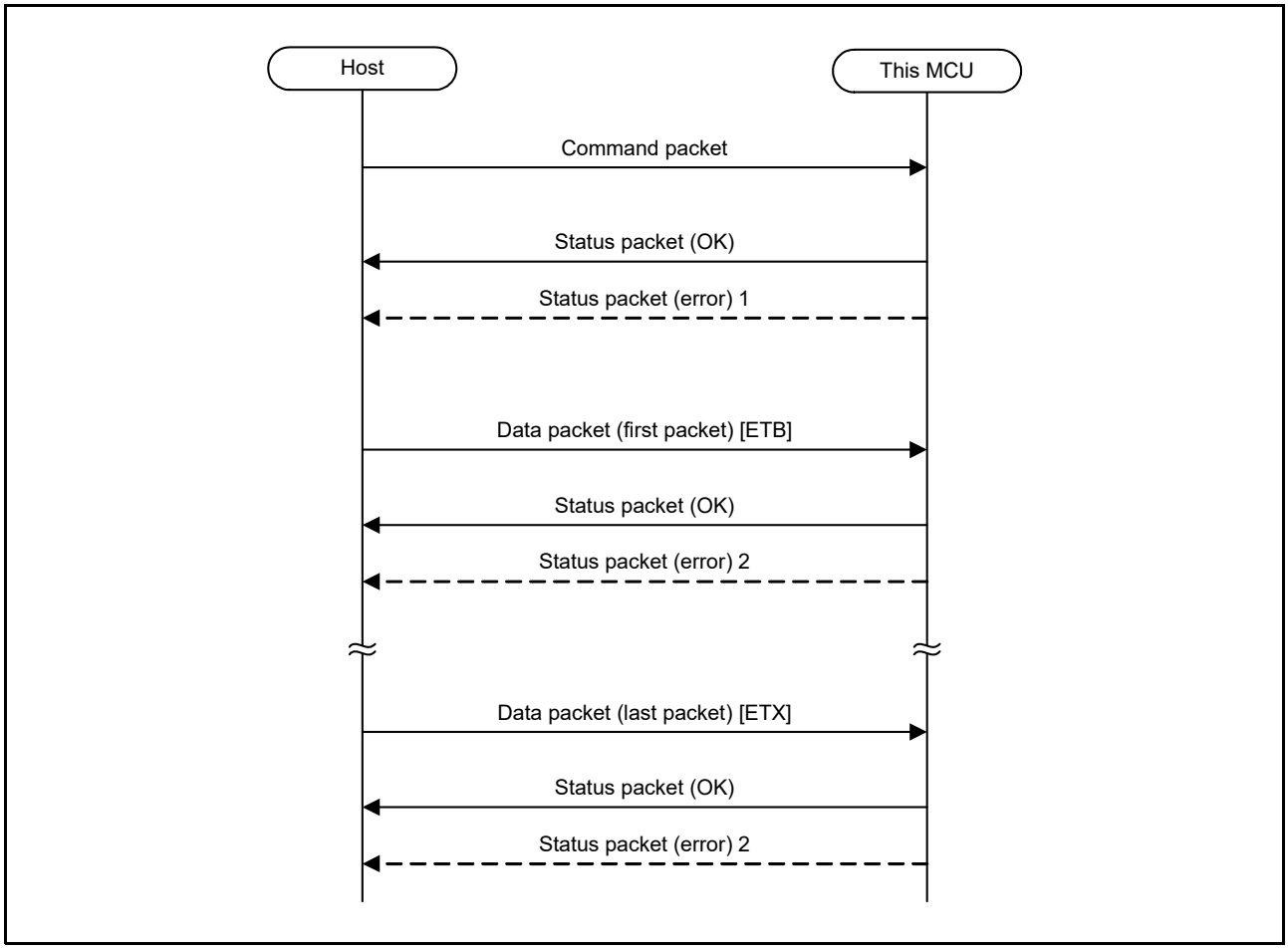


Figure 48.45 Programming Command Transfer Sequence

(2) Command packet structure

SOH	LNH	LNL	COM	SHH	SHL	SLH	SLL	EHH	EHL	ELH	ELL	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 09h
COM: 51h
SHH: Start address (b24 to b31)
SHL: Start address (b16 to b23)
SLH: Start address (b8 to b15)
SLL: Start address (b0 to b7)
EHH: End address (b24 to b31)
EHL: End address (b16 to b23)
ELH: End address (b8 to b15)
ELL: End address (b0 to b7)
SUM: Sum of values
ETX: 03h

(3) Data packet structure

SOD	LNH	LNL	RES	Data	SUM	ETB or ETX

SOD: 81h
LNH: Upper byte of “Data length + 1”
LNL: Lower byte of “Data length + 1”
RES: 51h (OK)
Data: Program data*1
SUM: Sum of values
ETB: 17h
ETX: 03h

Note 1. The data for programming must be sent from the host in order from that for the lowest address to that for the highest address.

<Write data>

Address	0h	1h	2h	3h	4h	5h	6h	7h	...
Data	00h	01h	02h	03h	04h	05h	06h	07h	...

<Program data>

1st byte	**2nd byte**	**3rd byte**	**4th byte**	**5th byte**	**6th byte**	**7th byte**	**8th byte**	**...**
00h	01h	02h	03h	04h	05h	06h	07h	...

(4) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 51h (OK)
SUM: AEh
ETX: 03h

(5) Status packet structure, error occurrence 1

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: D1h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
D0h (address error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

(6) Status packet structure, error occurrence 2

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: D1h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
E1h (erase error)
E2h (program error)
E7h (flash sequencer error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

48.8.23 Configuration Read Command

This command is used to read data from the area specified by the option-setting memory and to transmit the data to the host.

The minimum unit for reading is 4 bytes.

This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	SHH	SHL	SLH	SLL	EHH	EHL	ELH	ELL	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 09h
COM: 52h
SHH: Start address (b24 to b31)
SHL: Start address (b16 to b23)
SLH: Start address (b8 to b15)
SLL: Start address (b0 to b7)
EHH: End address (b24 to b31)
EHL: End address (b16 to b23)
ELH: End address (b8 to b15)
ELL: End address (b0 to b7)
SUM: Sum of values
ETX: 03h

(2) Data packet structure

SOD	LNH	LNL	RES	Data	SUM	ETB or ETX

SOD: 81h
LNH: Upper byte of “Data length + 1”
LNL: Lower byte of “Data length + 1”
RES: 52h (OK)
Data: Read data*1
SUM: Sum of values
ETB: 17h
ETX: 03h

Note 1. The data is read in the order from the lower address to the higher address.

<Program data>

Address	0h	1h	2h	3h	4h	5h	6h	7h	...
Data	00h	01h	02h	03h	04h	05h	06h	07h	...

<Read data>

1st byte	**2nd byte**	**3rd byte**	**4th byte**	**5th byte**	**6th byte**	**7th byte**	**8th byte**	**...**
00h	01h	02h	03h	04h	05h	06h	07h	...

(3) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 52h (OK)
SUM: ADh
ETX: 03h

(4) Status packet structure, error occurrence 1

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: D2h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
D0h (address error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

(5) Status packet structure, error occurrence 2

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: D2h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

48.8.24 Simple Addition Checksum Command

This command is used to calculate the sum of values in the specified area and send the result to the host. While the TM function is enabled, however, the values in the TM target areas are not included in the calculation.
The target area of this command can be selected from the code flash memory and data flash memory. Calculation is by simple addition. The initial value is 0 and the sum of values in the specified area is obtained by adding the values of all bytes.
When code flash memory is specified as the target area, all 512 Kbytes, which is the maximum capacity of this MCU group, is added.
When data flash memory is specified as the target area, the result becomes undefined if an erased area exists. When a simple addition checksum is to be executed for the data flash memory, make sure that data have been written throughout the area.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	ARE	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 02h
COM: 4Dh
ARE: Area information
00h (code flash memory)
20h (data flash memory)
SUM: Sum of values
ETX: 03h

(2) Data packet structure

SOD	LNH	LNL	RES	SD1	SD2	SD3	SD4	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 05h
RES: 4Dh (OK)
SD1: Sum of values
SD2: Sum of values
SD3: Sum of values
SD4: Sum of values
If the sum of values is 01234567h, the settings are as follows.
SD1 = 01h
SD2 = 23h
SD3 = 45h
SD4 = 67h
SUM: Sum of values
ETX: 03h

(3) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 4Dh (OK)
SUM: B2h
ETX: 03h

(4) Status packet structure, error occurrence 1

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: CDh (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
D5h (area error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

(5) Status packet structure, error occurrence 2

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: CDh (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

48.8.25 Acquisition of the Number of the Area Information Command

This command is used to transmit the number of area that the MCU has to the host.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 01h
COM: 53h
SUM: ACh
ETX: 03h

(2) Data packet structure

SOD	LNH	LNL	RES	NOA	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: 53h (OK)
NOA: Number of area information (1 byte)
05h (in linear mode)
08h (in dual mode)
SUM: Sum of values
ETX: 03h

(3) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 53h (OK)
SUM: ACh
ETX: 03h

(4) Status packet structure, error occurrence 1

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: D3h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

(5) Status packet structure, error occurrence 2

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: D3h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

48.8.26 Area Information Acquisition Command

This command is used to transmit information of the specified area to the host.
The address of the erase command, programming command, or configuration program command must be specified according to the range and unit (alignment) returned by this command.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	NUM	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 02h
COM: 54h
NUM: Area number [0 to NOA-1]
SUM: C5h
ETX: 03h

(2) Data packet structure

SOD	LNH	LNL	RES	KOA	SAD	EAD	EAU	WAU	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 12h
RES: 54h (OK)
KOA: Kind of the area
00h (code flash memory)
20h (data flash memory)
30h (option-setting memory)
40h (Trusted Memory disabled)
41h (Trusted Memory enabled)
SAD: Start address (4 bytes)
EAD: End address (4 bytes)
EAU: Erase access unit (alignment) [Byte] (4 bytes)
WAU: Write access unit (alignment) [Byte] (4 bytes)
SUM: Sum of values
ETX: 03h

Data packet in linear mode

NUM	KOA	SAD	EAD	EAU	WAU	Description
00h	00h	FFFF 8000h	FFFF FFFFh	0000 1000h	0000 0080h	Code flash memory blocks 0 to 7 (4-Kbyte block)
01h	00h	FFF8 0000h (512KB) FFFC 0000h (256KB)	FFFF 7FFFh	0000 4000h	0000 0080h	Code flash memory blocks 8 to 37 (512KB) blocks 8 to 21 (256KB) (16-Kbyte block)
02h	20h	0010 0000h	0010 3FFFh	0000 0040h	0000 0004h	Data flash memory
03h	30h	0012 0040h	0012 00FFh	0000 00C0h	0000 0010h	Option-setting memory
04h	40h	FFFF 0000h	FFFF 7FFFh	0000 4000h	0000 0080h	Trusted Memory blocks 8 and 9 disabled
	41h	FFFF 0000h	FFFF 7FFFh	0000 4000h	0000 0080h	Trusted Memory blocks 8 and 9 enabled

Settings of the TMEF.TMEF[2:0] bits and TMEF.TMEFDB[2:0] bits and setting of KOA while NUM is 04h in linear mode

TMEF.TMEF[2:0] bits	TMEF.TMEFDB[2:0] bits	KOA when NUM is 04h
111b	Don't care	40h
000b	Don't care	41h

Data packet in dual mode

NUM	KOA	SAD	EAD	EAU	WAU	Description
00h	00h	FFFF 8000h	FFFF FFFFh	0000 1000h	0000 0080h	Code flash memory blocks 0 to 7 (4-Kbyte block)
01h	00h	FFFC 0000h	FFFF 7FFFh	0000 4000h	0000 0080h	Code flash memory blocks 8 to 21 (16-Kbyte block)
02h	00h	FFFB 8000h	FFFB FFFFh	0000 1000h	0000 0080h	Code flash memory blocks 22 to 29 (4-Kbyte block)
03h	00h	FFF8 0000h	FFFB 7FFFh	0000 4000h	0000 0080h	Code flash memory blocks 30 to 43 (16-Kbyte block)
04h	20h	0010 0000h	0010 3FFFh	0000 0040h	0000 0004h	Data flash memory
05h	30h	0012 0040h	0012 00FFh	0000 00C0h	0000 0010h	Option-setting memory
06h	40h	FFFF 0000h	FFFF 7FFFh	0000 4000h	0000 0080h	Trusted Memory blocks 8 and 9 disabled
	41h	FFFF 0000h	FFFF 7FFFh	0000 4000h	0000 0080h	Trusted Memory blocks 8 and 9 enabled
07h	40h	FFFB 0000h	FFFB 7FFFh	0000 4000h	0000 0080h	Trusted Memory blocks 30 and 31 disabled
	41h	FFFB 0000h	FFFB 7FFFh	0000 4000h	0000 0080h	Trusted Memory blocks 30 and 31 enabled

Settings of the TMEF.TMEF[2:0] bits and TMEF.TMEFDB[2:0] bits and setting of KOA while NUM is 06h or 07h in dual mode

TMEF.TMEF[2:0] bits	TMEF.TMEFDB[2:0] bits	KOA when NUM is 06h or 07h
111b	111b	40h
000b	000b	41h

(3) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 54h (OK)
SUM: ABh
ETX: 03h

(4) Status packet structure, error occurrence 1

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: D4h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

(5) Status packet structure, error occurrence 2

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: D4h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

48.8.27 Usage Example

Figure 48.46 shows an example of the procedure for reprogramming.

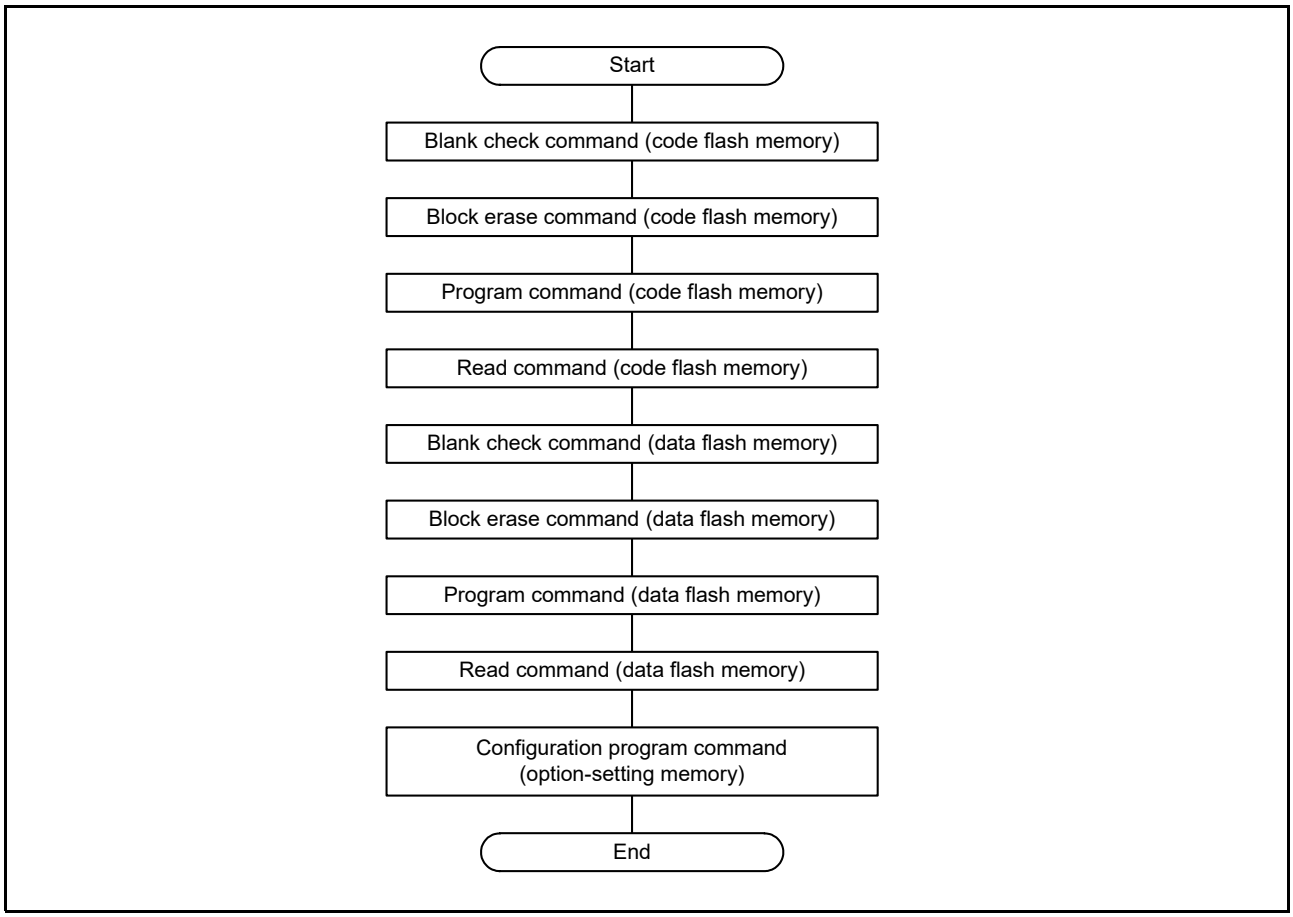


Figure 48.46 Example of Reprogramming Procedure

48.8.28 Rewriting Flash Memory When Using Dual Mode

The flash memory is erased (blank) as shipped by Renesas, so the setting of the MDE.BANKMD[2:0] bits is 111b (linear mode). To use the flash memory in dual mode, set the MDE.BANKMD[2:0] bits to 000b (dual mode).

(1) Rewriting in Dual Mode

Figure 48.47 shows an example of rewriting flash memory after switching to dual mode. The user program can be written to bank 0, bank 1, or both.

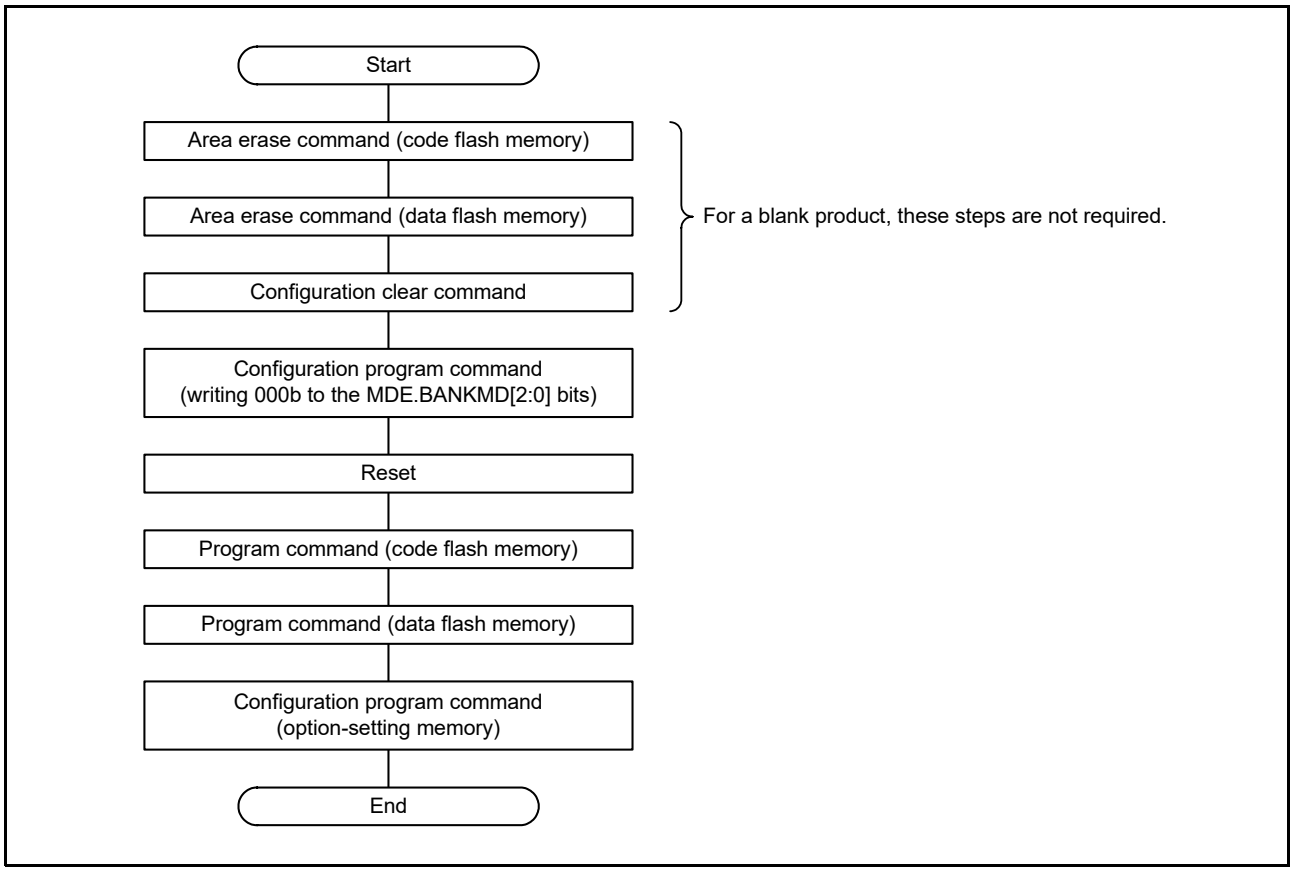


Figure 48.47 Example of Rewriting Flash Memory after Switching to Dual Mode

(2) Rewriting in Linear Mode

Figure 48.48 shows an example of rewriting flash memory in linear mode. Figure 48.49 shows the memory map before and after a reset.

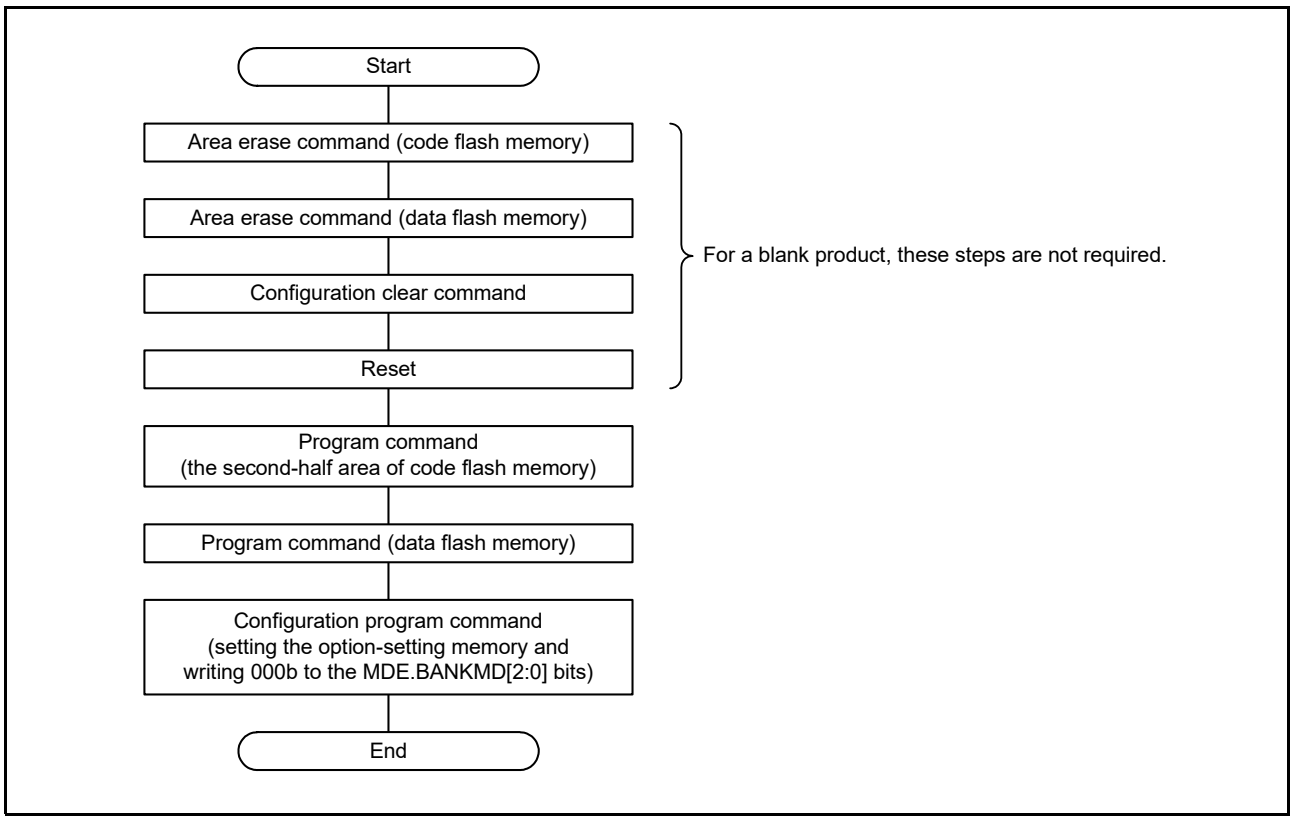


Figure 48.48 Example of Rewriting Flash Memory in Linear Mode

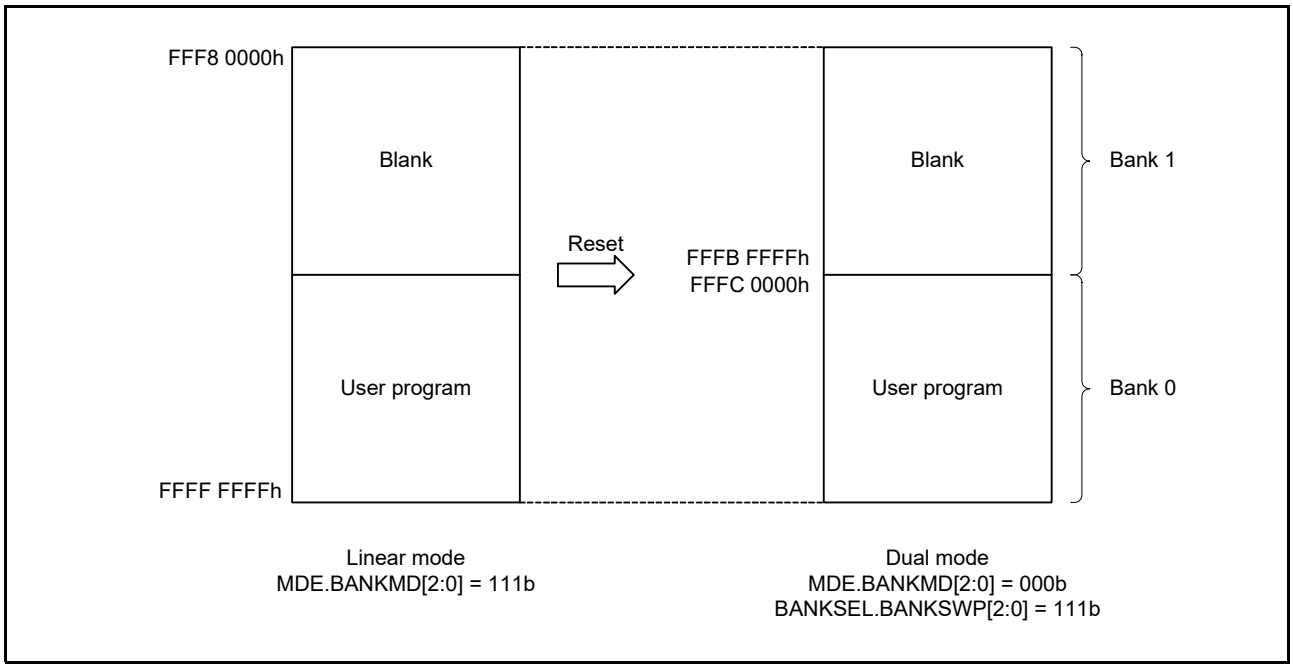


Figure 48.49 Memory Map before and after a Reset

48.9 Using the Serial Programmer for Rewriting (Serial Programming)

By providing a connector on the board, it is possible to program the MCU with a serial programmer while the MCU is mounted on the board.

48.9.1 Environments for Serial Programming

The recommended environments for rewriting the flash memory of the MCU are described below.

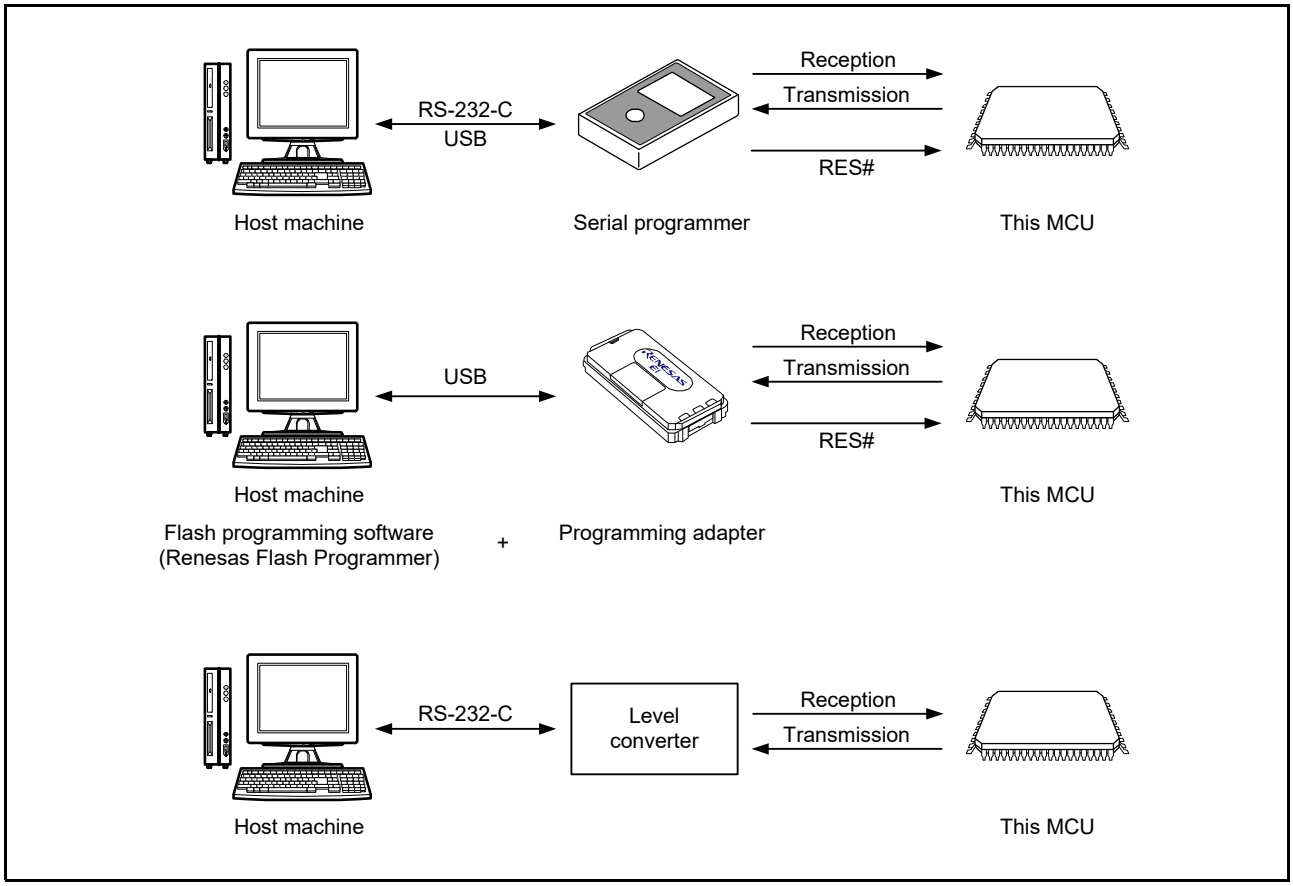


Figure 48.50 Environments for Rewriting the Flash Memory

Note: For details of the serial programmers, refer to the manual of each serial programmer; for details of the Renesas Flash Programmer flash programming software, refer to the Renesas Flash Programmer Flash Programming Software User's Manual.

48.10 Programming through Self-Programming

48.10.1 Overview

This MCU supports programming of the flash memory by the user program itself. The FACI commands can be used with user programs for writing to the flash memory. This allows upgrading of user programs and rewriting of constant data fields.

When the data flash memory is programmed, the background operation facility makes it possible to execute a programming program from the code flash memory to program the data flash memory. Furthermore, the programming program can be copied to internal RAM in advance of the program operation, and executed from the given destination to program the data flash memory.

Background operation is also available for use when the address ranges of the area of code flash memory to be programmed and the area of code flash memory to be read satisfy particular conditions (see Table 48.24). At the time of self-programming in this case, a programming program in code flash memory can be used to program the code flash memory. Also, the programming program can be copied to internal RAM in advance of the program operation, and executed from the given destination to program the code flash memory. This is useful when the address ranges do not satisfy the conditions for background operation.

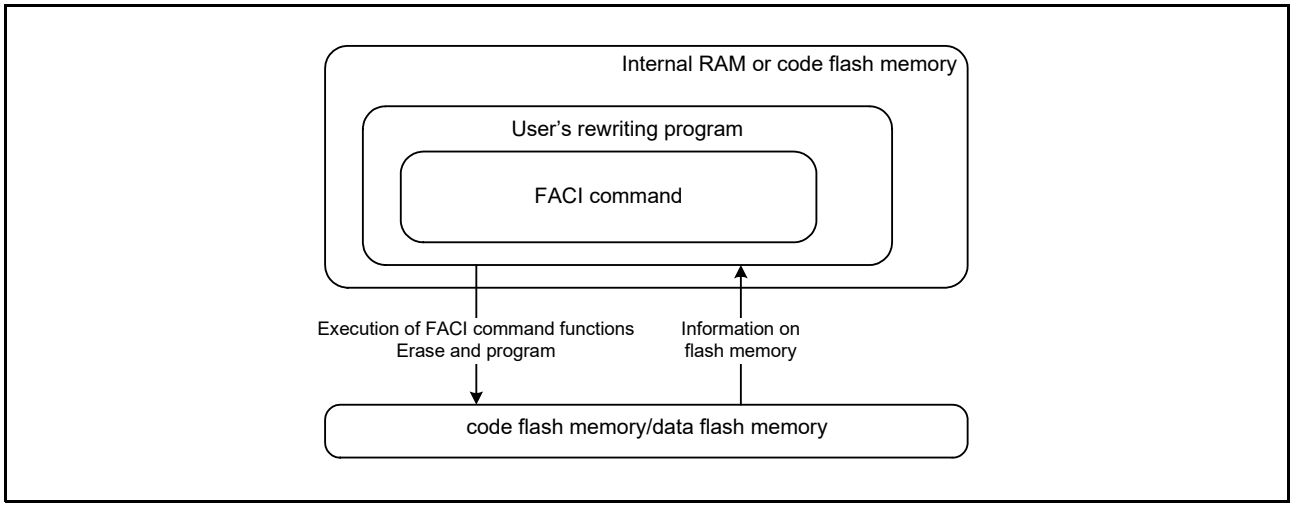


Figure 48.51 Schematic View of Self-Programming

48.10.2 Background Operation

The BGO is can be used to execute the flash rewrite routine on the code flash memory when the data flash memory or other area of code flash memory is rewritten.
Background operations can be used when the combination of the flash memory for rewriting and the flash memory for reading is any of those listed below.

Table 48.24 Conditions under which Background Operation is Usable

		Range for Rewriting	Range for Reading
Common to linear and dual modes		• Data flash memory	• Code flash memory
Linear mode	Products with 512 Kbytes of code flash memory	• First half (256-Kbyte area) of the code flash memory (addresses FFF8 0000h to FFFB FFFFh)	• Second half (256-Kbyte area) of the code flash memory (addresses FFFC 0000h to FFFF FFFFh) • Data flash memory
		• Second half (256-Kbyte area) of the code flash memory (addresses FFFC 0000h to FFFF FFFFh)	• First half (256-Kbyte area) of the code flash memory (addresses FFF8 0000h to FFFB FFFFh) • Data flash memory
Dual mode	When the BANKSEL.BANKSWP[2:0] bits are 111b:	• Bank 1 area of the code flash memory	• Bank 0 area of the code flash memory • Data flash memory
	When the BANKSEL.BANKSWP[2:0] bits are 000b:	• Bank 0 area of the code flash memory	• Bank 1 area of the code flash memory • Data flash memory

48.11 Usage Notes

(1) Reading Area Where Program or Erase Operation was Aborted or Suspended

The data stored in the area where program or erase operation has been aborted or suspended are undefined. To avoid malfunctions caused by reading undefined data, take care not to fetch instructions or read data from the area where program or erase operation was aborted or suspended.

(2) Suspension During Program/Erase Operation

When program or erase operation is interrupted by issuing the P/E suspend command, the program or erase operation can be resumed by issuing the P/E resume command. After the suspend processing is successfully completed and the ERSSPD or PRGSPD flag is set to 1, if the flash sequencer enters the command-locked state for some reason and a forced stop command is issued, the suspended operation cannot be resumed. Also, since the values in the area where the operation was suspended are not guaranteed, erase that area.

(3) Prohibition of Additional Programming

Programming to the code flash memory or data flash memory twice is not possible. To program code flash memory or data flash memory which has already been programmed, erase the relevant area.

(4) Resets During Program, Erase, or Blank Check Operation

In the case of a reset due to the signal on the RES# pin during program, erase, or blank check operation, wait for at least t_{RESWF} (refer to section 49, Electrical Characteristics) of the reset input period once the operating voltage is within the range stipulated in the electrical characteristics after assertion of the reset signal, and then release the device from the reset state.

(5) Allocation of Vectors for Interrupts and Other Exceptions During Program/Erase Operation

Generation of an interrupt or other exception during program or erase operation may lead to fetching of the vector from the code flash memory. If the conditions for using the background operation (BGO) are not satisfied, set the address for vector to an address that is not in the code flash memory or disable interrupts.

(6) Abnormal Termination During Program, Erase, or Blank Check Operation

Even if program or erase operation ends abnormally due to the generation of a reset by the RES# pin, the program or erase state of the flash memory with undefined data cannot be verified or checked. For the area where program or erase operation ends abnormally, the blank check function cannot judge whether the area is erased successfully or not. Erase the area again and confirm that the corresponding area is completely erased before using.

(7) Items Prohibited During Program, Erase, or Blank Check Operation

High voltage is applied to the flash memory during program, erase, or blank check operation. To prevent damage to the flash memory, do not perform the following operations.

- Have the operating voltage from the power supply go beyond the permitted range.
- Change the FWEPROR.FLWE[1:0] bits.
- Change the SCKCR.FCK[3:0] and PCLKB[3:0] bits.
- Change the SCKCR3.CKSEL[2:0] bits.
- Change the RSTCKCR.RSTCKEN bit.
- Transition to the all module clock stop mode or software standby mode.

49. Electrical Characteristics

49.1 Absolute Maximum Ratings

Table 49.1 Absolute Maximum Rating

Conditions: VSS = AVSS0 = AVSS1 = AVSS2 = 0V

Item		Symbol	Value	Unit
Power supply voltage*1		VCC	–0.3 to +6.5	V
Analog power supply voltage*1		AVCC0, AVCC1, AVCC2	–0.3 to +6.5	V
Input voltage	PB1 and PB2	V_{in}	–0.3 to +6.5	V
	P40 to P47, P50 to P55, and P60 to P65		–0.3 to AVCC2 + 0.3 (up to 6.5)	
	Other than above		–0.3 to VCC + 0.3 (up to 6.5)	
Junction temperature		T_j	–40 to +125	°C
Storage temperature		T_{stg}	–55 to +125	°C

Caution: Permanent damage to the LSI may result if absolute maximum ratings are exceeded.

Note 1. Insert capacitors with good frequency characteristics between each power supply pin and the ground. Specifically, place capacitors with a value around 0.1 µF as close as possible to every power supply pin, and use the shortest and thickest possible traces.

49.2 Recommended operating conditions

Table 49.2 Recommended operating conditions (1)

Item		Symbol	Min.	Typ.	Max.	Unit
Power supply voltage		VCC*1	2.7	—	5.5	V
		VSS	—	0	—	
Analog power supply voltage*2		AVCC0, AVCC1, AVCC2*1	3.0	—	5.5	V
		AVSS0, AVSS1, AVSS2	—	0	—	
Input voltage	PB1, PB2	V_{in}	–0.3	—	5.8	V
	P40 to P47, P50 to P55, and P60 to P65		–0.3	—	AVCC2 + 0.3	
	Other than above		–0.3	—	VCC + 0.3	
Operating temperature	D version	T_{opr}	–40	—	85	°C
	G version		–40	—	105	
Junction temperature	D version	T_j	–40	—	105	°C
	G version		–40	—	125	

Note 1. Comply with the following voltage condition: VCC ≤ AVCC0 = AVCC1 = AVCC2

Note 2. When not using any of the12-bit A/D converter (unit 0 to 2), 12-bit D/A converter, comparator C, or temperature sensor, connect AVCC0, AVCC1, and AVCC2 to VCC, and AVSS0, AVSS1, and AVSS2 to VSS, respectively. For details, refer to section 42.6.9, Voltage Range of Analog Power Supply Pins.

Table 49.3 Recommended operating conditions (2)

Item	Symbol	Value
Decoupling capacitance to stabilize the internal voltage	C_{VCL}	0.47 µF ± 30%*1

Note 1. Use a multilayer ceramic capacitor whose nominal capacitance is 0.47 µF and a capacitance tolerance is ±30% or better.

49.3 DC Characteristics

Table 49.4 DC Characteristics (1)

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = AVCC1 = AVCC2 = 3.0 to 5.5 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V,
$T_a = T_{opr}$

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Schmitt trigger input voltage	CANFD input pin MTU input pin GPTW input pin POE input pin POEG input pin TMR input pin SCI input pin RSCI input pin ADTRG# input pin RES#, NMI	V_{IH}	0.8 × VCC	—	—	V	
		V_{IL}	—	—	0.2 × VCC		
		ΔV_T	0.06 × VCC	—	—		
	IRQ input pin (except for P52 to P55, and P60 to P65)	V_{IH}	0.8 × VCC	—	—		
		V_{IL}	—	—	0.2 × VCC		
		ΔV_T	0.06 × VCC	—	—		
	IRQ input pin (P52 to P55, and P60 to P65)	V_{IH}	0.8 × AVCC2	—	—		
		V_{IL}	—	—	0.2 × AVCC2		
		ΔV_T	0.06 × AVCC2	—	—		
	RIIC input pin (except for SMBus)	V_{IH}	0.7 × VCC	—	—		
		V_{IL}	—	—	0.3 × VCC		
		ΔV_T	0.06 × VCC	—	—		
	RI3C input pin	V_{IH}	0.7 × VCC	—	—		VCC = 2.7 to 3.6 V
		V_{IL}	—	—	0.3 × VCC		
		ΔV_T	0.1 × VCC	—	—		
	Pins for 5 V tolerant (PB1 and PB2)	V_{IH}	0.8 × VCC	—	—		
		V_{IL}	—	—	0.2 × VCC		
	Analog input pins (P40 to P47, P50 to P55, and P60 to P63)	V_{IH}	0.8 × AVCC2	—	—		
		V_{IL}	—	—	0.2 × AVCC2		
	Other input pins (pins other than those above)	V_{IH}	0.8 × VCC	—	—		
		V_{IL}	—	—	0.2 × VCC		
High-level input voltage (except for Schmitt trigger input pin)	MD pin, EMLE	V_{IH}	0.9 × VCC	—	—	V	
	EXTAL, RSPI input pin, RSPIA input pin		0.8 × VCC	—	—		
	RIIC (SMBus)		2.1	—	—		
Low-level input voltage (except for Schmitt trigger input pin)	MD pin, EMLE	V_{IL}	—	—	0.1 × VCC	V	
	EXTAL, RSPI input pin, RSPIA input pin		—	—	0.2 × VCC		
	RIIC (SMBus)		—	—	0.8		

Table 49.5 DC Characteristics (2)

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = AVCC1 = AVCC2 = 3.0 to 5.5 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V,
$T_a = T_{opr}$

Item			Symbol	Min.	Typ.	Max.	Unit	Test Conditions
High-level output voltage	P40 to P47, P50 to P55, and P60 to P65		V_{OH}	AVCC2 – 0.5	—	—	V	I_{OH} = –1.0 mA
	P90 to P95, P71 to P76, P81, PB5, and PD3			VCC – 1.0	—	—		I_{OH} = –5.0 mA VCC < 4.0 V (when the large current output is set)
				VCC – 1.1	—	—		I_{OH} = –15.0 mA VCC ≥ 4.0 V (when the large current output is set)
	RI3C pins			VCC – 0.27	—	—		I_{OH} = –3.0 mA
	Other than above	Normal drive		VCC – 0.5	—	—		I_{OH} = –1.0 mA
		High drive		VCC – 0.5	—	—		I_{OH} = –2.0 mA
Low-level output voltage	P40 to P47, P50 to P55, and P60 to P65		V_{OL}	—	—	0.5	V	I_{OL} = 1.0 mA
	P90 to P95, P71 to P76, P81, PB5, and PD3			—	—	1.0		I_{OL} = 15.0 mA (when the large current output is set)
	RIIC pins			—	—	0.4		I_{OL} = 3.0 mA
				—	—	0.6		I_{OL} = 6.0 mA
	RI3C pins			—	—	0.27		I_{OL} = 3.0 mA
	Other than above	Normal drive		—	—	0.5		I_{OL} = 1.0 mA
		High drive		—	—	0.5		I_{OL} = 2.0 mA
Input leakage current	RES#, MD pin, PE2, and EMLE*1		$\lvert I_{in} \rvert$	—	—	1.0	μA	V_{in} = 0 V V_{in} = VCC
Three-state leakage current (off state)	RIIC pins		$\lvert I_{TSI} \rvert$	—	—	5.0	μA	V_{in} = 0 V V_{in} = VCC
	Other than above			—	—	1.0		
Input pull-up resistors	P40 to P47, P50 to P55, and P60 to P65		R_{PU}	10	—	100	kΩ	AVCC2 = 3.0 to 5.5 V V_{in} = 0 V
	Pins other than those above and PE2			10	—	100		VCC = 2.7 to 5.5 V V_{in} = 0 V
Input pull-down resistors	EMLE		R_{PD}	10	—	100	kΩ	V_{in} = VCC = AVCC
Input capacitance	RIIC pins		C_{in}	—	—	16	pF	V_{bias} = 0 V V_{amp} = 20 mV f = 1 MHz T_a = 25°C
	Other than above			—	—	8		
Output voltage of the VCL pin			V_{CL}	—	1.25	—	V	

Note 1. The input leakage current value at the EMLE pin is only when V_{in} = 0 V.

Table 49.6 DC Characteristics (3) (Products with 64 Kbytes of RAM)

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = AVCC1 = AVCC2 = 3.0 to 5.5 V, VSS = AVSS0 = AVSS1 = AVSS2 = 0 V,
$T_a = T_{opr}$

Item				Symbol	D version		G version		Unit	Test Conditions
					Typ.	Max.	Typ.	Max.		
Supply current*1	Normal operating mode	Full operation*2		I_{CC}*3	—	66	—	74	mA	ICLK = 120 MHz PCLKA = 120 MHz PCLKB = 60 MHz PCLKC = 120 MHz PCLKD = 60 MHz FCLK = 60 MHz
		Normal operation	Peripheral module clocks are supplied*4		22	—	22	—		
			Peripheral module clocks are stopped*4, *5		11	—	11	—		
		CoreMark	Peripheral module clocks are stopped*4, *5		18	—	18	—		
		Sleep mode: Peripheral module clocks are supplied*4			18	36	18	44		
		All module clock stop mode (reference value)			8.1	22	8.1	29		
		Increase current by BGO operation*6			16	—	16	—		
		Increase current by operating Trusted Secure IP			4.3	5.2	4.3	5.2		
	Software standby mode				0.9	8	0.9	13		

Note 1. Supply current values are measured when all output pins are unloaded and all input pull-up resistors are disabled.
Note 2. Peripheral module clocks are supplied. This does not include operations as BGO (background operations).
Note 3. I_{CC} depends on f (ICLK) as follows.
• D version product
I_{CC} Max. = 0.417 × f + 16 (full operation in normal operating mode)
I_{CC} Typ. = 0.144 × f + 5 (normal operation in normal operating mode)
I_{CC} Max. = 0.167 × f + 16 (sleep mode)
• G version product
I_{CC} Max. = 0.433 × f + 22 (full operation in normal operating mode)
I_{CC} Typ. = 0.144 × f + 5 (normal operation in normal operating mode)
I_{CC} Max. = 0.183 × f + 22 (sleep mode)
Note 4. This does not include operations as BGO (background operations). Whether the peripheral module clocks are supplied or stopped is controlled only by the bit settings in the module stop control registers A to D.
Note 5. When peripheral module clocks are stopped, each clock frequency is set for division by 64, and the frequencies of FCLK, PCLKA, PCLKB, PCLKC, and PCLKD are the same.
Note 6. This is an increase caused by program/erase operation to the code flash memory or data flash memory during executing the user program.

Table 49.7 DC Characteristics (3) (Products with 48 Kbytes of RAM)

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = AVCC1 = AVCC2 = 3.0 to 5.5 V, VSS = AVSS0 = AVSS1 = AVSS2 = 0 V,
$T_a = T_{opr}$

Item				Symbol	D version		G version		Unit	Test Conditions
					Typ.	Max.	Typ.	Max.		
Supply current*1	Normal operating mode	Full operation*2		I_{CC}*3	—	47	—	52	mA	ICLK = 120 MHz PCLKA = 120 MHz PCLKB = 60 MHz PCLKC = 120 MHz PCLKD = 60 MHz FCLK = 60 MHz
		Normal operation	Peripheral module clocks are supplied*4		17	—	17	—		
			Peripheral module clocks are stopped*4, *5		10	—	10	—		
		CoreMark	Peripheral module clocks are stopped*4, *5		16	—	16	—		
		Sleep mode: Peripheral module clocks are supplied*4			13	25	13	29		
		All module clock stop mode (reference value)			7.4	16	7.4	20		
		Increase current by BGO operation*6			12	—	12	—		
	Software standby mode				0.9	5	0.9	8		

Note 1. Supply current values are measured when all output pins are unloaded and all input pull-up resistors are disabled.
Note 2. Peripheral module clocks are supplied. This does not include operations as BGO (background operations).
Note 3. I_{CC} depends on f (ICLK) as follows.
• D version product
I_{CC} Max. = 0.283 × f + 13 (full operation in normal operating mode)
I_{CC} Typ. = 0.107 × f + 4.3 (normal operation in normal operating mode)
I_{CC} Max. = 0.100 × f + 13 (sleep mode)
• G version product
I_{CC} Max. = 0.285 × f + 17.8 (full operation in normal operating mode)
I_{CC} Typ. = 0.107 × f + 4.3 (normal operation in normal operating mode)
I_{CC} Max. = 0.093 × f + 17.8 (sleep mode)
Note 4. This does not include operations as BGO (background operations). Whether the peripheral module clocks are supplied or stopped is controlled only by the bit settings in the module stop control registers A to D.
Note 5. When peripheral module clocks are stopped, each clock frequency is set for division by 64, and the frequencies of FCLK, PCLKA, PCLKB, PCLKC, and PCLKD are the same.
Note 6. This is an increase caused by program/erase operation to the code flash memory or data flash memory during executing the user program.

Table 49.8 DC Characteristics (4)

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = AVCC1 = AVCC2 = 3.0 to 5.5 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V,
$T_a = T_{opr}$

Item			Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Analog power supply current	Unit 0	During 12-bit A/D conversion (Channel-dedicated sample-and-hold circuits: operation for all channels; PGA: enabled for all channels)	AI_{CC}	—	2.7	6.1	mA	IAVCC0_AD + SH + PGA
		During 12-bit A/D conversion (Channel-dedicated sample-and-hold circuits: operation for all channels; PGA: disabled for all channels)		—	2.0	3.0		IAVCC0_AD + SH
		During 12-bit A/D conversion (Channel-dedicated sample-and-hold circuits: stopping of all channels; PGA: enabled for all channels)		—	1.9	5.0		IAVCC0_AD + PGA
		During 12-bit A/D conversion (Channel-dedicated sample-and-hold circuits: stopping of all channels; PGA: disabled for all channels)		—	1.0	1.5		IAVCC0_AD
	Unit 1	During 12-bit A/D conversion (Channel-dedicated sample-and-hold circuits: operation for all channels; PGA: enabled for all channels)		—	2.7	6.1		IAVCC1_AD + SH + PGA
		During 12-bit A/D conversion (Channel-dedicated sample-and-hold circuits: operation for all channels; PGA: disabled for all channels)		—	2.0	3.0		IAVCC1_AD + SH
		During 12-bit A/D conversion (Channel-dedicated sample-and-hold circuits: stopping of all channels; PGA: enabled for all channels)		—	1.9	5.0		IAVCC1_AD + PGA
		During 12-bit A/D conversion (Channel-dedicated sample-and-hold circuits: stopping of all channels; PGA: disabled for all channels)		—	1.0	1.5		IAVCC1_AD
	Unit 2	During 12-bit A/D conversion with the temperature sensor operating		—	1.0	1.5		IAVCC2_AD + TEMP
		During 12-bit A/D conversion with the temperature sensor stopped		—	0.9	1.4		IAVCC2_AD
	Comparator (6 channels)			—	0.6	0.8		IAVCC2_CMP
	During 12-bit D/A conversion (2 channels)			—	0.6	0.8		IAVCC2_DA
	Waiting for 12-bit A/D, 12-bit D/A, Comparator C, and temperature sensor conversion (all units)			—	0.05	0.1		IAVCC0_AD + IAVCC1_AD + IAVCC2_AD + IAVCC2_DA
	12-bit A/D, 12-bit D/A, Comparator C, and temperature sensor are in module stop status (all units)			—	0.3	11.1	μA	IAVCC0_AD + IAVCC1_AD + IAVCC2_AD + IAVCC2_DA
RAM retention voltage			V_{RAM}	2.7	—	—	V	

Table 49.9 DC Characteristics (5)

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = AVCC1 = AVCC2 = 3.0 to 5.5 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V,
$T_a = T_{opr}$

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
VCC ramp rate at power-on	At normal startup	SrVCC	0.02	—	8	ms/V	
	Voltage monitoring 0 reset enabled at startup*1, *2		0.02	—	20		
VCC ramp rate at power fluctuation		dt/dVCC	1.0	—	—	ms/V	When VCC change exceeds VCC ±10%

Note 1. When OFS1.LVDAS = 0.
Note 2. Settings of the OFS1 register are not read in boot mode, so turn on the power supply voltage with a ramp rate at normal startup.

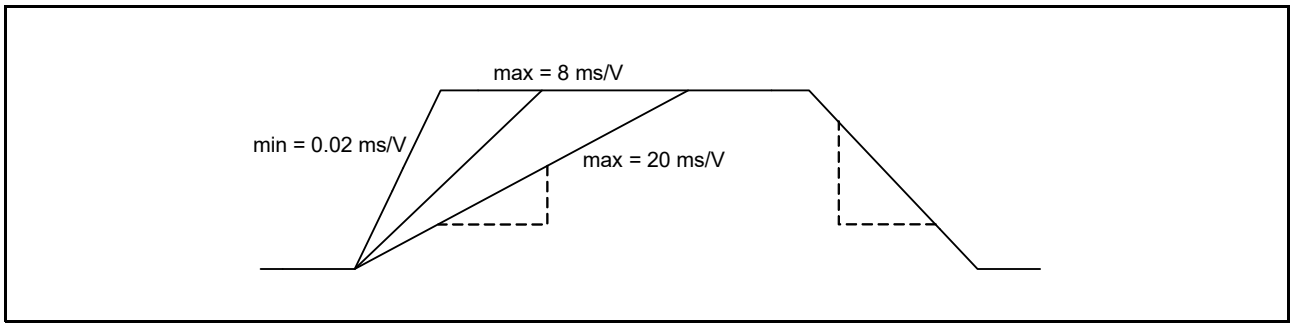


Figure 49.1 VCC Ramp Rate at Power-On

Table 49.10 Permissible Output Currents

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = AVCC1 = AVCC2 = 3.0 to 5.5 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V,
$T_a = T_{opr}$

Item			Symbol	Min.	Typ.	Max.	Unit
Permissible low-level output current (average value per pin)	All output pins (except for RIIC pins, P40 to P47, P50 to P55, and P60 to P65)	Normal drive*1	I_{OL}	—	—	2.0	mA
		High drive*2		—	—	2.0	
		Large current output*3		—	—	15.0	
	RIIC pins	Standard mode		—	—	3	
		Fast mode		—	—	6	
	P40 to P47, P50 to P55, and P60 to P65			—	—	2.0	
Permissible low-level output current (max. value per pin)	All output pins (except for RIIC pins, P40 to P47, P50 to P55, and P60 to P65)	Normal drive*1		—	—	4.0	
		High drive*2		—	—	4.0	
		High drive*2, *4		—	—	15.0	
		Large current output*3		—	—	15.0	
	RIIC pins	Standard mode		—	—	3	
		Fast mode		—	—	6	
	P40 to P47, P50 to P55, and P60 to P65			—	—	4.0	
Permissible low-level output current (total)	Total of all output pins		ΣI_{OL}	—	—	110	mA
Permissible high-level output current (average value per pin)	All output pins (except for P40 to P47, P50 to P55, and P60 to P65)	Normal drive*1	I_{OH}	—	—	–2.0	mA
		High drive*2		—	—	–2.0	
		Large current output*3		—	—	–5.0	
		Large current output*3, *5		—	—	–15.0	
	P40 to P47, P50 to P55, and P60 to P65			—	—	–2.0	
Permissible high-level output current (max. value per pin)	All output pins (except for P40 to P47, P50 to P55, and P60 to P65)	Normal drive*1		—	—	–4.0	
		High drive*2		—	—	–4.0	
		Large current output*3		—	—	–5.0	
		Large current output*3, *5		—	—	–15.0	
	P40 to P47, P50 to P55, and P60 to P65			—	—	–4.0	
Permissible high-level output current (total)	Total of all output pins		ΣI_{OH}	—	—	–35	mA

Caution: To protect the LSI's reliability, the output current values should not exceed the values in this table.

Note 1. The listed value applies when normal driving ability is set with a pin for which normal driving ability is selectable.

Note 2. The listed value applies when high driving ability is set with a pin for which normal driving ability is selectable, or when the pin to which high driving ability is fixed is in use.

Note 3. The listed value applies when large current output is set with a pin for which large current output ability is selectable.

Note 4. The listed value applies when VCC is at least 4.5 V.

Note 5. The listed value applies when VCC is at least 4.0 V.

Table 49.11 Standard Output Characteristics (1)

Conditions: VCC = AVCC0 = AVCC1 = AVCC2 = 5.0 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V, T_a = 25°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Output high voltage	Normal drive output (all output pins)	V_{OH}	—	4.97	—	V	I_{OH} = –0.5 mA
			—	4.94	—		I_{OH} = –1.0 mA
			—	4.87	—		I_{OH} = –2.0 mA
			—	4.74	—		I_{OH} = –4.0 mA
	High-drive output (P00, P01, P10, P11, P20 to P24, P27, P30 to P33, P70 to P76, P80 to P82, P90 to P96, PA0 to PA5, PB0, PB3 to PB7, PD0 to PD7, PE0, PE1, PE3 to PE5, and PN6)		—	4.98	—		I_{OH} = –0.5 mA
			—	4.97	—		I_{OH} = –1.0 mA
			—	4.94	—		I_{OH} = –2.0 mA
			—	4.87	—		I_{OH} = –4.0 mA
	Large current output (P71 to P76, P81, P90 to P95, PB5, PD3)		—	4.99	—		I_{OH} = –0.5 mA
			—	4.98	—		I_{OH} = –1.0 mA
			—	4.96	—		I_{OH} = –2.0 mA
			—	4.92	—		I_{OH} = –4.0 mA
			—	4.91	—		I_{OH} = –5.0 mA
Output low voltage	Normal drive output (all output pins)	V_{OL}	—	0.02	—	V	I_{OL} = 0.5 mA
			—	0.04	—		I_{OL} = 1.0 mA
			—	0.09	—		I_{OL} = 2.0 mA
			—	0.18	—		I_{OL} = 4.0 mA
	High-drive output (P00, P01, P10, P11, P20 to P24, P27, P30 to P33, P70 to P76, P80 to P82, P90 to P96, PA0 to PA5, PB0, PB3 to PB7, PD0 to PD7, PE0, PE1, PE3 to PE5, and PN6)		—	0.01	—		I_{OL} = 0.5 mA
			—	0.03	—		I_{OL} = 1.0 mA
			—	0.05	—		I_{OL} = 2.0 mA
			—	0.10	—		I_{OL} = 4.0 mA
			—	0.42	—		I_{OL} = 15.0 mA
	Large current output (P71 to P76, P81, P90 to P95, PB5, PD3)		—	0.01	—		I_{OL} = 0.5 mA
			—	0.02	—		I_{OL} = 1.0 mA
			—	0.04	—		I_{OL} = 2.0 mA
			—	0.07	—		I_{OL} = 4.0 mA
			—	0.09	—		I_{OL} = 5.0 mA
			—	0.18	—		I_{OL} = 10.0 mA
			—	0.28	—		I_{OL} = 15.0 mA

Table 49.12 Standard Output Characteristics (2)

Conditions: VCC = AVCC0 = AVCC1 = AVCC2 = 3.3 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V, T_a = 25°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Output high voltage	Normal drive output (all output pins)	V_{OH}	—	3.26	—	V	I_{OH} = –0.5 mA
			—	3.22	—		I_{OH} = –1.0 mA
			—	3.13	—		I_{OH} = –2.0 mA
			—	2.94	—		I_{OH} = –4.0 mA
	High-drive output (P00, P01, P10, P11, P20 to P24, P27, P30 to P33, P70 to P76, P80 to P82, P90 to P96, PA0 to PA5, PB0, PB3 to PB7, PD0 to PD7, PE0, PE1, PE3 to PE5, and PN6)		—	3.28	—		I_{OH} = –0.5 mA
			—	3.26	—		I_{OH} = –1.0 mA
			—	3.22	—		I_{OH} = –2.0 mA
			—	3.13	—		I_{OH} = –4.0 mA
	Large current output (P71 to P76, P81, P90 to P95, PB5, PD3)		—	3.29	—		I_{OH} = –0.5 mA
			—	3.27	—		I_{OH} = –1.0 mA
			—	3.25	—		I_{OH} = –2.0 mA
			—	3.20	—		I_{OH} = –4.0 mA
			—	3.17	—		I_{OH} = –5.0 mA
Output low voltage	Normal drive output (all output pins)	V_{OL}	—	0.03	—	V	I_{OL} = 0.5 mA
			—	0.06	—		I_{OL} = 1.0 mA
			—	0.12	—		I_{OL} = 2.0 mA
			—	0.25	—		I_{OL} = 4.0 mA
	High-drive output (P00, P01, P10, P11, P20 to P24, P27, P30 to P33, P70 to P76, P80 to P82, P90 to P96, PA0 to PA5, PB0, PB3 to PB7, PD0 to PD7, PE0, PE1, PE3 to PE5, and PN6)		—	0.02	—		I_{OL} = 0.5 mA
			—	0.03	—		I_{OL} = 1.0 mA
			—	0.07	—		I_{OL} = 2.0 mA
			—	0.13	—		I_{OL} = 4.0 mA
	Large current output (P71 to P76, P81, P90 to P95, PB5, PD3)		—	0.01	—		I_{OL} = 0.5 mA
			—	0.02	—		I_{OL} = 1.0 mA
			—	0.05	—		I_{OL} = 2.0 mA
			—	0.09	—		I_{OL} = 4.0 mA
			—	0.11	—		I_{OL} = 5.0 mA
			—	0.24	—		I_{OL} = 10.0 mA
			—	0.36	—		I_{OL} = 15.0 mA

Table 49.13 Thermal Resistance Value (Reference)

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = AVCC1 = AVCC2 = 3.0 to 5.5 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V,
$T_a = T_{opr}$

Item	Package	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Thermal resistance	100-pin LFQFP (PLQP0100KB-B)	θ_{ja}	—	—	50.5	°C/W	JESD51-2 and JESD51-7 compliant
	80-pin LFQFP (PLQP0080KB-B)		—	—	47.7		
	64-pin LFQFP (PLQP0064KB-C)		—	—	51.9		
	64-pin HWQFN (PWQN0064KF-A)		—	—	18.4*1		
	48-pin LFQFP (PLQP0048KB-B)		—	—	60.8		
	48-pin HWQFN (PWQN0048KC-A)		—	—	19.5*1		
	100-pin LFQFP (PLQP0100KB-B)	Ψ_{jt}	—	—	1.39	°C/W	JESD51-2 and JESD51-7 compliant
	80-pin LFQFP (PLQP0080KB-B)		—	—	1.39		
	64-pin LFQFP (PLQP0064KB-C)		—	—	1.88		
	64-pin HWQFN (PWQN0064KF-A)		—	—	0.12*1		
	48-pin LFQFP (PLQP0048KB-B)		—	—	2.38		
	48-pin HWQFN (PWQN0048KC-A)		—	—	0.12*1		

Note: The values are reference values when the 4-layer printed circuit board is used. Thermal resistance depends on the number of layers and size of the board. For details, refer to the JEDEC standards.

Note 1. The listed value applies when the exposed die pad is connected to VSS.

49.4 AC Characteristics

Table 49.14 Operating Frequency

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = AVCC1 = AVCC2 = 3.0 to 5.5 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V,
$T_a = T_{opr}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
System clock (ICLK)	f	—	—	120	MHz	
Peripheral module clock (PCLKA)		—	—	120		
Peripheral module clock (PCLKB)		—	—	60		
Peripheral module clock (PCLKC)		—	—	120		
Peripheral module clock (PCLKD)		8*1	—	60		AVCC0 = AVCC1 = AVCC2 ≥ 4.5 V
		8*1	—	40		AVCC0 = AVCC1 = AVCC2 < 4.5 V
Flash-IF clock (FCLK)		4*2	—	60		

Note 1. This restriction is only applied when a 12-bit A/D converter is to be used.
Note 2. This restriction is only applied when flash memory is to be programmed or erased.

49.4.1 Reset Timing

Table 49.15 Reset Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = AVCC1 = AVCC2 = 3.0 to 5.5 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V,
$T_a = T_{opr}$

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
RES# pulse width	Power-on	t_{RESWP}	2.0	—	—	ms	Figure 49.2
	Software standby mode	t_{RESWS}	0.3	—	—		Figure 49.3
	Programming or erasure of the code flash memory, or programming, erasure or blank checking of the data flash memory	t_{RESWF}	200	—	—	µs	
	Other than above	t_{RESW}	200	—	—		
Waiting time after release from the RES# pin reset		t_{RESWT}	70	—	71	t_{Lcyc}	Figure 49.2
Internal reset time (independent watchdog timer reset, watchdog timer reset, software reset)		t_{RESW2}	116	—	124	t_{Lcyc}	

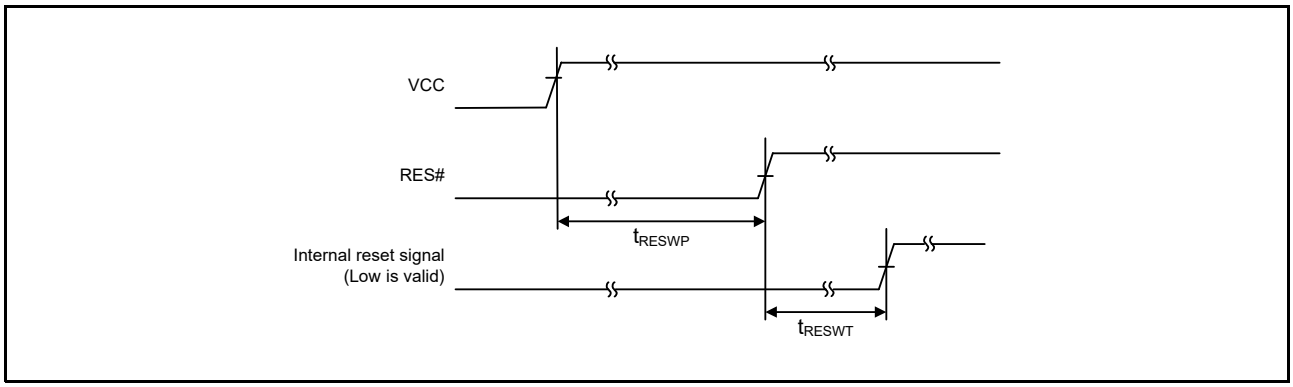


Figure 49.2 Reset Input Timing at Power-On

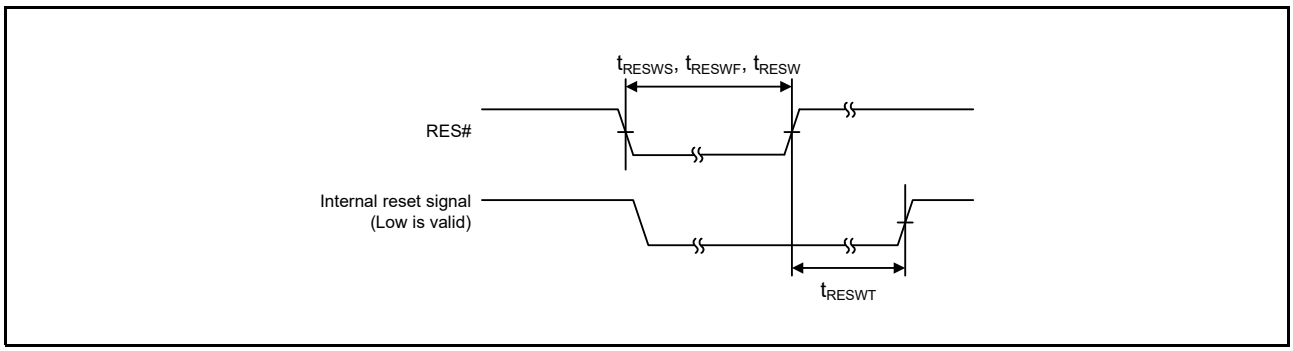


Figure 49.3 Reset Input Timing

49.4.2 Clock Timing

Table 49.16 EXTAL Clock Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = AVCC1 = AVCC2 = 3.0 to 5.5 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V,
$T_a = T_{opr}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
EXTAL external clock input cycle time	t_{EXcyc}	41.66	—	—	ns	Figure 49.4
EXTAL external clock input frequency	f_{EXMAIN}	—	—	24	MHz	
EXTAL external clock input high pulse width	t_{EXH}	15.83	—	—	ns	
EXTAL external clock input low pulse width	t_{EXL}	15.83	—	—	ns	
EXTAL external clock rising time	t_{EXr}	—	—	5	ns	
EXTAL external clock falling time	t_{EXf}	—	—	5	ns	

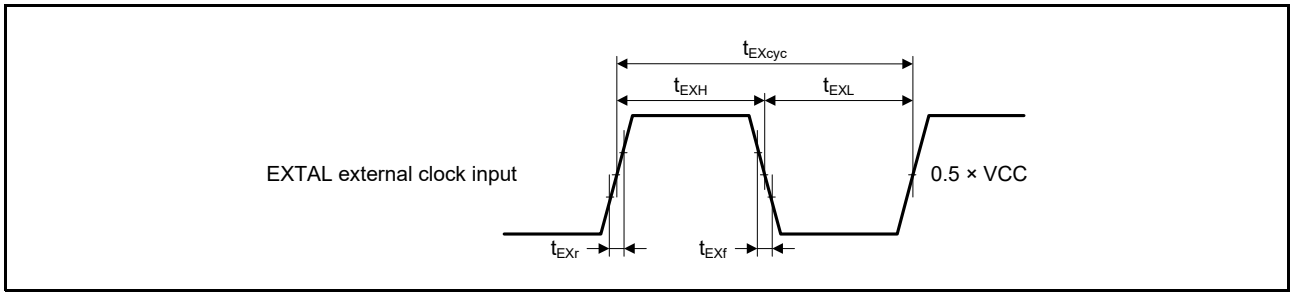


Figure 49.4 EXTAL External Clock Input Timing

Table 49.17 Main Clock Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = AVCC1 = AVCC2 = 3.0 to 5.5 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V,
$T_a = T_{opr}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Main clock oscillation frequency	f_{MAIN}	8	—	24	MHz	
Main clock oscillator stabilization time (crystal)	$t_{MAINOSC}$	—	—	—*1	ms	Figure 49.5
Main clock oscillation stabilization wait time (crystal)	$t_{MAINOSCWT}$	—	—	—*2	ms	

Note 1. When using a main clock, ask the manufacturer of the oscillator to evaluate its oscillation. Refer to the results of evaluation provided by the manufacturer for the oscillation stabilization time.

Note 2. The number of cycles selected by the value of the MOSCWTCR.MSTS[7:0] bits determines the main clock oscillation stabilization wait time in accord with the formula below.
$t_{MAINOSCWT} = [(MSTS[7:0] \text{ bits} \times 32) + 7] / f_{LOCO}$

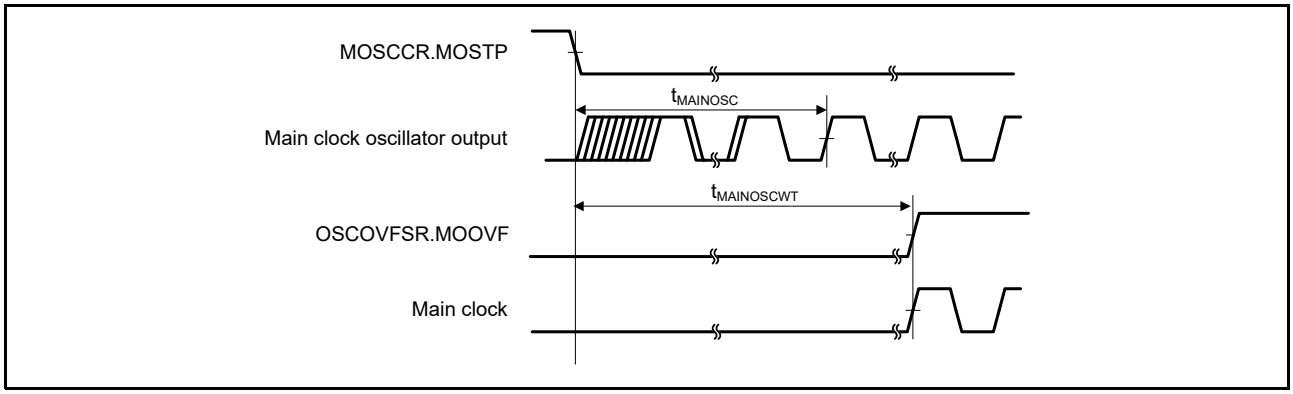


Figure 49.5 Main Clock Oscillation Start Timing

Table 49.18 LOCO and IWDT-Dedicated Low-Speed Clock Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = AVCC1 = AVCC2 = 3.0 to 5.5 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V,
$T_a = T_{opr}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
LOCO clock cycle time	t_{Lcyc}	3.78	4.16	4.63	µs	
LOCO clock oscillation frequency	f_{LOCO}	216 (–10%)	240	264 (+10%)	kHz	
LOCO clock oscillation stabilization time	t_{LOCOWT}	—	—	44	µs	Figure 49.6
IWDT-dedicated low-speed clock cycle time	t_{ILcyc}	7.57	8.33	9.26	µs	
IWDT-dedicated low-speed clock oscillation frequency	f_{ILOCO}	108 (–10%)	120	132 (+10%)	kHz	
IWDT-dedicated low-speed clock oscillation stabilization wait time	$t_{ILOCOWT}$	—	142	190	µs	Figure 49.7

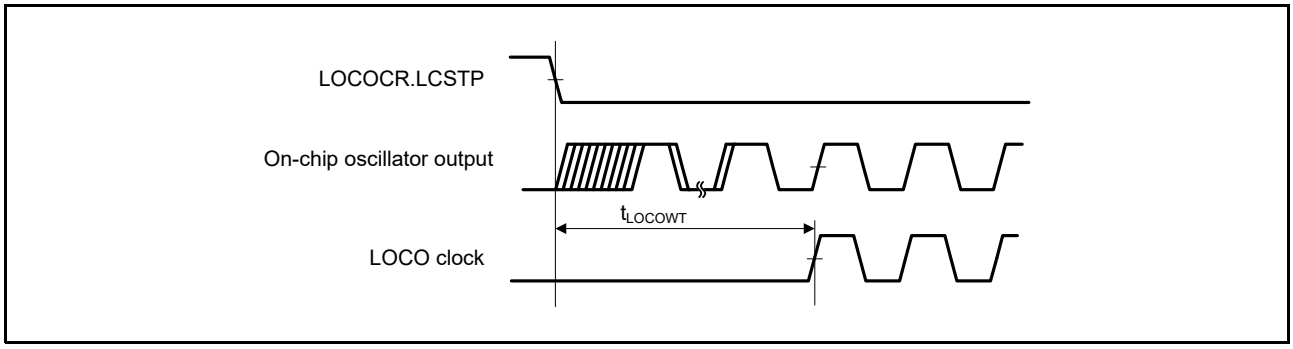


Figure 49.6 LOCO Clock Oscillation Start Timing

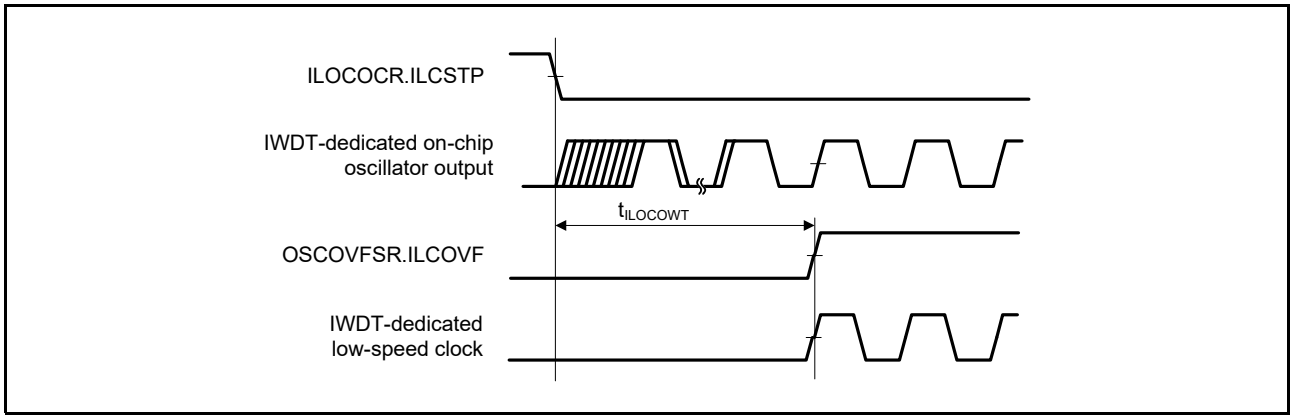


Figure 49.7 IWDT-dedicated Low-Speed Clock Oscillation Start Timing

Table 49.19 HOCO Clock Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = AVCC1 = AVCC2 = 3.0 to 5.5 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V,
$T_a = T_{opr}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
HOCO clock oscillation frequency	f_{HOCO}	15.84 (–1.0%)	16	16.16 (+1.0%)	MHz	–20°C ≤ T_a
		17.82 (–1.0%)	18	18.18 (+1.0%)		
		19.80 (–1.0%)	20	20.20 (+1.0%)		
		15.76 (–1.5%)	16	16.24 (+1.5%)		T_a < –20°C
		17.73 (–1.5%)	18	18.27 (+1.5%)		
		19.70 (–1.5%)	20	20.30 (+1.5%)		
HOCO clock oscillation stabilization wait time	t_{HOCOWT}	—	105	149	µs	Figure 49.8
HOCO clock power supply stabilization time	t_{HOCOP}	—	—	150	µs	Figure 49.9

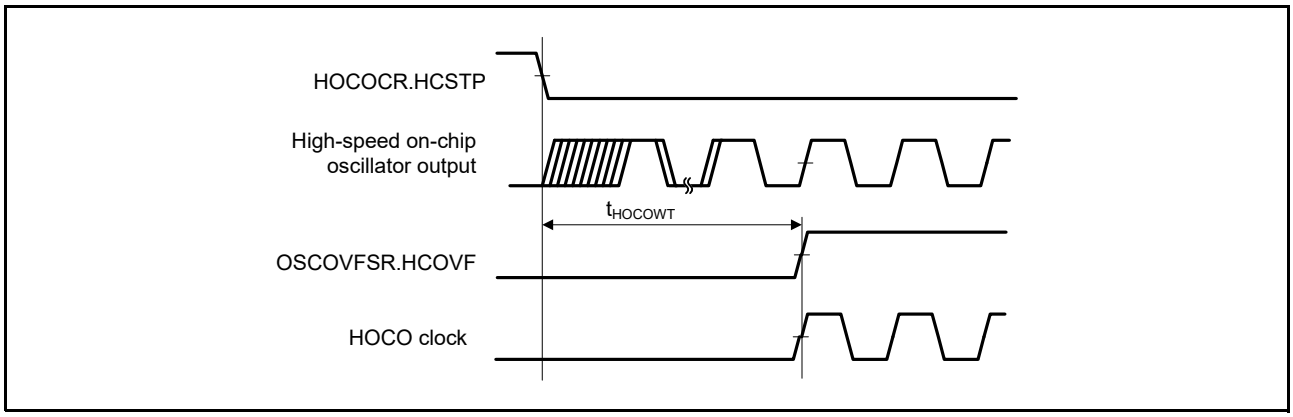


Figure 49.8 HOCO Clock Oscillation Start Timing (Oscillation is Started by Setting the HOCOCR.HCSTP Bit)

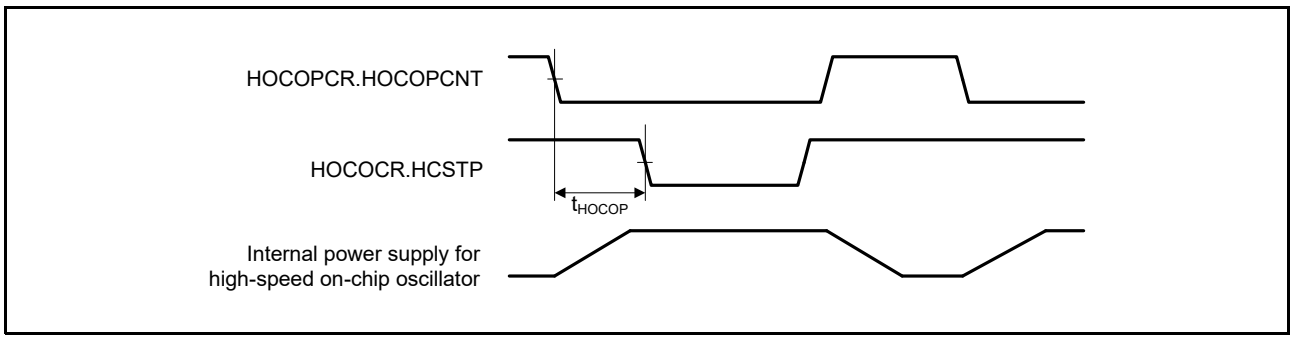


Figure 49.9 High-Speed On-Chip Oscillator Power Supply Control Timing

Table 49.20 PLL Clock Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = AVCC1 = AVCC2 = 3.0 to 5.5 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V,
$T_a = T_{opr}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
PLL clock oscillation frequency	f_{PLL}	120	—	240	MHz	
PLL clock oscillation stabilization wait time	t_{PLLWT}	—	259	320	µs	Figure 49.10

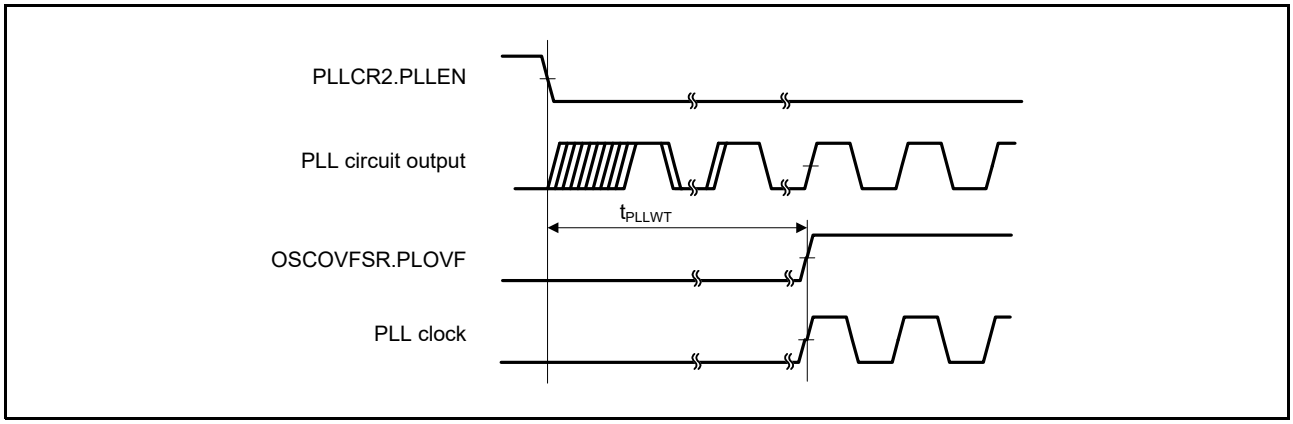


Figure 49.10 PLL Clock Oscillation Start Timing

49.4.3 Timing of Recovery from Low Power Consumption Modes

Table 49.21 Timing of Recovery from Low Power Consumption Modes (1)

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = AVCC1 = AVCC2 = 3.0 to 5.5 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V,
$T_a = T_{opr}$

Item			Symbol	Min.	Typ.	Max.		Unit	Test Conditions
						$t_{SBYOSCWT}$*2	t_{SBYSEQ}*3		
Recovery time after cancellation of software standby mode*1	Crystal resonator connected to main clock oscillator	Main clock oscillator operating	t_{SBYMC}	—	—	{(MSTS[7:0] bits × 32) + 76} / 0.216	100 + 7 / f_{ICLK} + 2n / f_{MAIN}	µs	Figure 49.11
		Main clock oscillator and PLL circuit operating	t_{SBYPC}	—	—	{(MSTS[7:0] bits × 32) + 138} / 0.216	100 + 7 / f_{ICLK} + 2n / f_{PLL}		
	External clock input to main clock oscillator	Main clock oscillator operating	t_{SBYEX}	—	—	352	100 + 7 / f_{ICLK} + 2n / f_{EXMAIN}	µs	
		Main clock oscillator and PLL circuit operating	t_{SBYPE}	—	—	639	100 + 7 / f_{ICLK} + 2n / f_{PLL}		
	High-speed on-chip oscillator operating	High-speed on-chip oscillator operating	t_{SBYHO}	—	—	454	100 + 7 / f_{ICLK} + 2n / f_{HOCO}	µs	
		High-speed on-chip oscillator operating and PLL circuit operating	t_{SBYPH}	—	—	741	100 + 7 / f_{ICLK} + 2n / f_{PLL}		
	Low-speed on-chip oscillator operating*4		t_{SBYLO}	—	—	338	100 + 7 / f_{ICLK} + 2n / f_{LOCO}	µs	

Note 1. The time for return after release from software standby is determined by the value obtained by adding the oscillation stabilization waiting time ($t_{SBYOSCWT}$) and the time required for operations by the software standby release sequencer (t_{SBYSEQ}).
Note 2. When several oscillators were running before the transition to software standby, the greatest value of the oscillation stabilization waiting time $t_{SBYOSCWT}$ is selected.
Note 3. For n, the greatest value is selected from among the internal clock division settings.
Note 4. This condition applies when $f_{ICLK} : f_{FCLK}$ = 1 : 1, 2 : 1, or 4 : 1.

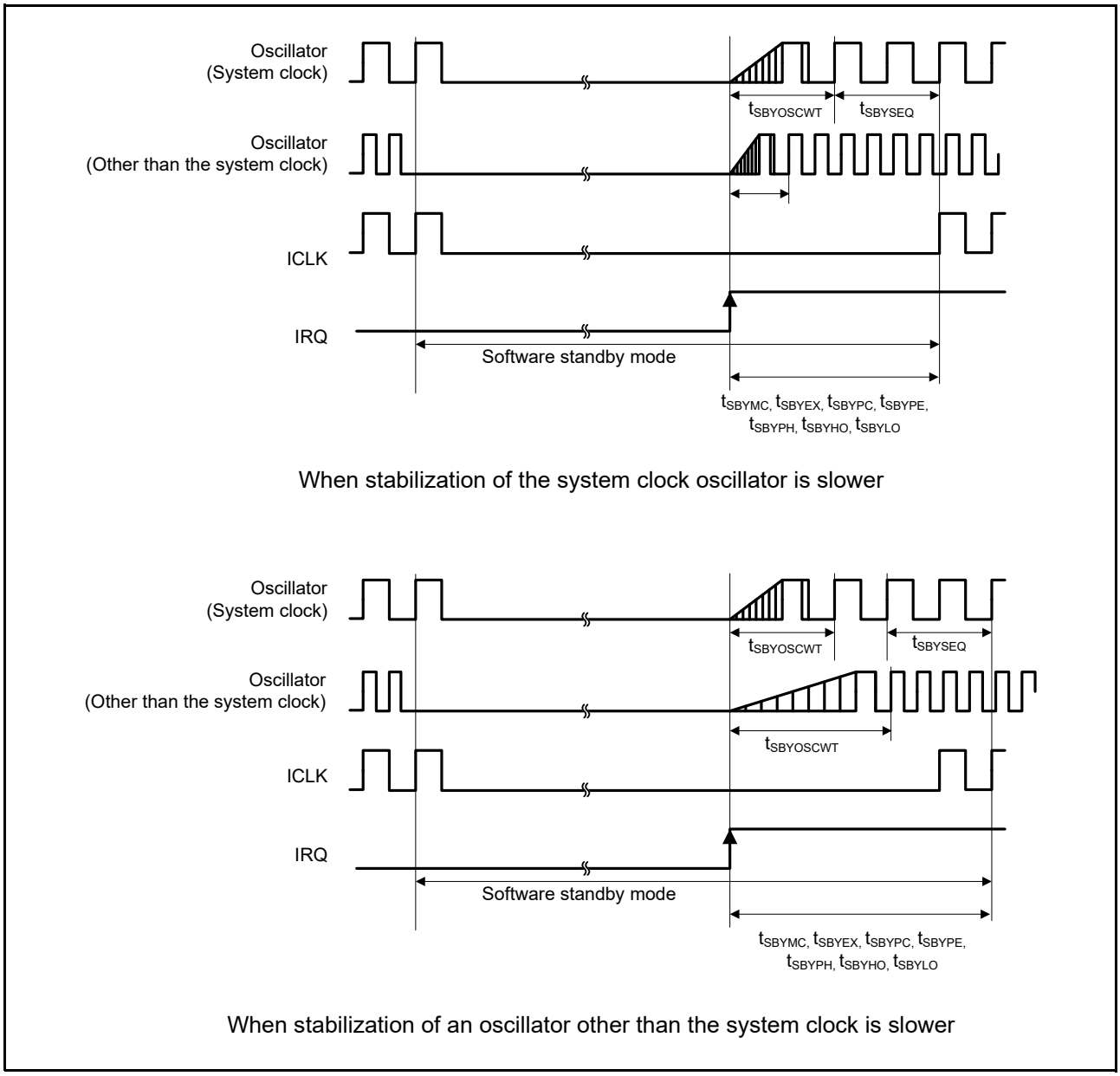


Figure 49.11 Software Standby Mode Cancellation Timing

49.4.4 Control Signal Timing

Table 49.22 Control Signal Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = AVCC1 = AVCC2 = 3.0 to 5.5 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V,
$T_a = T_{opr}$

Item	Symbol	Min.*1	Typ.	Max.	Unit	Test Conditions*1
NMI pulse width	t_{NMIW}	200	—	—	ns	$2 \times t_{PBcyc} \leq 200$ ns, Figure 49.12
		$2 \times t_{PBcyc}$	—	—		$2 \times t_{PBcyc} > 200$ ns, Figure 49.12
IRQ pulse width	t_{IRQW}	200	—	—	ns	$2 \times t_{PBcyc} \leq 200$ ns, Figure 49.13
		$2 \times t_{PBcyc}$	—	—		$2 \times t_{PBcyc} > 200$ ns, Figure 49.13

Note 1. t_{PBcyc}: PCLKB cycle

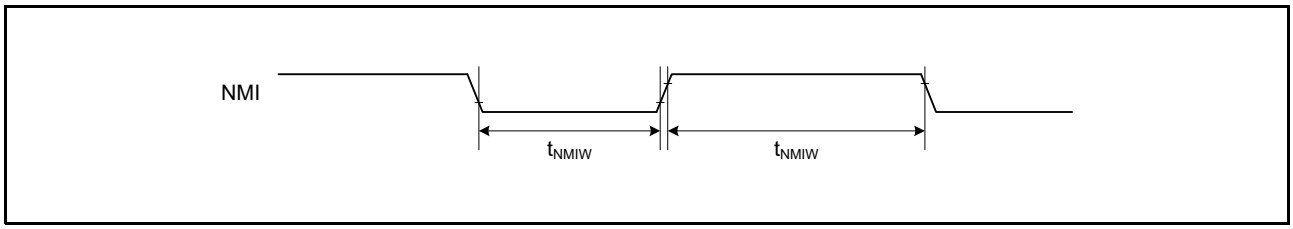


Figure 49.12 NMI Interrupt Input Timing

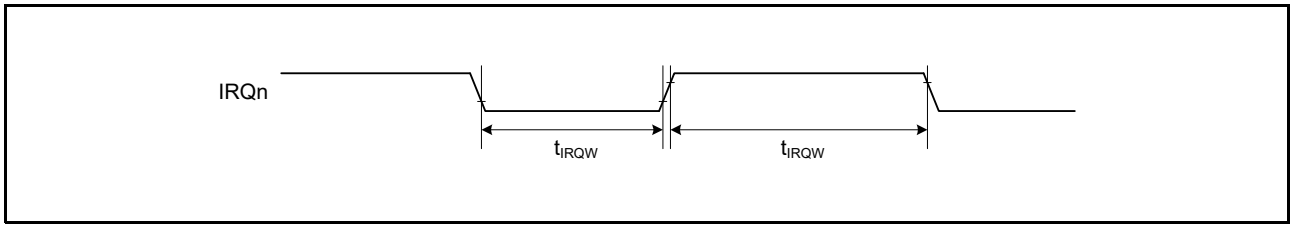


Figure 49.13 IRQ Interrupt Input Timing

49.4.5 Timing of On-Chip Peripheral Modules

49.4.5.1 I/O Port

Table 49.23 I/O Port Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = AVCC1 = AVCC2 = 3.0 to 5.5 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V, $T_a = T_{opr}$,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz,
Output load conditions: V_{OH} = 0.5 × VCC, V_{OL} = 0.5 × VCC, C = 30 pF,
High-drive output is selected by the driving ability control register.

Item		Symbol	Min.	Max.	Unit*1	Test Conditions
I/O ports	Input data pulse width	t_{PRW}	1.5	—	t_{PBcyc}	Figure 49.14

Note 1. t_{PBcyc}: PCLKB cycle

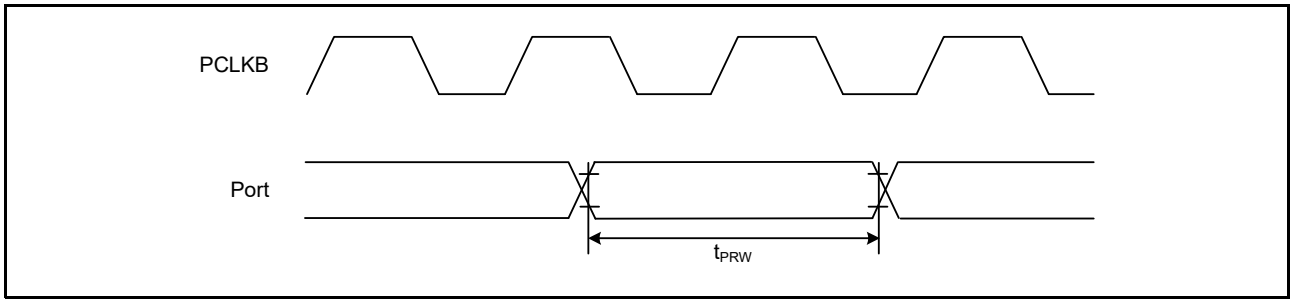


Figure 49.14 I/O Port Input Timing

49.4.5.2 TMR

Table 49.24 TMR Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = AVCC1 = AVCC2 = 3.0 to 5.5 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V, $T_a = T_{opr}$,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz,
Output load conditions: V_{OH} = 0.5 × VCC, V_{OL} = 0.5 × VCC, C = 30 pF,
High-drive output is selected by the driving ability control register.

Item			Symbol	Min.	Max.	Unit*1	Test Conditions
TMR	Timer clock pulse width	Single-edge setting	t_{TMCWH}, t_{TMCWL}	1.5	—	t_{PBcyc}	Figure 49.15
		Both-edge setting		2.5	—		

Note 1. t_{PBcyc}: PCLKB cycle

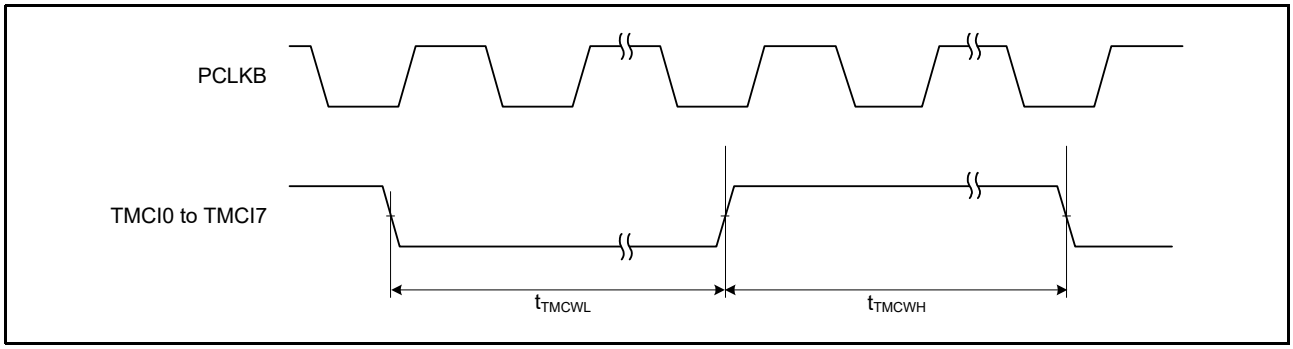


Figure 49.15 TMR Clock Input Timing

49.4.5.3 MTU

Table 49.25 MTU Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = AVCC1 = AVCC2 = 3.0 to 5.5 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V, $T_a = T_{opr}$,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz,
Output load conditions: V_{OH} = 0.5 × VCC, V_{OL} = 0.5 × VCC, C = 30 pF,
High-drive output is selected by the driving ability control register.

Item			Symbol	Min.	Max.	Unit*1	Test Conditions
MTU	Input capture input pulse width	Single-edge setting	t_{MTICW}	1.5	—	t_{PCcyc}	Figure 49.16
		Both-edge setting		2.5	—		
	Timer clock pulse width	Single-edge setting	t_{MTCKWH}, t_{MTCKWL}	1.5	—	t_{PCcyc}	Figure 49.17
		Both-edge setting		2.5	—		
		Phase counting mode		2.5	—		

Note 1. t_{PCcyc}: PCLKC cycle

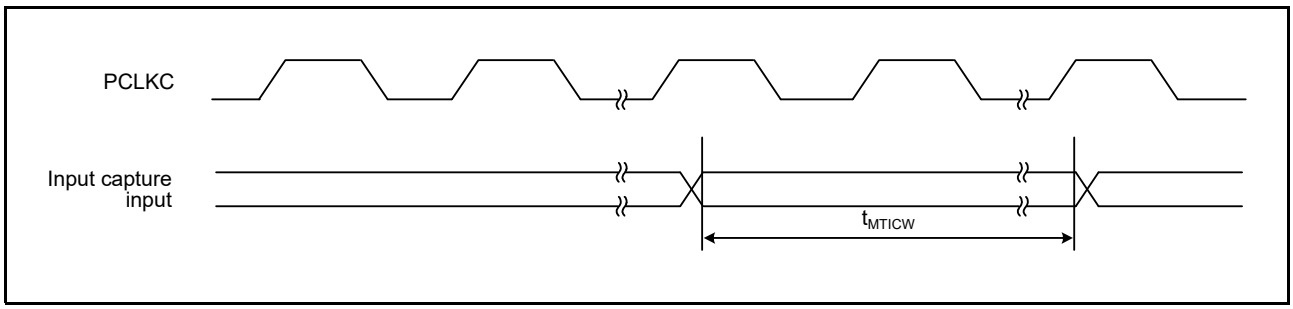


Figure 49.16 MTU Input Capture Input Timing

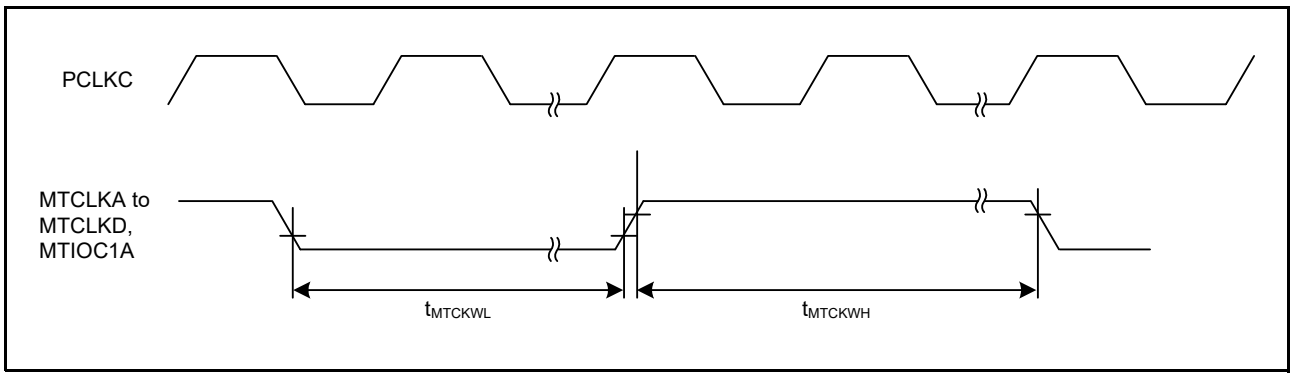


Figure 49.17 MTU Clock Input Timing

49.4.5.4 POE3

Table 49.26 POE3 Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = AVCC1 = AVCC2 = 3.0 to 5.5 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V, $T_a = T_{opr}$,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz,
Output load conditions: V_{OH} = 0.5 × VCC, V_{OL} = 0.5 × VCC, C = 30 pF,
High-drive output is selected by the driving ability control register.

Item			Symbol	Min.	Typ.	Max.	Unit*1	Test Conditions
POE	POEn# input pulse width (n = 0, 4, and 8 to 12)		t_{POEW}	1.5	—	—	t_{PBcyc}	Figure 49.18
	Output disable time	Transition of the POEn# signal level	t_{POEDI}	—	—	5 PCLKB + 0.24	µs	Figure 49.19 When detecting falling edges (ICSRm.POEnM[3:0] = 0000 (m = 1 to 8, n = 0, 4, 8 to 12))
		Simultaneous conduction of output pins	t_{POEDO}	—	—	3 PCLKB + 0.2	µs	Figure 49.20
		Detection of comparator outputs	t_{POEDC}	—	—	5 PCLKB + 0.2	µs	Figure 49.21 The time is that when the noise filter for comparator C is not in use (CMPCTL.CDFS[1:0] = 00) and excludes the time for detection by comparator C.
		Register setting	t_{POEDS}	—	—	1 PCLKB + 0.2	µs	Figure 49.22 Time for access to the register is not included.
		Oscillation stop detection	t_{POEDOS}	—	—	21	µs	Figure 49.23

Note 1. t_{PBcyc}: PCLKB cycle

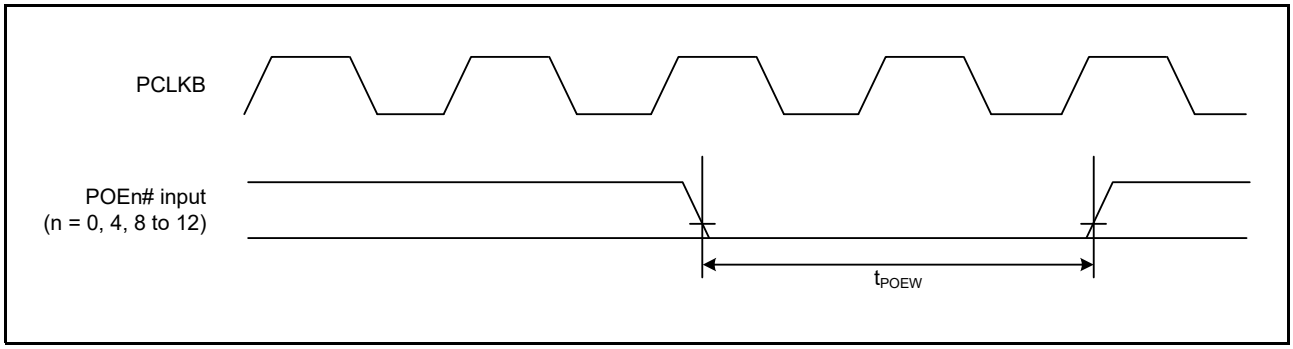


Figure 49.18 POE Input Timing

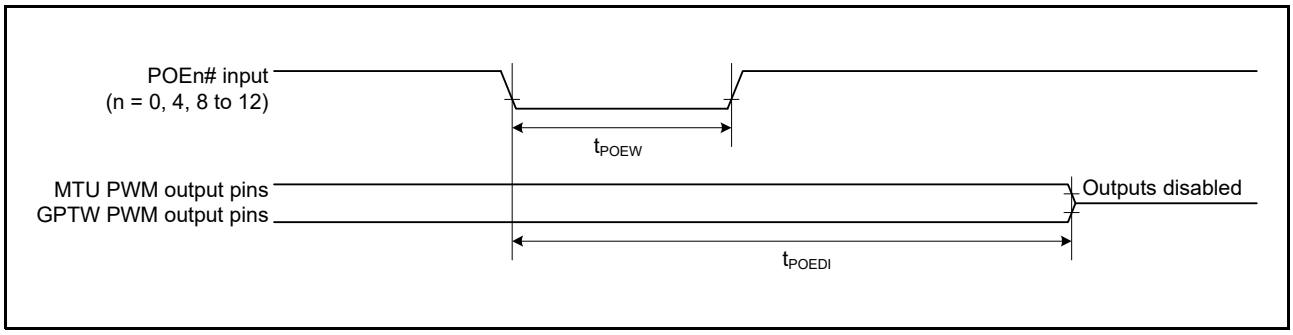
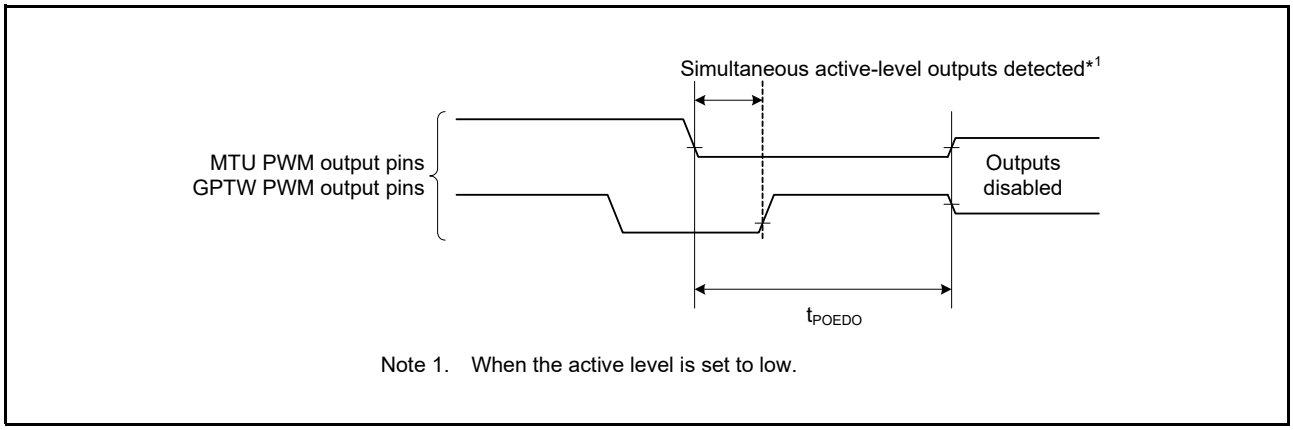


Figure 49.19 Output Disable Time for POE in Response to Transition of the POEn# Signal Level



Note 1. When the active level is set to low.

Figure 49.20 Output Disable Time for POE in Response to the Simultaneous Conduction of Output Pins

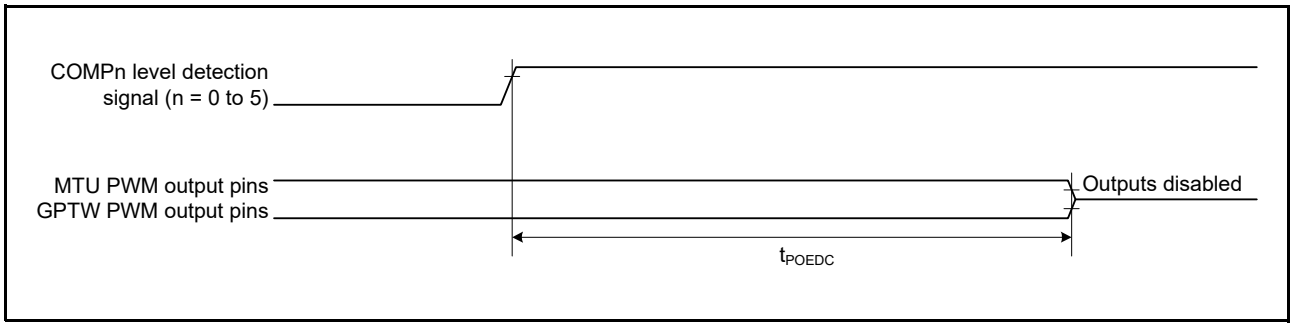


Figure 49.21 Output Disable Time for POE in Response to Detection of the Comparator Outputs

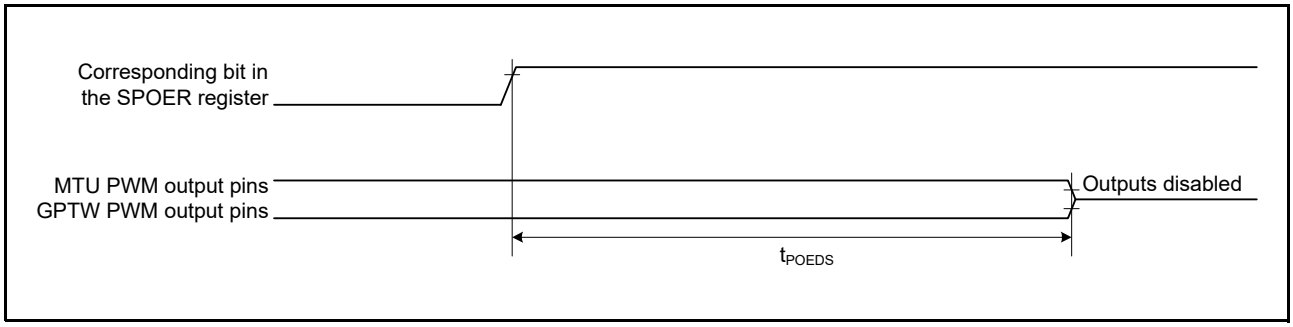


Figure 49.22 Output Disable Time for POE in Response to the Register Setting

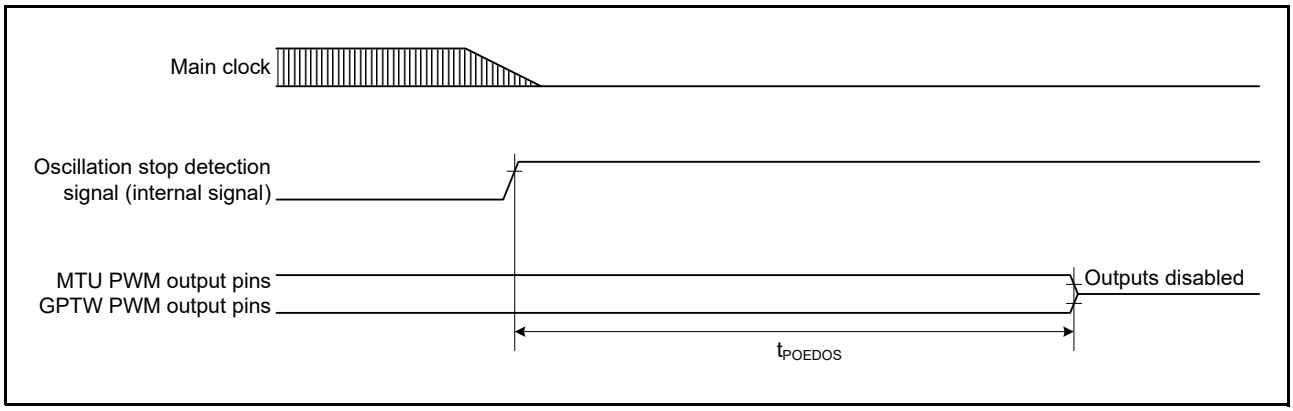


Figure 49.23 Output Disable Time for POE in Response to the Oscillation Stop Detection

49.4.5.5 POEG

Table 49.27 POE and POEG Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = AVCC1 = AVCC2 = 3.0 to 5.5 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V, $T_a = T_{opr}$,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz,
Output load conditions: V_{OH} = 0.5 × VCC, V_{OL} = 0.5 × VCC, C = 30 pF,
High-drive output is selected by the driving ability control register.

Item			Symbol	Min.	Typ.	Max.	Unit*1	Test Conditions
POEG	GTETRGn input pulse width (n = A to D)		t_{POEGW}	1.5	—	—	t_{PBcyc}	Figure 49.24
	Output disable time	Input level detection of the GTETRGn pin (via flag)	t_{POEGDI}	—	—	3 PCLKB + 0.34	µs	Figure 49.25 When the digital noise filter is not in use (POEGGn.NFEN = 0 (n = A to D))
		Detection of the output stopping signal from GPTW (deadtime error, simultaneous high output, or simultaneous low output)	t_{POEGDE}	—	—	0.5	µs	Figure 49.26
		Edge detection signal from a comparator	t_{POEGDC}	—	—	4 PCLKB + 0.5	µs	Figure 49.27 The time is that when the noise filter for comparator C is not in use (CMPCTL.CDFS[1:0] = 00) and excludes the time for detection by comparator C.
		Register setting	t_{POEGDS}	—	—	1 PCLKB + 0.3	µs	Figure 49.28 Time for access to the register is not included.
		Oscillation stop detection	$t_{POEGDOS}$	—	—	21	µs	Figure 49.29
		Input level detection of the GTETRGn pin (direct path)	$t_{POEGDDI}$	—	—	2 PCLKB + 1 PCLKC + 0.34	µs	Figure 49.30
		Level detection signal from a comparator	$t_{POEGDDC}$	—	—	3 PCLKB + 0.3	µs	Figure 49.31 The time is that when the noise filter for comparator C is not in use (CMPCTL.CDFS[1:0] = 00) and excludes the time for detection by comparator C.

Note 1. t_{PBcyc}: PCLKB cycle

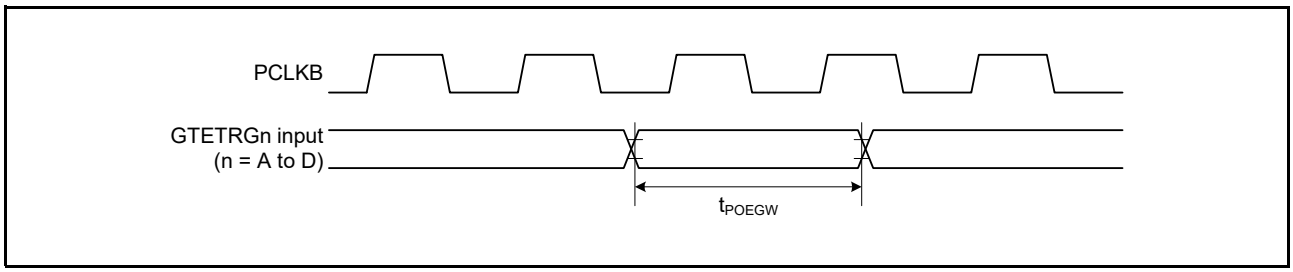


Figure 49.24 POEG Input Timing

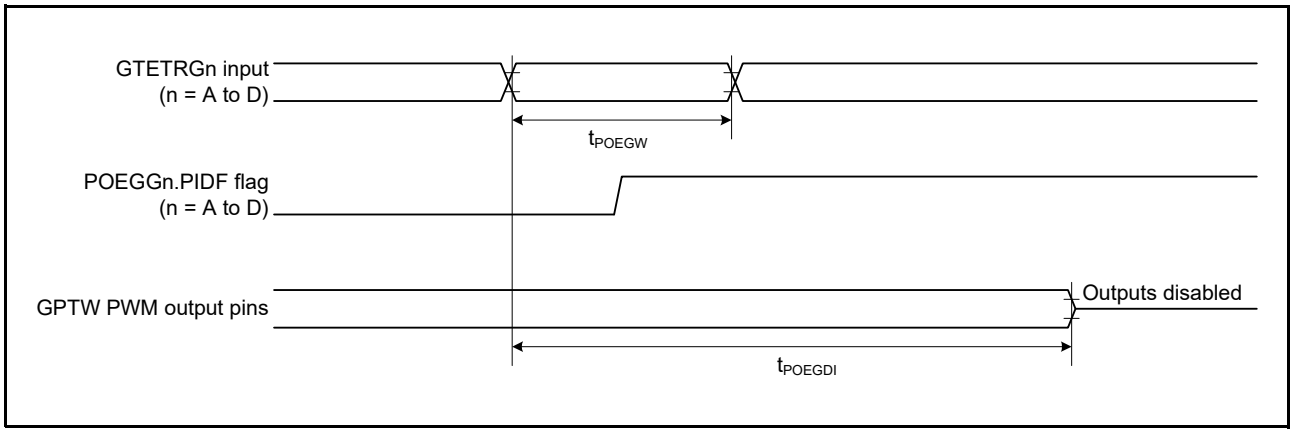
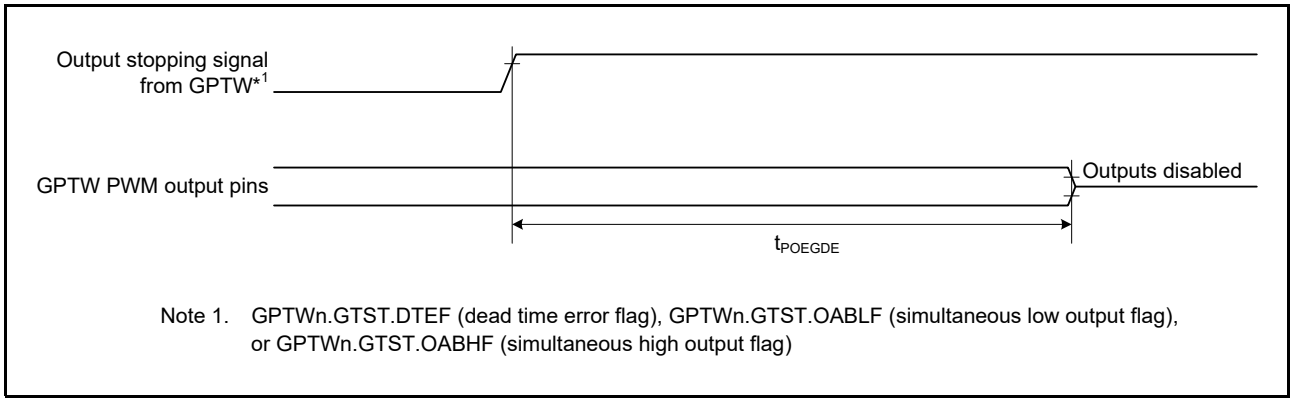


Figure 49.25 Output Disable Time for POEG via Detection Flag in Response to the Input Level Detection of the GTETRGn pin



Note 1. GPTWn.GTST.DTEF (dead time error flag), GPTWn.GTST.OABLF (simultaneous low output flag), or GPTWn.GTST.OABHF (simultaneous high output flag)

Figure 49.26 Output Disable Time for POEG in Response to Detection of the Output Stopping Signal from GPTW

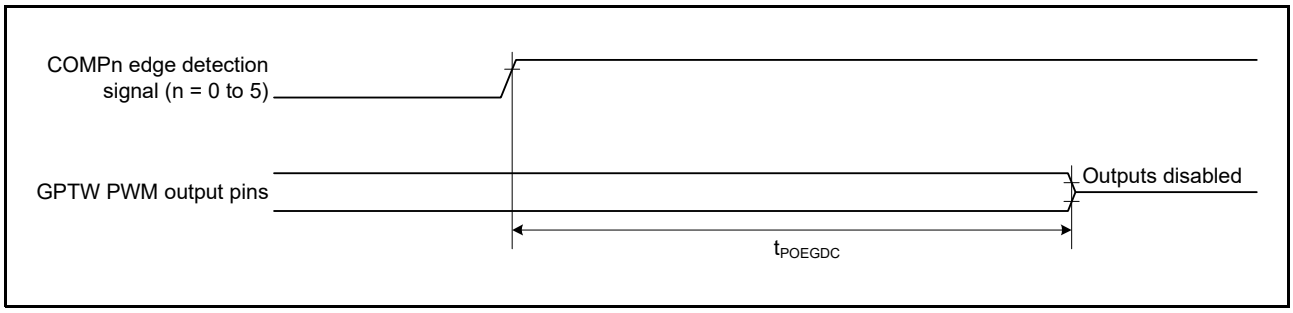


Figure 49.27 Output Disable Time for POEG in Response to Edge Detection Signal from a Comparator

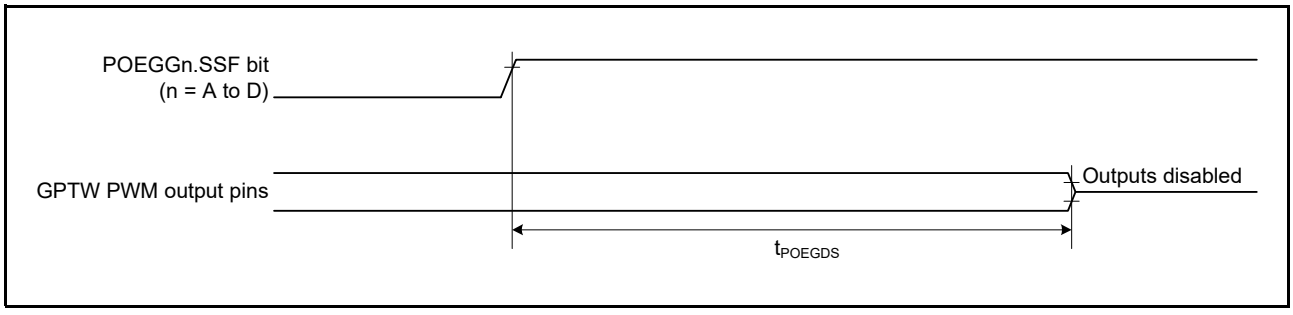


Figure 49.28 Output Disable Time for POEG in Response to the Register Setting

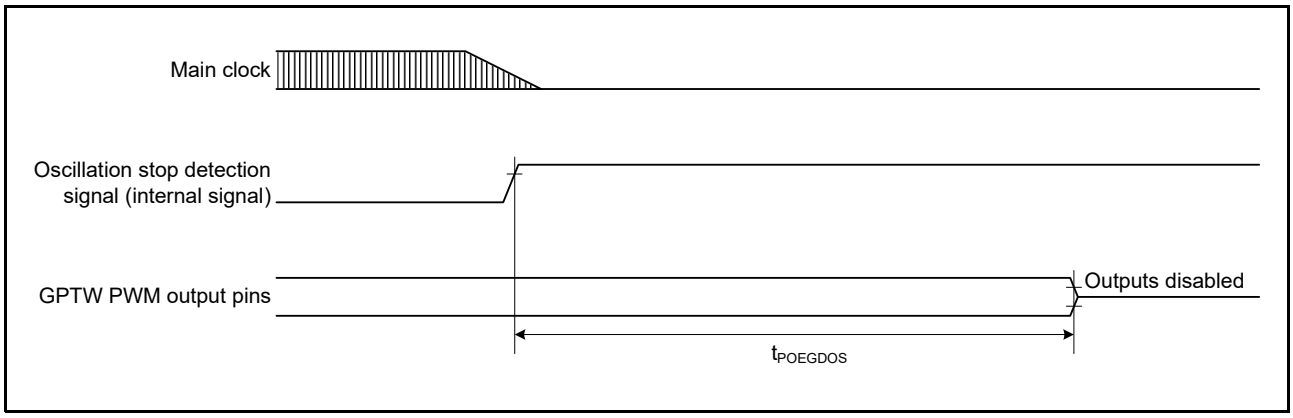


Figure 49.29 Output Disable Time of POEG in Response to the Oscillation Stop Detection

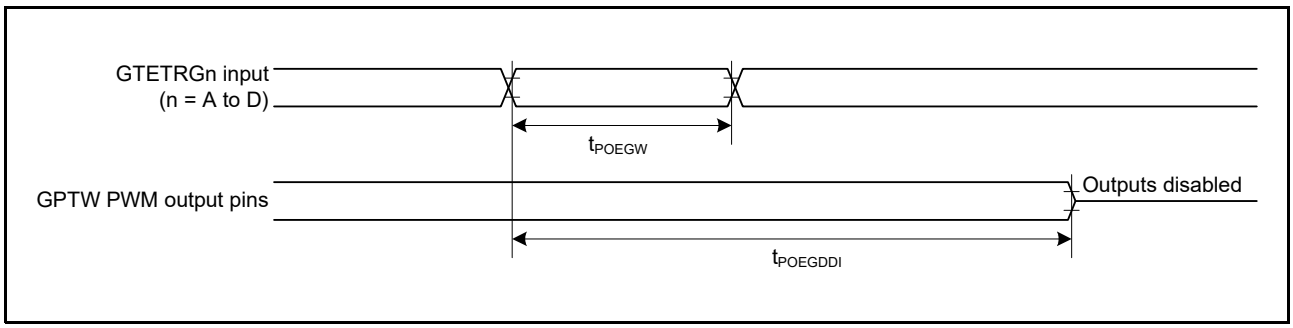


Figure 49.30 Output Disable Time for POEG in Direct Response to the Input Level Detection of the GTETRGn pin

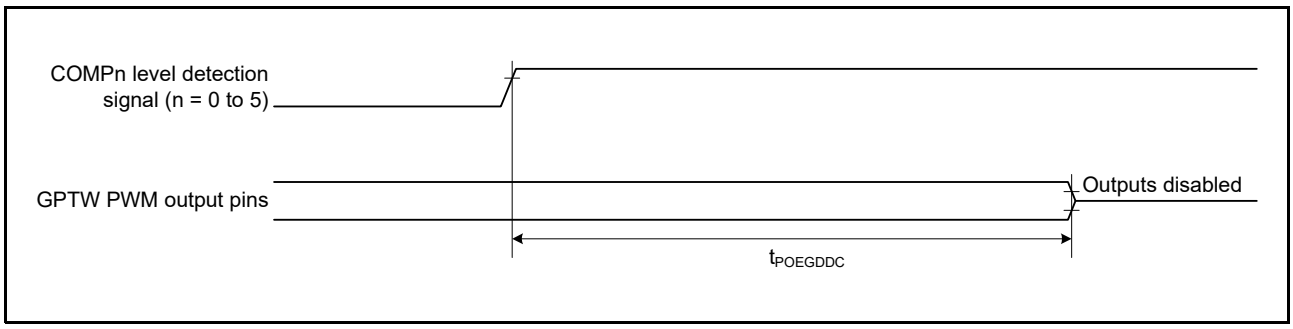


Figure 49.31 Output Disable Time for POEG in Response to Level Detection Signal from a Comparator

49.4.5.6 GPTW

Table 49.28 GPTW Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = AVCC1 = AVCC2 = 3.0 to 5.5 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V, $T_a = T_{opr}$,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz,
Output load conditions: V_{OH} = 0.5 × VCC, V_{OL} = 0.5 × VCC, C = 30 pF,
High-drive output is selected by the driving ability control register.

Item			Symbol	Min.	Max.	Unit*1, *2	Test Conditions
GPTW	Input capture input pulse width	Single-edge setting	t_{GTICW}	1.5	—	t_{PCcyc}	Figure 49.32
		Both-edge setting		2.5	—		
	External trigger input pulse width	Single-edge setting	t_{GTEW}	1.5	—	t_{PBcyc}	Figure 49.33
		Both-edge setting		2.5	—		
	Timer clock pulse width		t_{GTCKWH}	1.5	—	t_{PBcyc}	Figure 49.34
			t_{GTCKWL}				

Note 1. t_{PCcyc}: PCLKC cycle
Note 2. t_{PBcyc}: PCLKB cycle

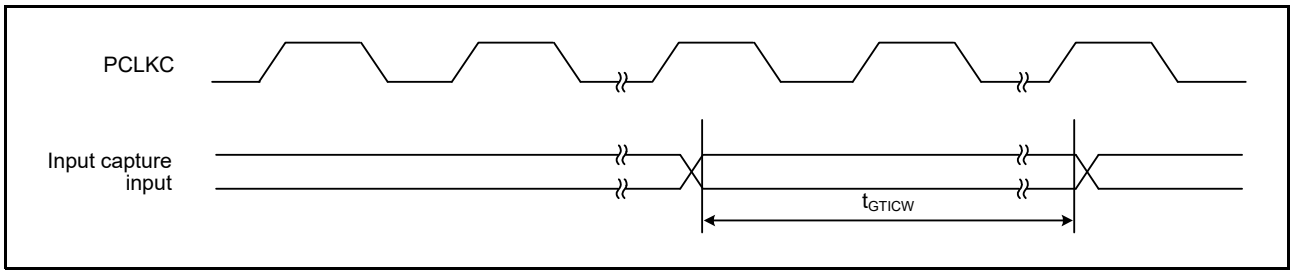


Figure 49.32 GPTW Input Capture Input Timing

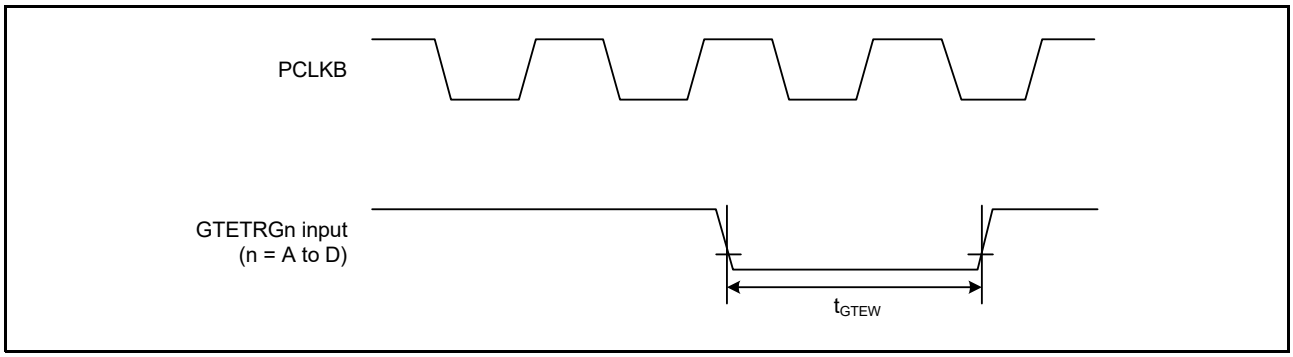


Figure 49.33 GPTW External Trigger Input Timing

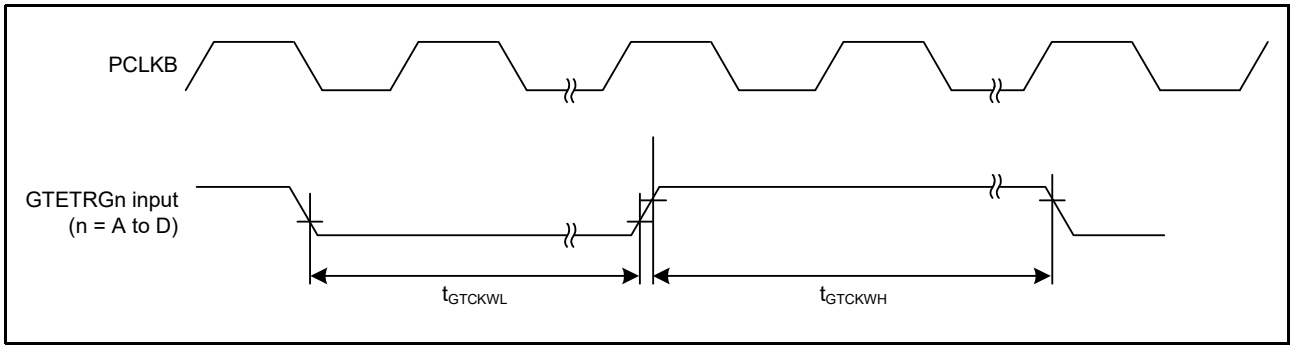


Figure 49.34 GPTW Clock Input Timing

49.4.5.7 A/D Converter Trigger

Table 49.29 A/D Converter Trigger Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = AVCC1 = AVCC2 = 3.0 to 5.5 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V, $T_a = T_{opr}$,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz,
Output load conditions: V_{OH} = 0.5 × VCC, V_{OL} = 0.5 × VCC, C = 30 pF,
High-drive output is selected by the driving ability control register (other than for P53 to P55 and P60 to P65).

Item		Symbol	Min.	Max.	Unit*1	Test Conditions
A/D converter	A/D converter trigger input pulse width	t_{TRGW}	1.5	—	t_{PBcyc}	Figure 49.35

Note 1. t_{PBcyc}: PCLKB cycle

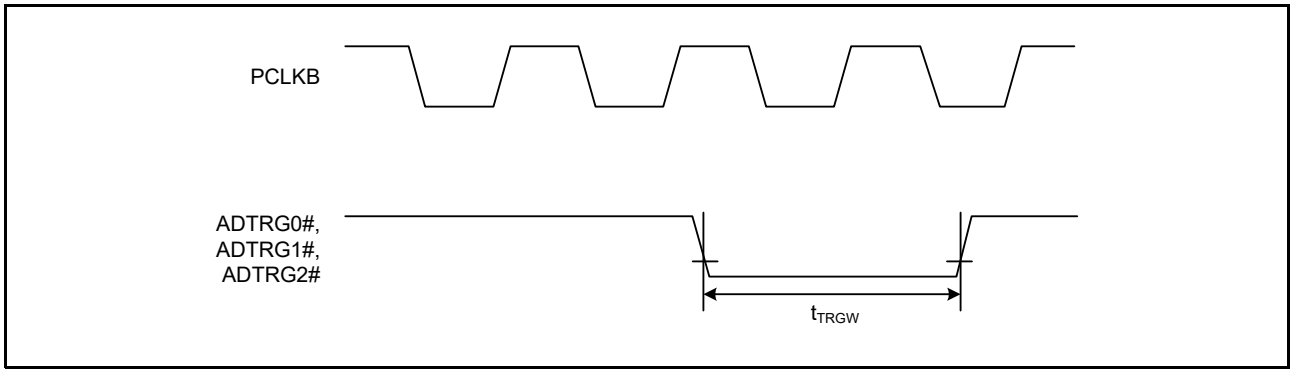


Figure 49.35 A/D Converter Trigger Input Timing

49.4.5.8 CAC

Table 49.30 CAC Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = AVCC1 = AVCC2 = 3.0 to 5.5 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V, $T_a = T_{opr}$,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz,
Output load conditions: V_{OH} = 0.5 × VCC, V_{OL} = 0.5 × VCC, C = 30 pF,
High-drive output is selected by the driving ability control register.

Item*1, *2			Symbol	Min.*1, *2	Max.	Unit	Test Conditions
CAC	CACREF input pulse width	$t_{PBcyc} \le t_{cac}$	t_{CACREF}	$4.5\ t_{cac} + 3\ t_{PBcyc}$	—	ns	
		$t_{PBcyc} > t_{cac}$		$5\ t_{cac} + 6.5\ t_{PBcyc}$	—		

Note 1. t_{PBcyc}: PCLKB cycle
Note 2. t_{cac}: CAC count clock source cycle

49.4.5.9 SCI

Table 49.31 SCIk and SCIh Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = AVCC1 = AVCC2 = 3.0 to 5.5 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V, $T_a = T_{opr}$,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz,
Output load conditions: V_{OH} = 0.5 × VCC, V_{OL} = 0.5 × VCC, C = 30 pF,
High-drive output is selected by the driving ability control register.

Item			Symbol	Min.	Max.	Unit*1	Test Conditions	
SCIk, SCIh	Input clock cycle	Asynchronous	t_{Scyc}	4	—	t_{PBcyc}	Figure 49.36	
		Clock synchronous		6	—			
	Input clock pulse width		t_{SCKW}	0.4	0.6	t_{Scyc}		
	Input clock rise time		t_{SCKr}	—	5	ns		
	Input clock fall time		t_{SCKf}	—	5	ns		
	Output clock cycle	Asynchronous (SCIk)	t_{Scyc}	6	—	t_{PBcyc}		
		Asynchronous (SCIh)		8	—			
		Clock synchronous		4	—			
	Output clock pulse width		t_{SCKW}	0.4	0.6	t_{Scyc}		
	Output clock rise time		t_{SCKr}	—	5	ns		
	Output clock fall time		t_{SCKf}	—	5	ns		
	Transmit data delay time	Clock synchronous	t_{TXD}	—	28	ns	VCC ≥ 4.5 V	Figure 49.37
				—	33		VCC < 4.5 V	
	Receive data setup time	Clock synchronous	t_{RXS}	15	—	ns	Figure 49.37	
	Receive data hold time	Clock synchronous	t_{RXH}	5	—	ns		

Note 1. t_{PBcyc}: PCLKB cycle

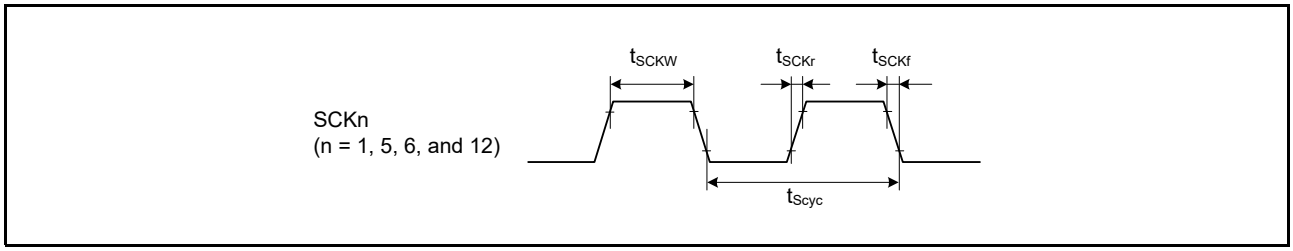


Figure 49.36 SCK Clock Input Timing

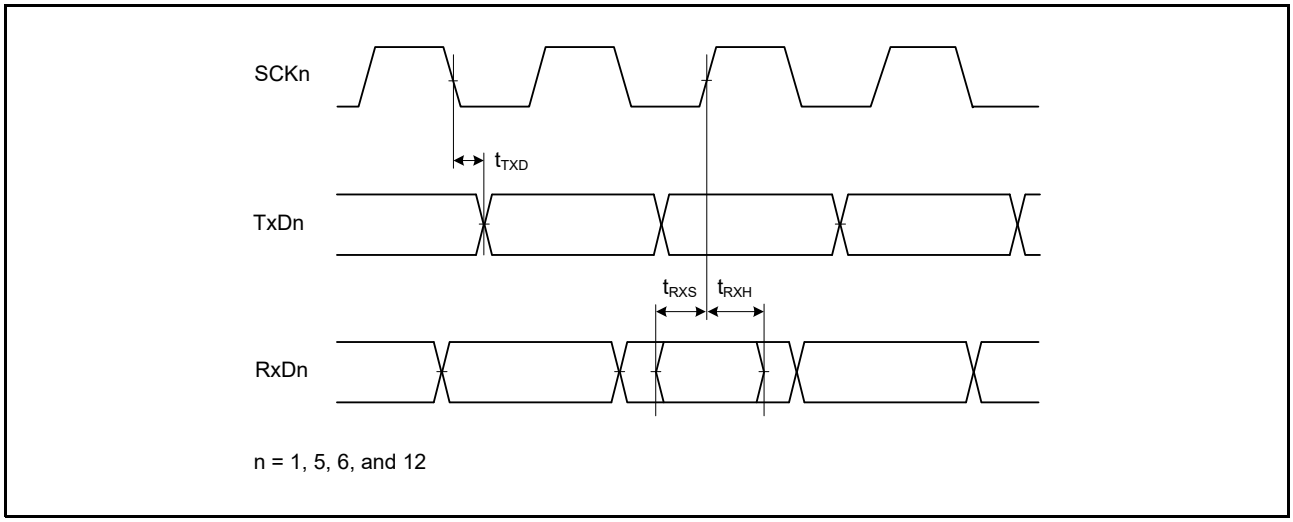


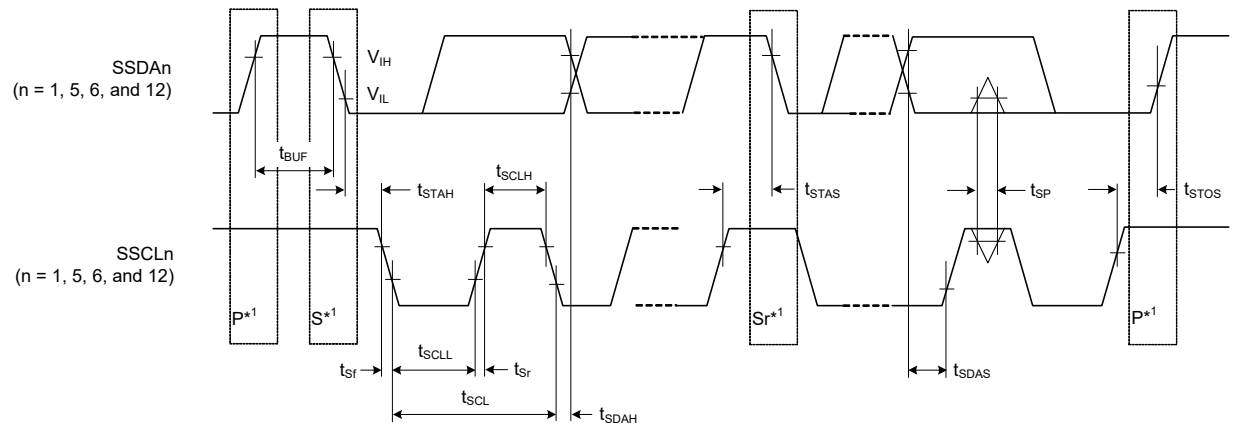
Figure 49.37 SCI Input/Output Timing: Clock Synchronous Mode

Table 49.32 Simple IIC Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = AVCC1 = AVCC2 = 3.0 to 5.5 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V, $T_a = T_{opr}$,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz,
High-drive output is selected by the driving ability control register.

Item		Symbol	Min.	Max.*1	Unit	Test Conditions
Simple IIC (Standard-mode)	SSDA input rise time	t_{Sr}	—	1000	ns	Figure 49.38
	SSDA input fall time	t_{Sf}	—	300		
	SSCL, SSDA input spike pulse removal time	t_{SP}	0	4 × t_{Pcyc}		
	Data input setup time	t_{SDAS}	250	—		
	Data input hold time	t_{SDAH}	0	—		
	SSCL, SSDA capacitive load	C_b*2	—	400	pF	
Simple IIC (Fast-mode)	SSDA input rise time	t_{Sr}	—	300	ns	
	SSDA input fall time	t_{Sf}	—	300		
	SSCL, SSDA input spike pulse removal time	t_{SP}	0	4 × t_{Pcyc}		
	Data input setup time	t_{SDAS}	100	—		
	Data input hold time	t_{SDAH}	0	—		
	SSCL, SSDA capacitive load	C_b*2	—	400	pF	

Note 1. t_{Pcyc} refers to the period of PCLKB.
Note 2. C_b is the total capacitance of the bus lines.



Note 1. S, P, and Sr indicate the following conditions.
S: Start condition
P: Stop condition
Sr: Restart condition

Test conditions
V_{IH} = 0.7 × VCC, V_{IL} = 0.3 × VCC
V_{OL} = 0.6 V, I_{OL} = 6 mA
V_{OL} = 0.4 V, I_{OL} = 3 mA

Figure 49.38 Simple IIC Bus Interface Input/Output Timing

Table 49.33 Simple SPI Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = AVCC1 = AVCC2 = 3.0 to 5.5 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V, $T_a = T_{opr}$,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz,
Output load conditions: V_{OH} = 0.5 × VCC, V_{OL} = 0.5 × VCC, C = 30 pF,
High-drive output is selected by the driving ability control register.

Item		Symbol	Min.	Max.	Unit*1	Test Conditions
Simple SPI	SCK clock cycle output (master)	t_{SPcyc}	4	—	t_{Pcyc}	Figure 49.39
	SCK clock cycle input (slave)		6	—		
	SCK clock high pulse width	t_{SPCKWH}	0.4	0.6	t_{SPcyc}	
	SCK clock low pulse width	t_{SPCKWL}	0.4	0.6	t_{SPcyc}	
	SCK clock rise/fall time	t_{SPCKr}, t_{SPCKf}	—	20	ns	
	Data input setup time	t_{SU}	33.3	—	ns	Figure 49.40 to Figure 49.43
	Data input hold time	t_H	33.3	—	ns	
	SS input setup time	t_{LEAD}	1	—	t_{SPcyc}	
	SS input hold time	t_{LAG}	1	—	t_{SPcyc}	
	Data output delay time	t_{OD}	—	33.3	ns	
	Data output hold time	t_{OH}	–10	—	ns	
	Data rise/fall time	t_{Dr}, t_{Df}	—	16.6	ns	
	SS input rise/fall time	t_{SSLr}, t_{SSLf}	—	16.6	ns	
	Slave access time	t_{SA}	—	5	t_{Pcyc}	Figure 49.42, Figure 49.43
	Slave output release time	t_{REL}	—	5	t_{Pcyc}	

Note 1. t_{Pcyc} refers to the period of PCLKB.

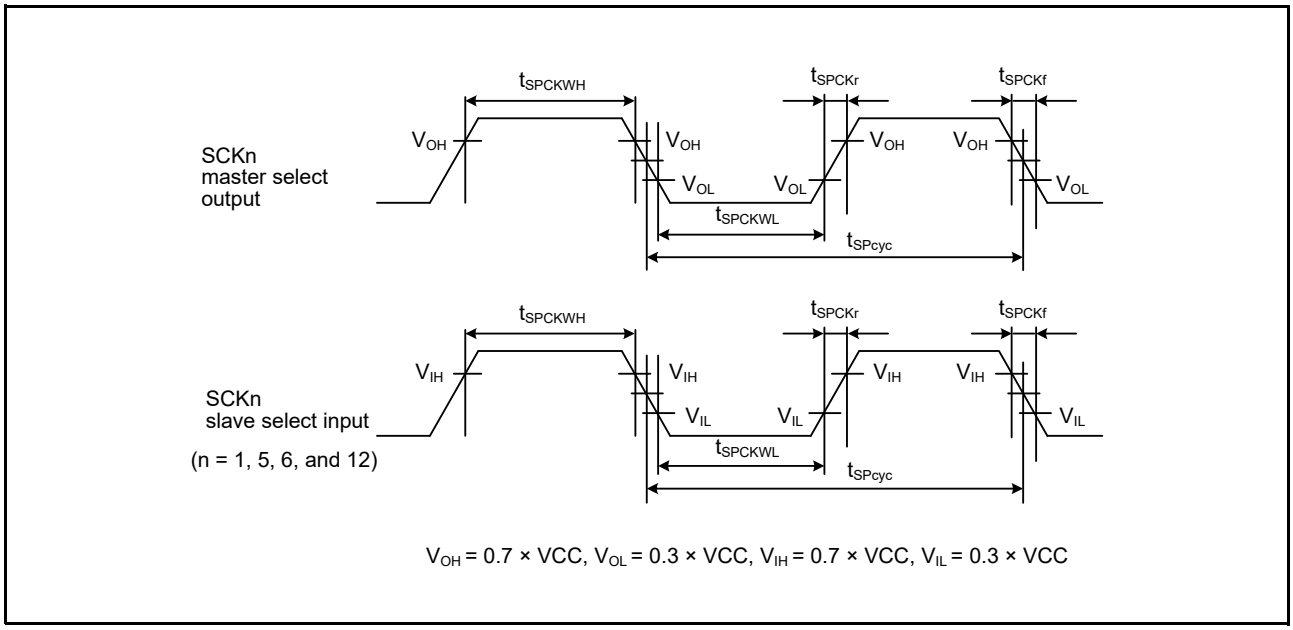


Figure 49.39 Simple SPI Clock Timing

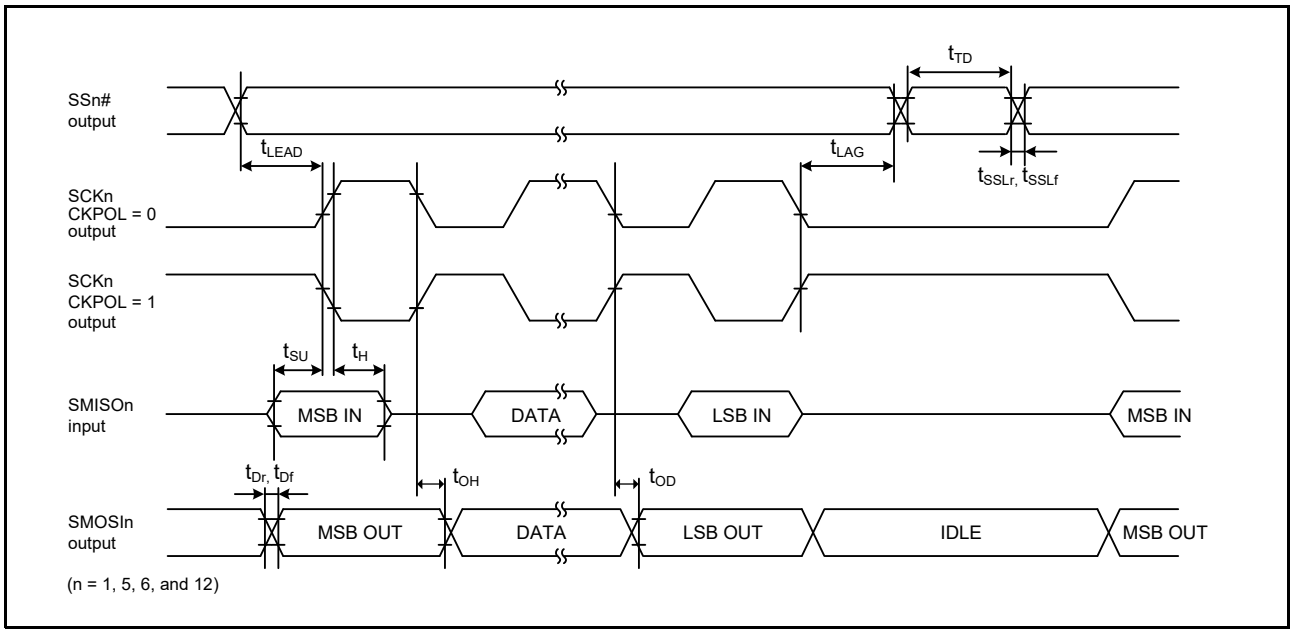


Figure 49.40 Simple SPI Timing (Master, CKPH = 1)

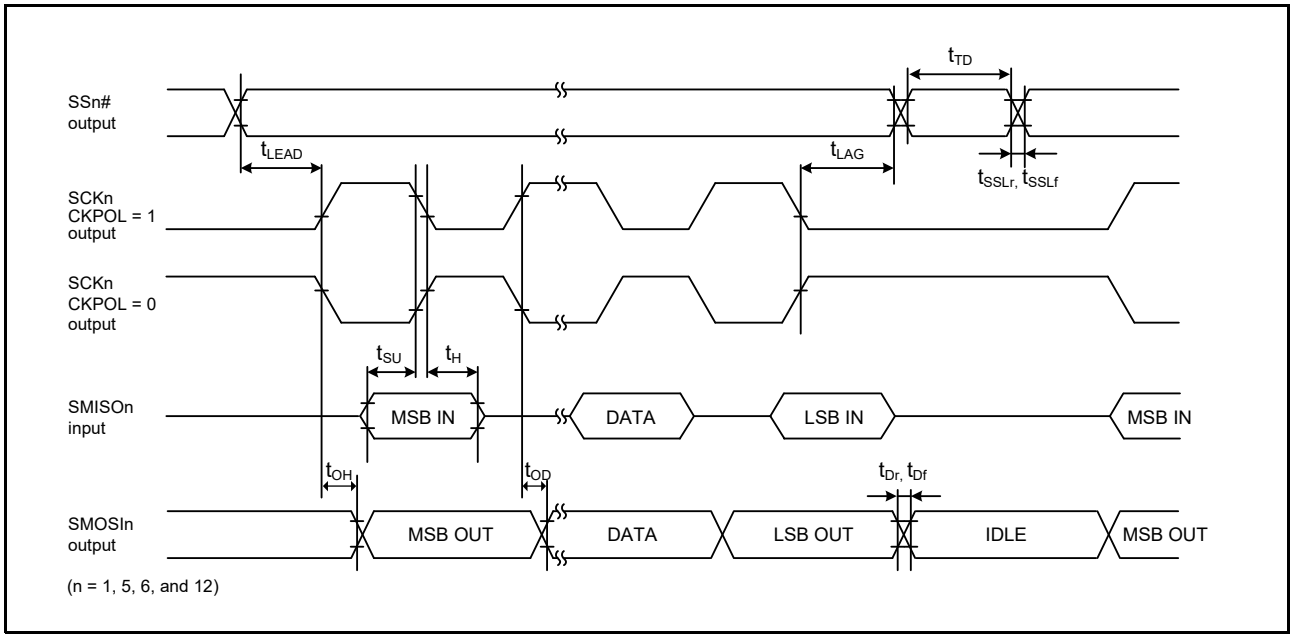


Figure 49.41 Simple SPI Timing (Master, CKPH = 0)

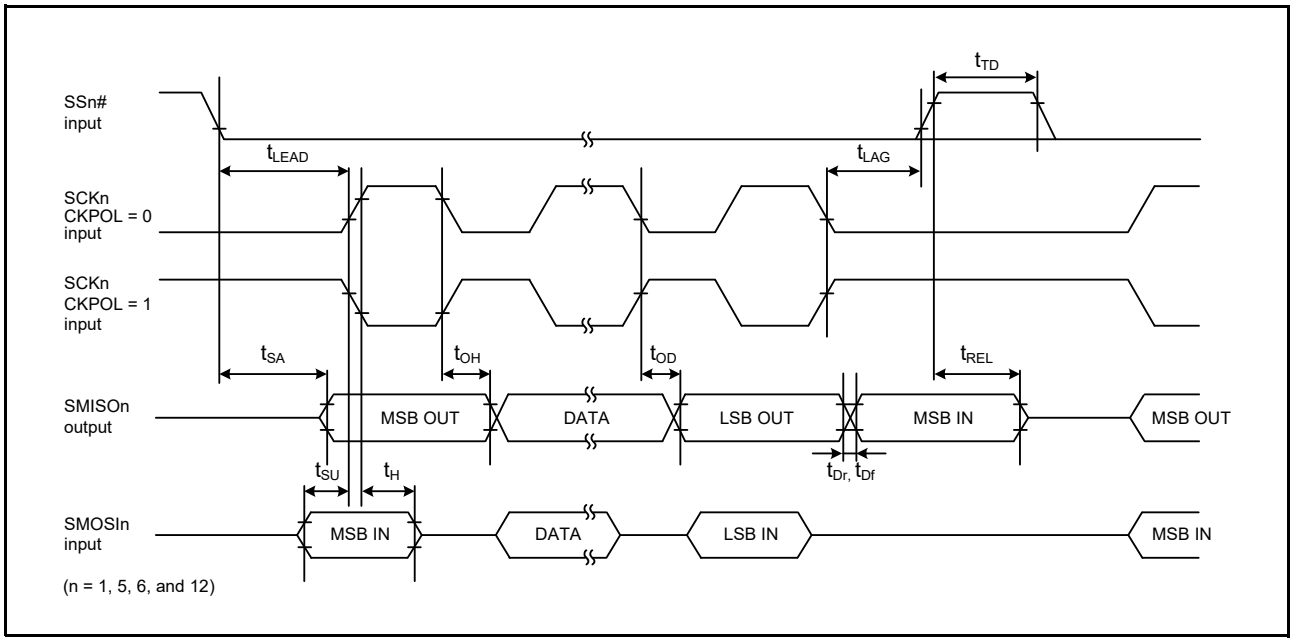


Figure 49.42 Simple SPI Timing (Slave, CKPH = 1)

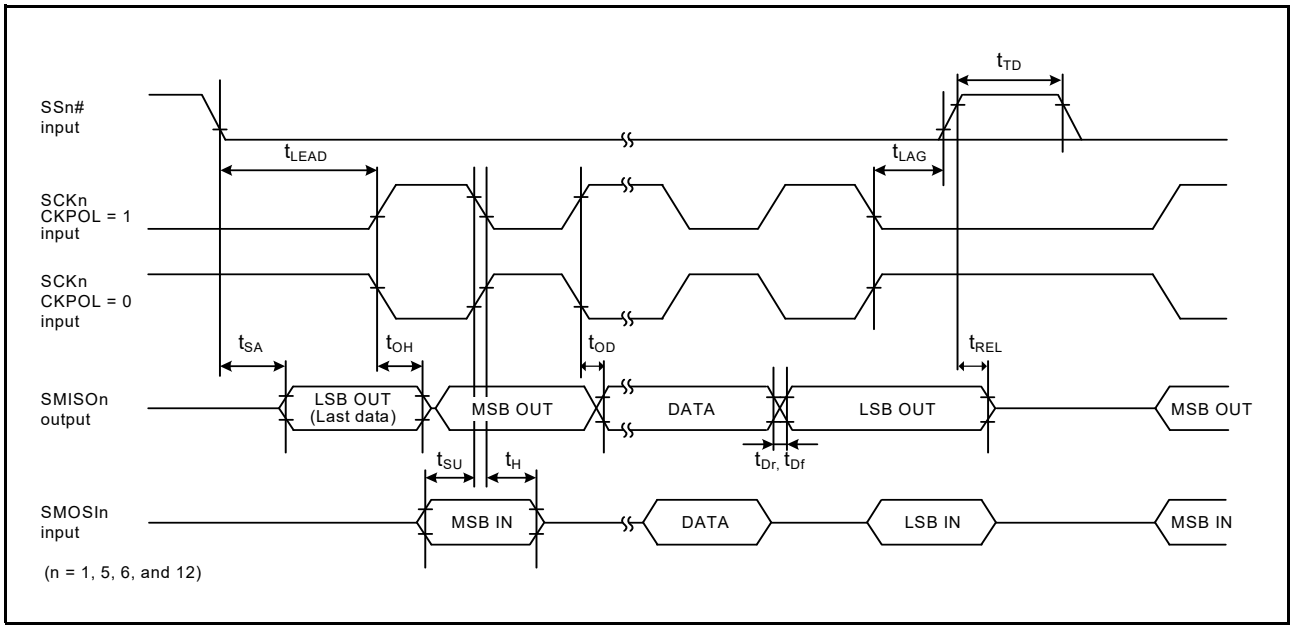


Figure 49.43 Simple SPI Timing (Slave, CKPH = 0)

49.4.5.10 RSCI

Table 49.34 RSCI Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = AVCC1 = AVCC2 = 3.0 to 5.5 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V, $T_a = T_{opr}$,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz,
Output load conditions: V_{OH} = 0.5 × VCC, V_{OL} = 0.5 × VCC, C = 30 pF,
High-drive output is selected by the driving ability control register.

Item			Symbol	Min.	Max.	Unit*1	Test Conditions	
RSCI	Input clock cycle	Asynchronous	t_{Scyc}	4	—	t_{Pcyc}	Figure 49.44	
		Clock synchronous		2	—			
	Input clock pulse width		t_{SCKW}	0.4	0.6	t_{Scyc}		
	Input clock rise time		t_{SCKr}	—	5	ns		
	Input clock fall time		t_{SCKf}	—	5	ns		
	Output clock cycle	Asynchronous	t_{Scyc}	6	—	t_{Pcyc}		
		Clock synchronous		2	—			
	Output clock pulse width		t_{SCKW}	0.4	0.6	t_{Scyc}		
	Output clock rise time		t_{SCKr}	—	5	ns		
	Output clock fall time		t_{SCKf}	—	5	ns		
	Receive data setup time	Master	t_{RXS}	–1.5	—	ns	VCC ≥ 4.5 V	Figure 49.45
				3.5	—		VCC < 4.5 V	
		Slave		2.5	—		Figure 49.45	
	Receive data hold time	Master	t_{RXH}	11	—	ns		
		Slave		2.5	—			
	Transmit data delay time	Master	t_{TXD}	—	4	ns		
		Slave		—	17		VCC ≥ 4.5 V	Figure 49.45
				—	22		VCC < 4.5 V	

Note 1. t_{Pcyc} refers to the period of PCLKB in RSCI8 and RSCI9, and of PCLKA in RSCI11.

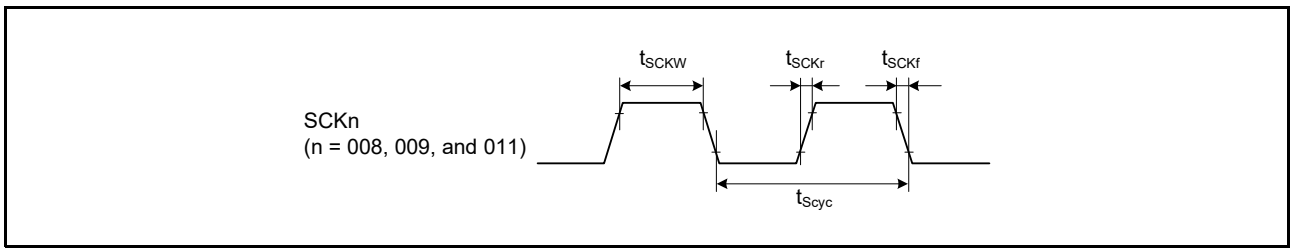


Figure 49.44 SCK Clock Input Timing

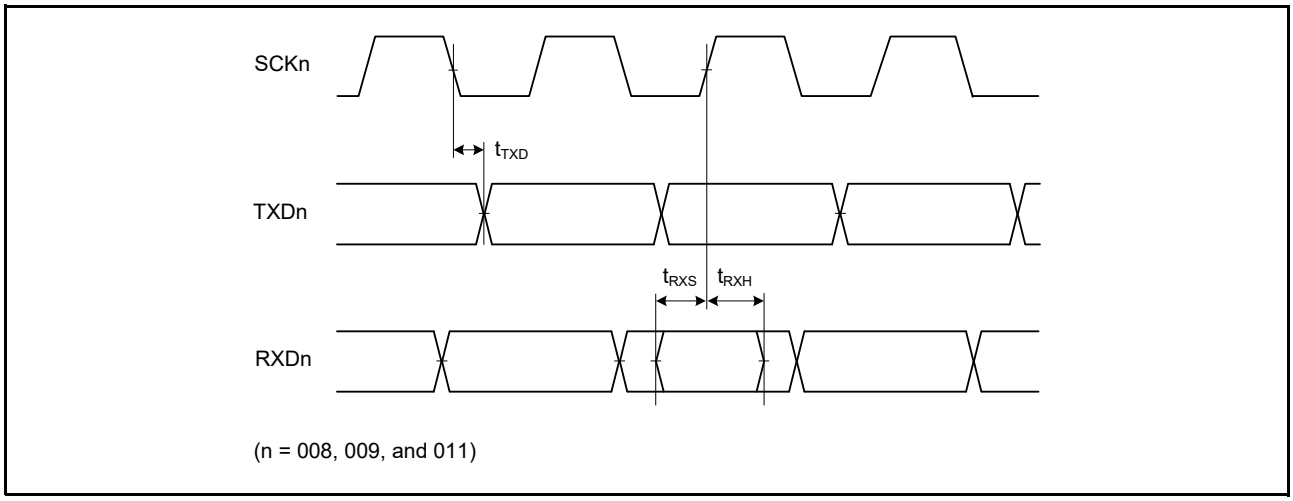


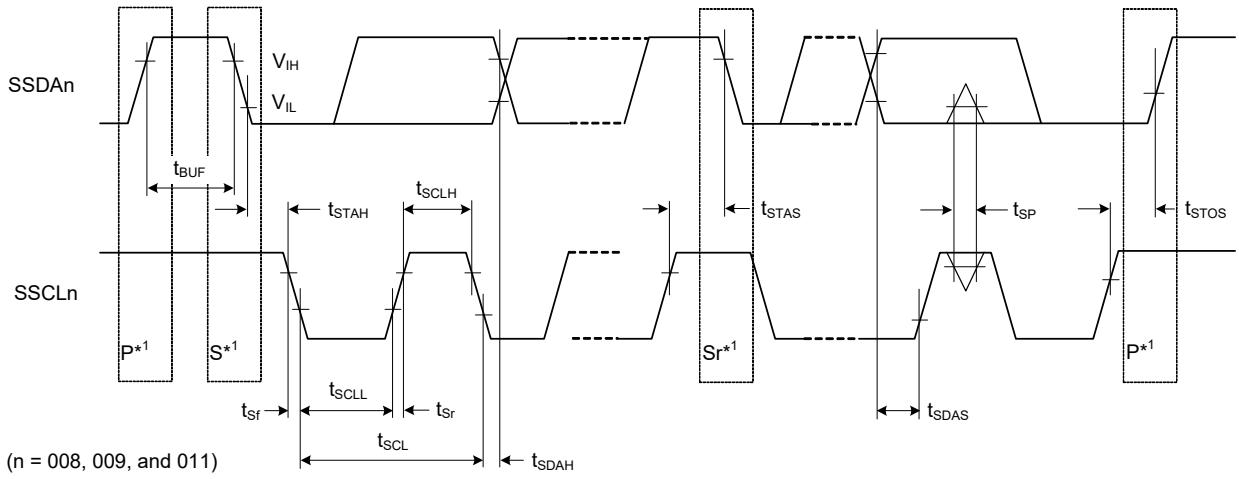
Figure 49.45 RSCI Input/Output Timing: Clock Synchronous Mode

Table 49.35 Simple IIC Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = AVCC1 = AVCC2 = 3.0 to 5.5 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V, $T_a = T_{opr}$,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz,
High-drive output is selected by the driving ability control register.

Item		Symbol	Min.	Max.	Unit	Test Conditions
Simple IIC (Standard-mode)	SSCL, SSDA input rise time	t_{Sr}	—	1000	ns	Figure 49.46
	SSCL, SSDA input fall time	t_{Sf}	—	300	ns	
	SSCL, SSDA input spike pulse removal time	t_{SP}	0	4 × t_{Pcyc}	ns	
	Data input setup time	t_{SDAS}	250	—	ns	
	Data input hold time	t_{SDAH}	0	—	ns	
	SSCL, SSDA capacitive load	C_b*1	—	400	pF	
Simple IIC (Fast-mode)	SSCL, SSDA input rise time	t_{Sr}	—	300	ns	
	SSCL, SSDA input fall time	t_{Sf}	—	300	ns	
	SSCL, SSDA input spike pulse removal time	t_{SP}	0	4 × t_{Pcyc}	ns	
	Data input setup time	t_{SDAS}	100	—	ns	
	Data input hold time	t_{SDAH}	0	—	ns	
	SSCL, SSDA capacitive load	C_b*1	—	400	pF	

Note: t_{Pcyc} refers to the period of PCLKB in RSCI8 and RSCI9, and of PCLKA in RSCI11.
Note 1. C_b is the total capacitance of the bus lines.



Note 1. S, P, and Sr indicate the following conditions.
S: Start condition
P: Stop condition
Sr: Restart condition

Test conditions
V_{IH} = 0.7 × VCC, V_{IL} = 0.3 × VCC
V_{OL} = 0.6 V, I_{OL} = 6 mA

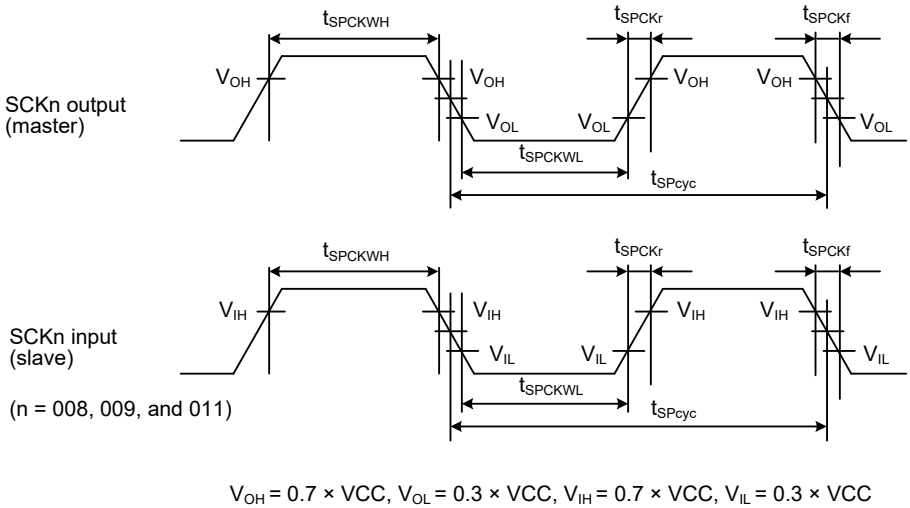
Figure 49.46 Simple IIC Bus Interface Input/Output Timing

Table 49.36 Simple SPI Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = AVCC1 = AVCC2 = 3.0 to 5.5 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V, $T_a = T_{opr}$,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz,
Output load conditions: V_{OH} = 0.5 × VCC, V_{OL} = 0.5 × VCC, C = 30 pF,
High-drive output is selected by the driving ability control register.

Item			Symbol	Min.	Max.	Unit*1	Test Conditions
Simple SPI	SCK clock cycle output (master)		t_{SPcyc}	2	—	t_{Pcyc}	Figure 49.47
	SCK clock cycle input (slave)			2	—		
	SCK clock high pulse width		t_{SPCKWH}	0.4	0.6	t_{SPcyc}	
	SCK clock low pulse width		t_{SPCKWL}	0.4	0.6	t_{SPcyc}	
	SCK clock rise/fall time	Output	t_{SPCKr}, t_{SPCKf}	—	5	ns	
		Input		—	1	μs	
	Data input setup time	Master	t_{SU}	0.5	—	ns	Figure 49.48 to Figure 49.51
		Slave		2.5	—		
	Data input hold time	Master	t_H	11	—	ns	
		Slave		2.5	—		
	Data output delay time	Master	t_{OD}	—	4	ns	Figure 49.48 to Figure 49.51
		Slave		—	17		VCC ≥ 4.5V: Figure 49.48 to Figure 49.51
				—	22		VCC < 4.5V: Figure 49.48 to Figure 49.51
	Data output hold time	Master	t_{OH}	–1	—	ns	Figure 49.48 to Figure 49.51
		Slave		0	—		
	Data rise/fall time	Output	t_{Dr}, t_{Df}	—	5	ns	
		Input		—	1	—	
	Slave access time		t_{SA}	—	5	t_{Pcyc}	Figure 49.50, Figure 49.51
	Slave output release time		t_{REL}	—	5	t_{Pcyc}	
	SS input setup time		t_{LEAD}	1	—	t_{SPcyc}	Figure 49.48 to Figure 49.51
	SS input hold time		t_{LAG}	1	—	t_{SPcyc}	
	SS input rise/fall time		t_{SSLr}, t_{SSLf}	—	1	μs	

Note 1. t_{Pcyc} refers to the period of PCLKB in RSCI8 and RSCI9, and of PCLKA in RSCI11.



V_{OH} = 0.7 × VCC, V_{OL} = 0.3 × VCC, V_{IH} = 0.7 × VCC, V_{IL} = 0.3 × VCC

Figure 49.47 Simple SPI Clock Timing

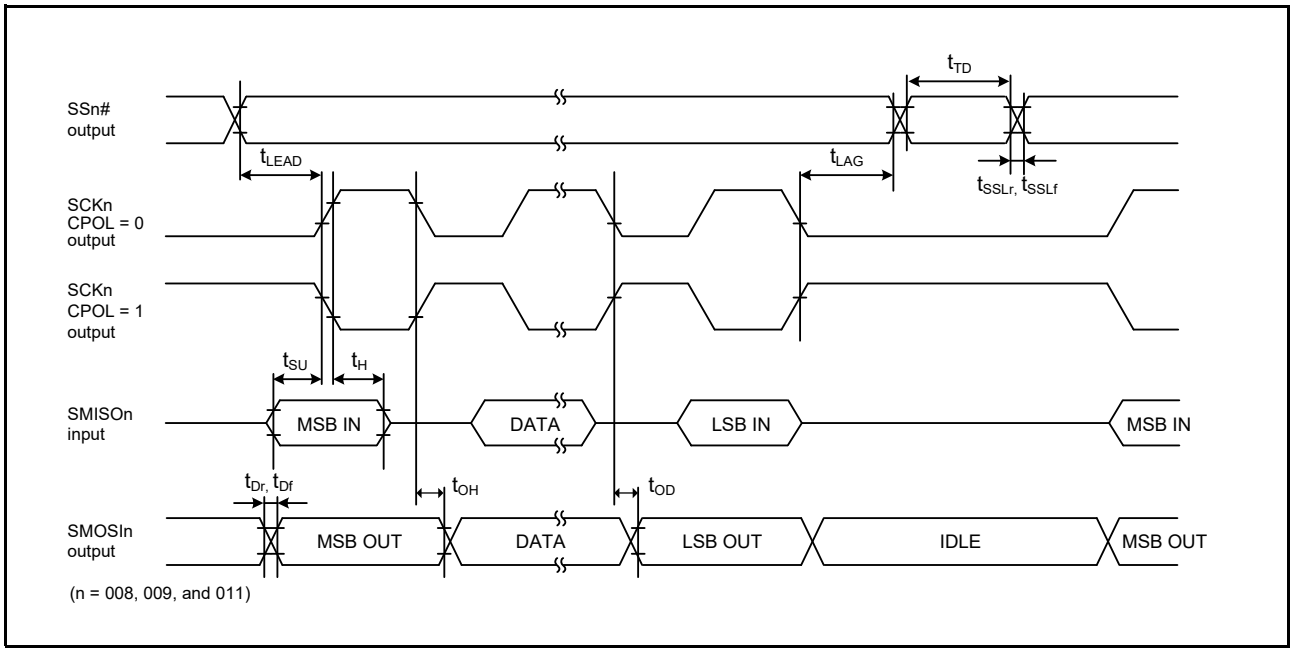


Figure 49.48 Simple SPI Timing (Master, CPHA = 0)

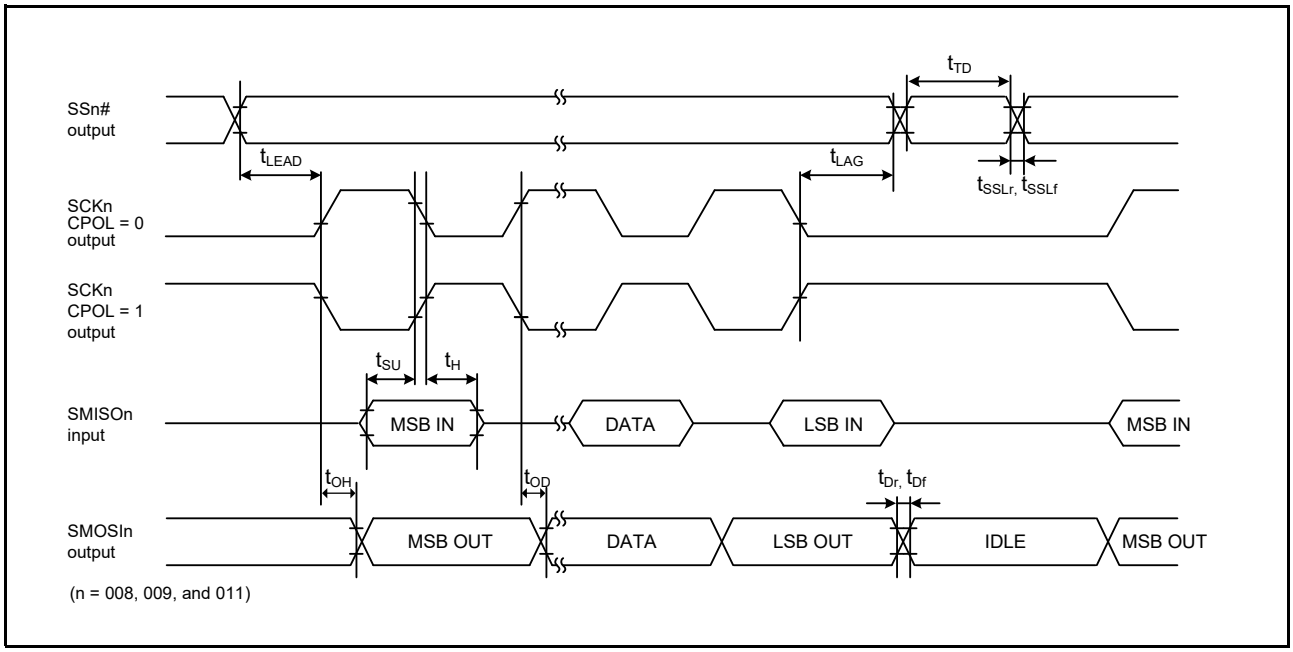


Figure 49.49 Simple SPI Timing (Master, CPHA = 1)

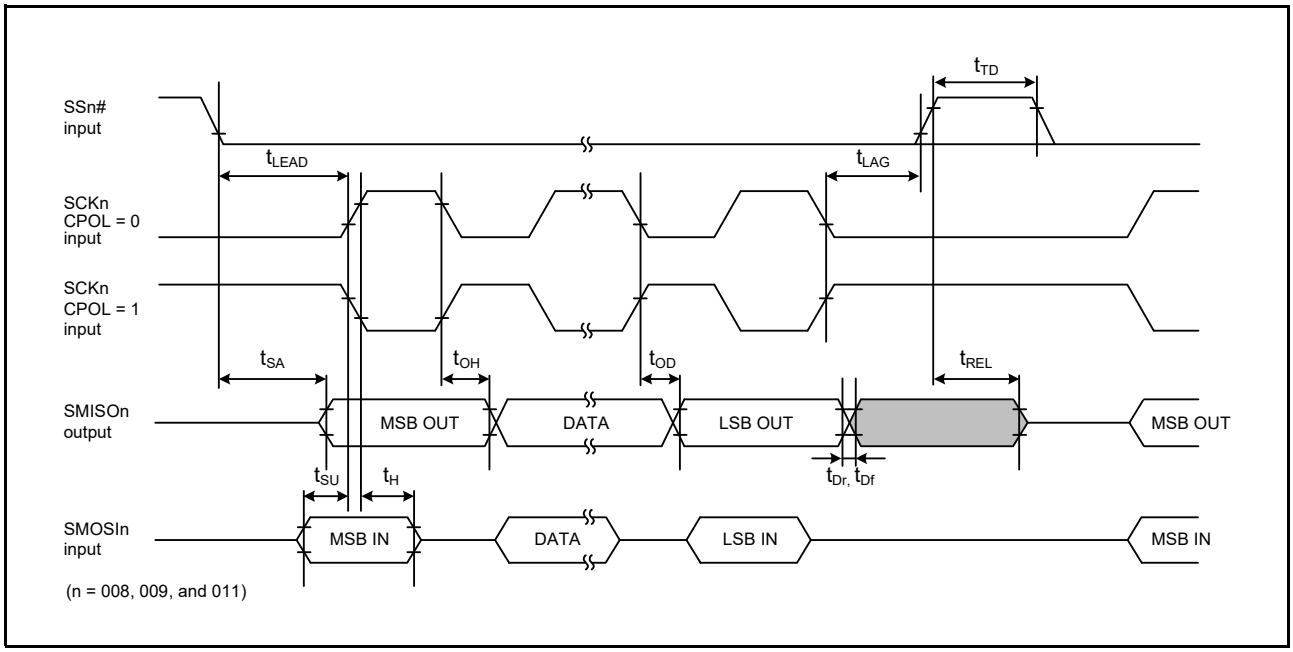


Figure 49.50 Simple SPI Timing (Slave, CPHA = 0)

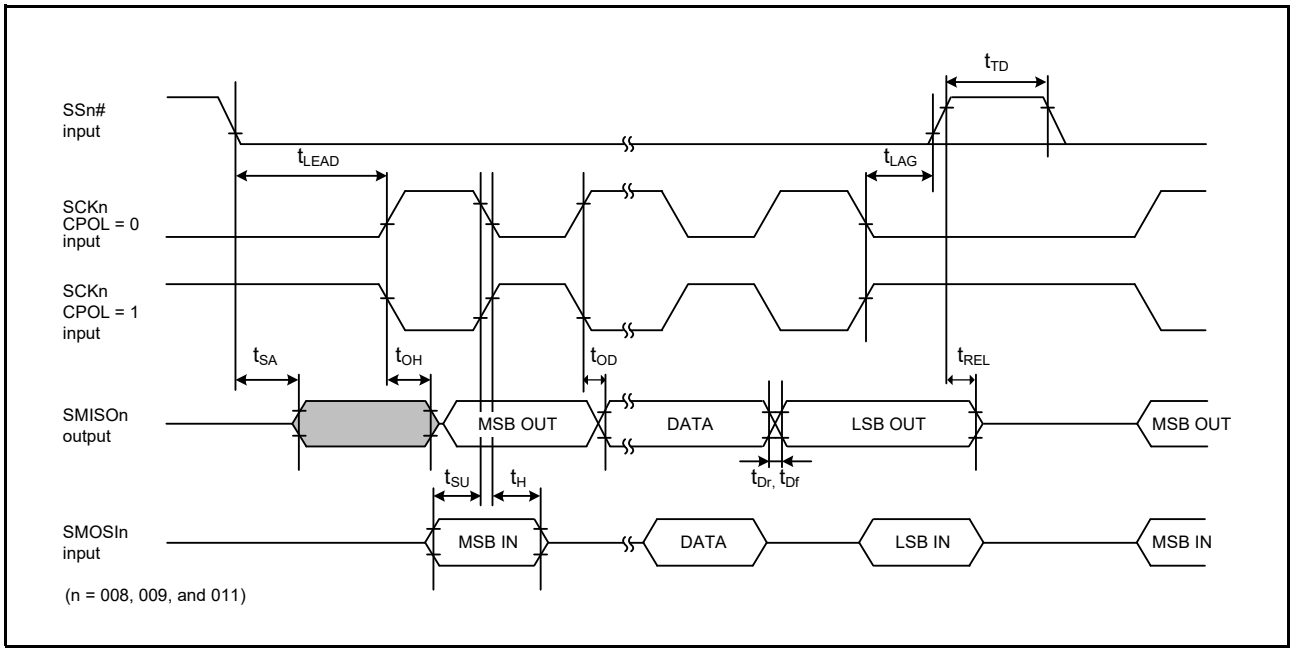


Figure 49.51 Simple SPI Timing (Slave, CPHA = 1)

49.4.5.11 RSPI

Table 49.37 RSPI Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = AVCC1 = AVCC2 = 3.0 to 5.5 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V, $T_a = T_{opr}$,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz,
Output load conditions: V_{OH} = 0.5 × VCC, V_{OL} = 0.5 × VCC, C = 30 pF,
High-drive output is selected by the driving ability control register.

Item				Symbol	Min.*1	Max.*1	Unit*1	Test Conditions	
RSPI	RSPCK clock cycle	Master		t_{SPcyc}	2	—	t_{PAcyc}	Figure 49.52	
		Slave			4	—			
	RSPCK clock high pulse width	Master		t_{SPCKWH}	$(t_{SPcyc} - t_{SPCKr} - t_{SPCKf}) / 2 - 3$	—	ns		
		Slave			0.4	0.6	t_{SPcyc}		
	RSPCK clock low pulse width	Master		t_{SPCKWL}	$(t_{SPcyc} - t_{SPCKr} - t_{SPCKf}) / 2 - 3$	—	ns		
		Slave			0.4	0.6	t_{SPcyc}		
	RSPCK clock rise/fall time	Output		t_{SPCKr}, t_{SPCKf}	—	5	ns		
		Input			—	1	µs		
	Data input setup time	Master		t_{SU}	6	—	ns	VCC ≥ 4.5 V	Figure 49.53 to Figure 49.58
					11	—		VCC < 4.5 V	
		Slave			8.3	—		Figure 49.53 to Figure 49.58	
	Data input hold time	Master	PCLKA division ratio set to 1/2	t_{HF}	0	—	ns		
			PCLKA division ratio set to a value other than 1/2	t_H	t_{PAcyc}	—			
		Slave			8.3	—			
	SSL setup time	Master		t_{LEAD}	1	8	t_{SPcyc}		
		Slave			4	—	t_{PAcyc}		
	SSL hold time	Master		t_{LAG}	1	8	t_{SPcyc}		
		Slave			4	—	t_{PAcyc}		
	Data output delay time	Master		t_{OD}	—	6.3	ns	VCC ≥ 4.5 V	Figure 49.53 to Figure 49.58
					—	11.3		VCC < 4.5 V	
		Slave			—	28		VCC ≥ 4.5 V	
					—	33		VCC < 4.5 V	
	Data output hold time	Master		t_{OH}	0	—	ns	Figure 49.53 to Figure 49.58	
		Slave			0	—			
	Successive transmission delay time	Master		t_{TD}	$t_{SPcyc} + 2 \times t_{PAcyc}$	$8 \times t_{SPcyc} + 2 \times t_{PAcyc}$	ns		
		Slave			$4 \times t_{PAcyc}$	—			
	MOSI and MISO rise/fall time	Output		t_{Dr}, t_{Df}	—	5	ns		
		Input			—	1	µs		
	SSL rise/fall time	Output		t_{SSLr}, t_{SSLf}	—	5	ns		
		Input			—	1	µs		
	Slave access time			t_{SA}	—	28	ns	VCC ≥ 4.5 V	Figure 49.57, Figure 49.58
					—	33		VCC < 4.5 V	
	Slave output release time			t_{REL}	—	28	ns	VCC ≥ 4.5 V	
					—	33		VCC < 4.5 V	

Note 1. t_{PAcyc}: PCLKA cycle

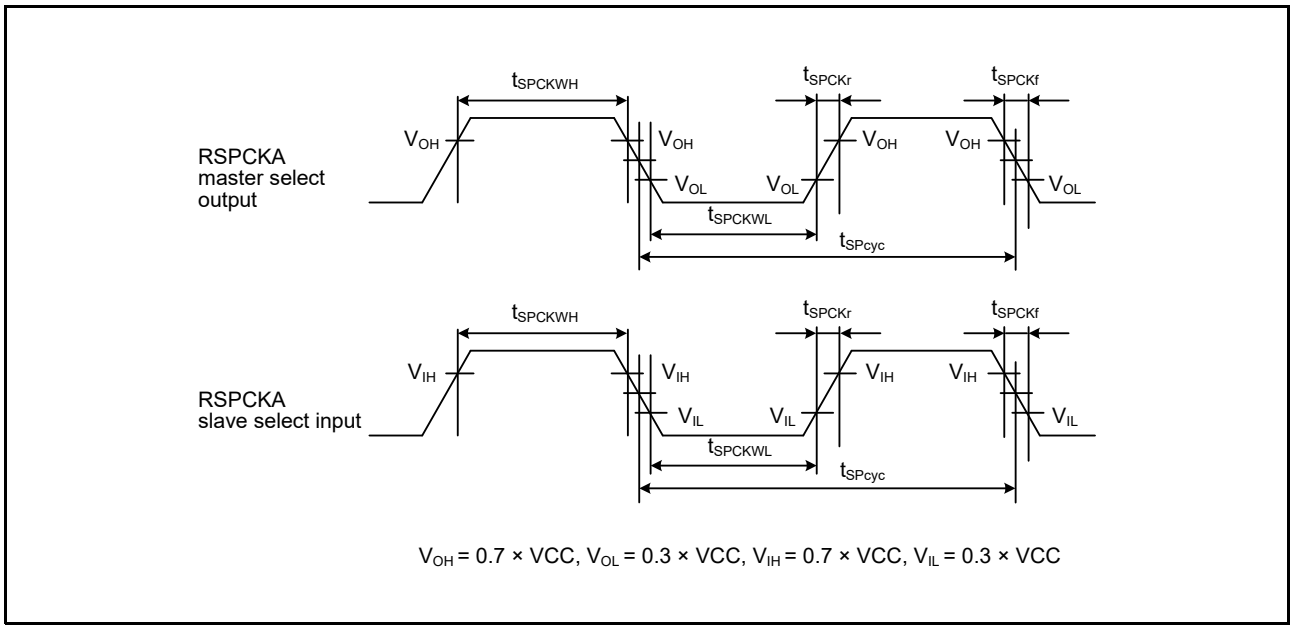


Figure 49.52 RSPI Clock Timing

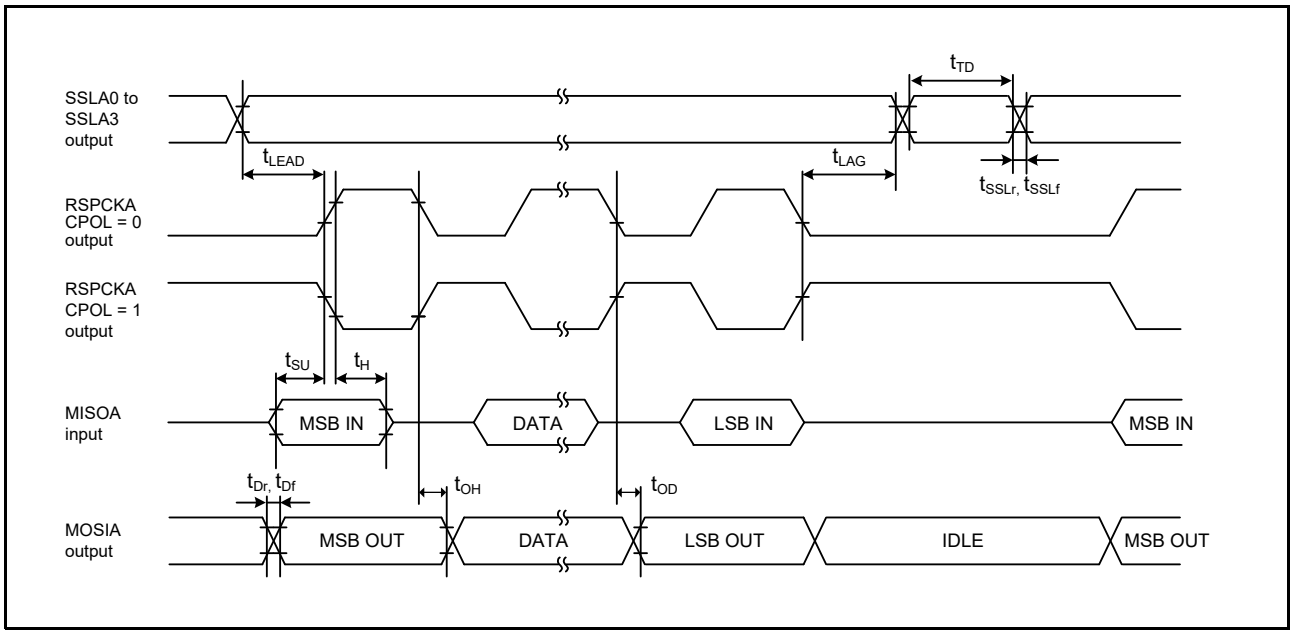


Figure 49.53 RSPI Timing (Master, CPHA = 0) (Bit Rate: PCLKA Division Ratio Set to a Value Other Than 1/2)

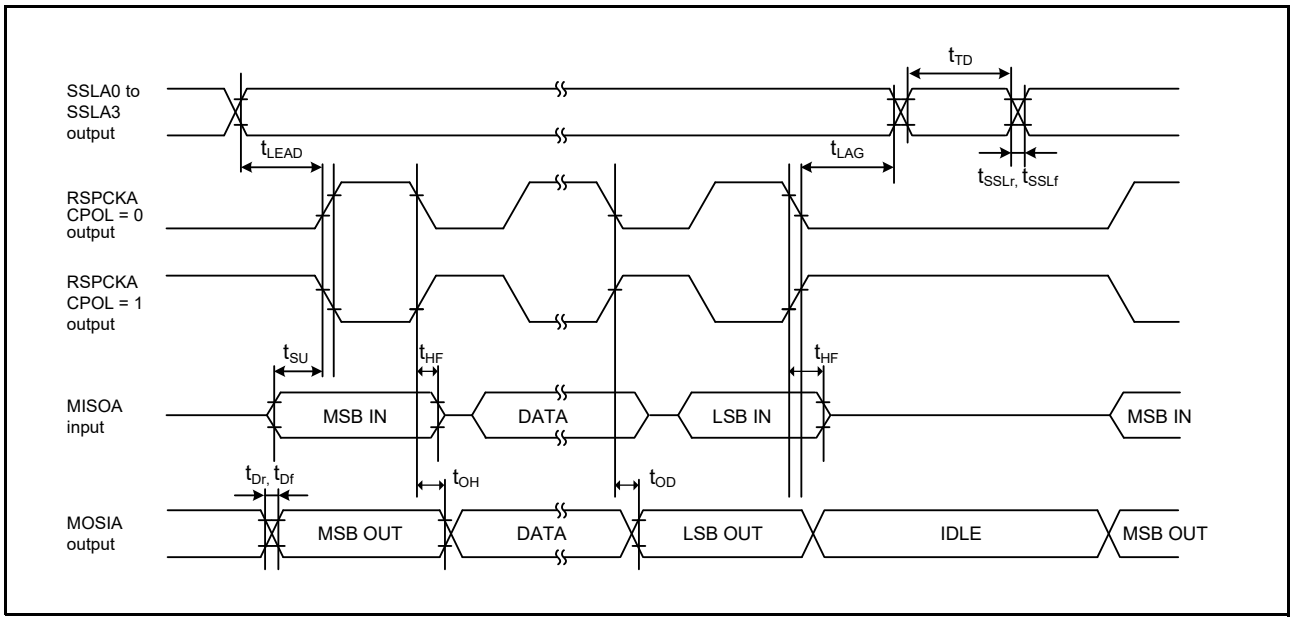


Figure 49.54 RSPI Timing (Master, CPHA = 0) (Bit Rate: PCLKA Division Ratio Set to 1/2)

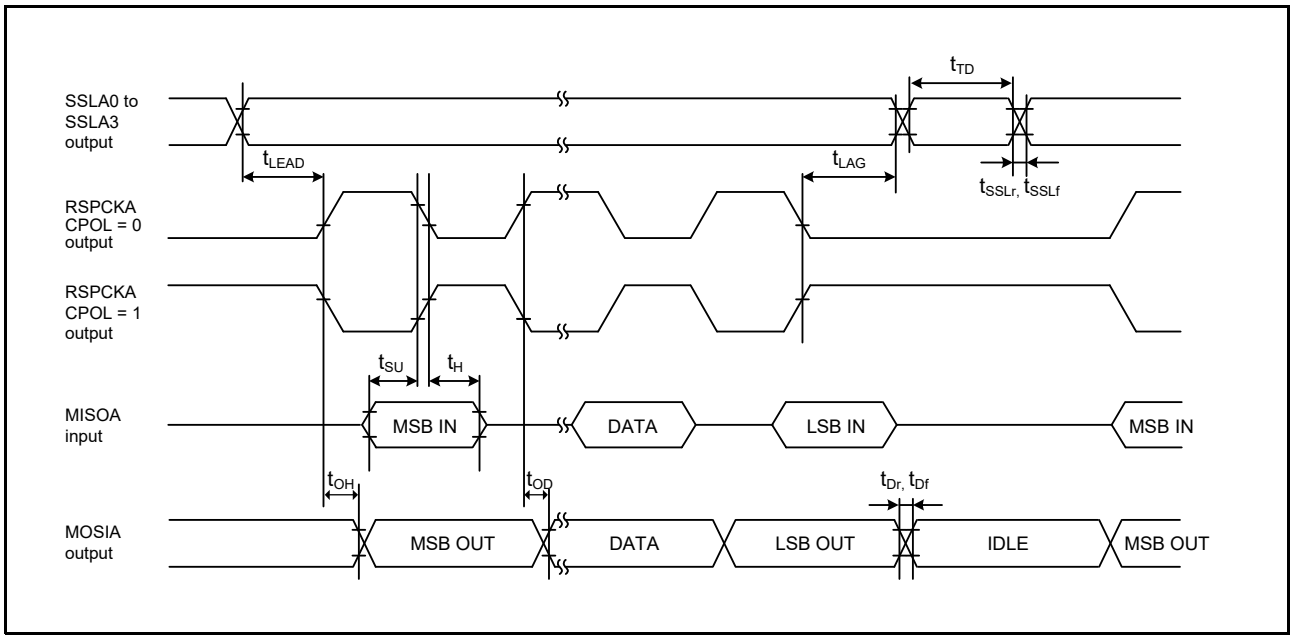


Figure 49.55 RSPI Timing (Master, CPHA = 1) (Bit Rate: PCLKA Division Ratio Set to a Value Other Than 1/2)

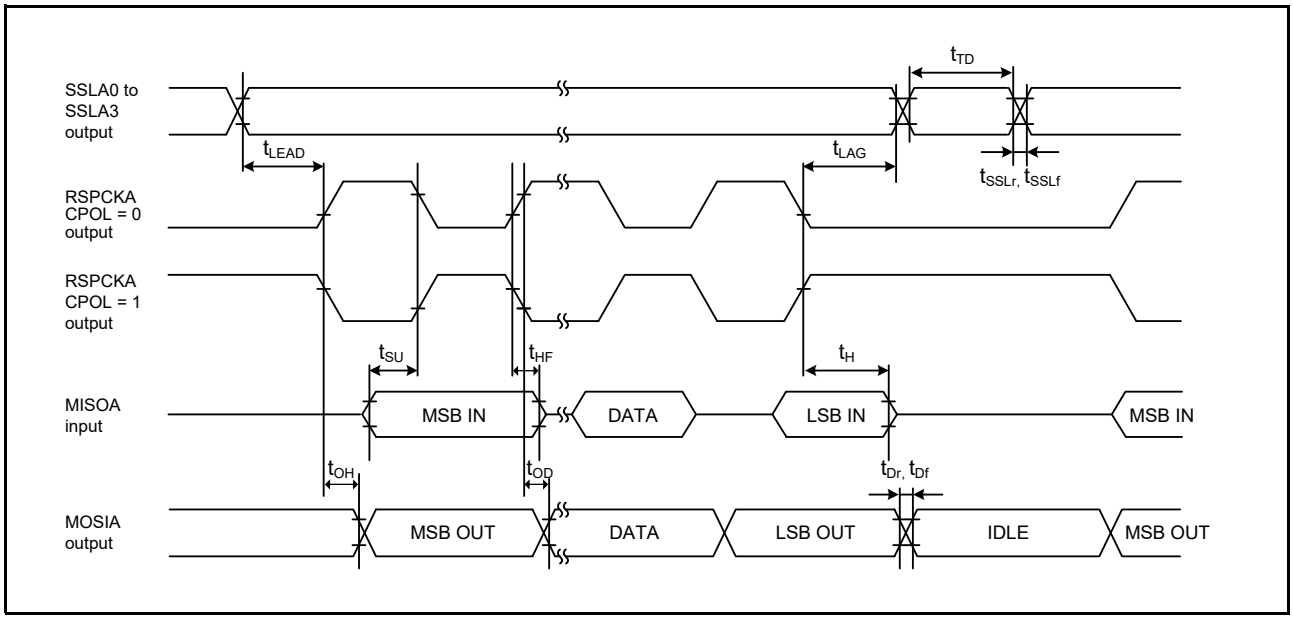


Figure 49.56 RSPI Timing (Master, CPHA = 1) (Bit Rate: PCLKA Division Ratio Set to 1/2)

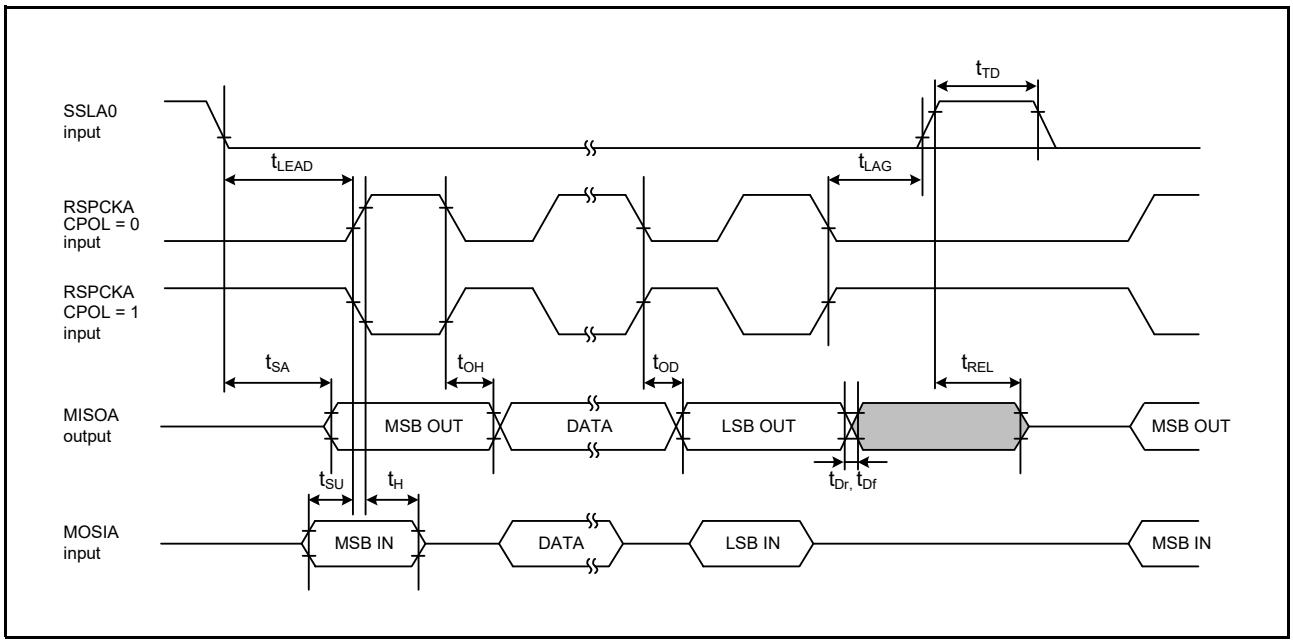


Figure 49.57 RSPI Timing (Slave, CPHA = 0)

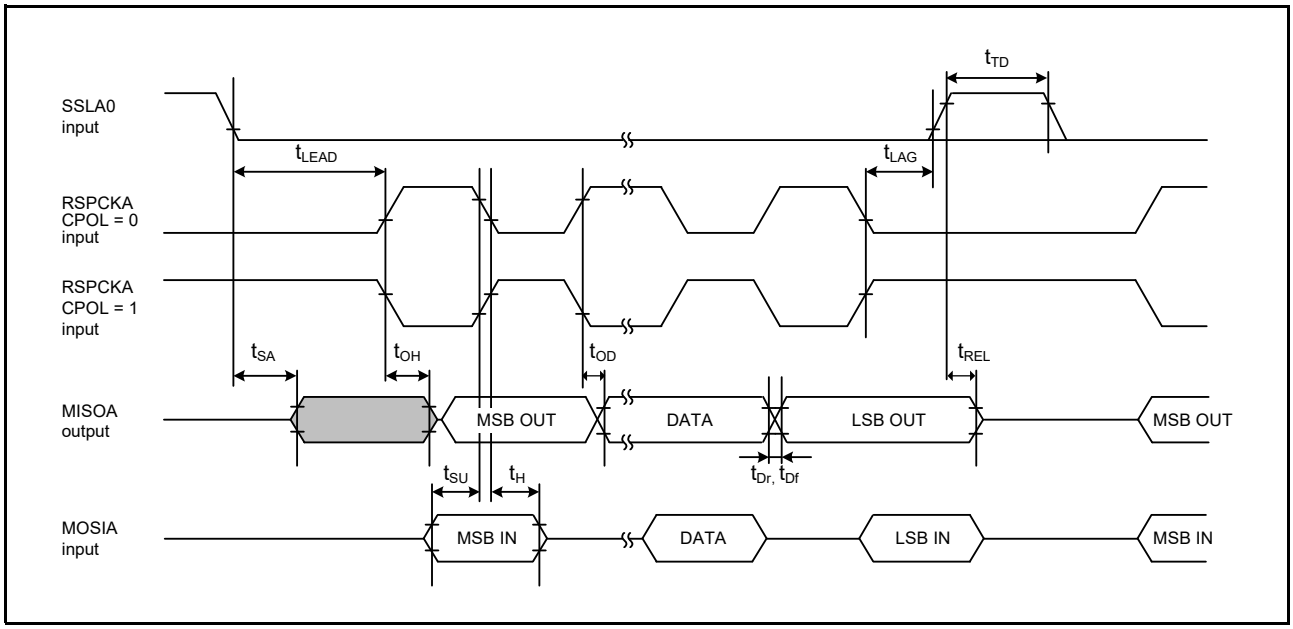


Figure 49.58 RSPI Timing (Slave, CPHA = 1)

49.4.5.12 RSPIA

Table 49.38 RSPIA Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = AVCC1 = AVCC2 = 3.0 to 5.5 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V, $T_a = T_{opr}$,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz,
Output load conditions: V_{OH} = 0.5 × VCC, V_{OL} = 0.5 × VCC, C = 15 pF,
High-drive output is selected by the driving ability control register.

Item			Symbol	Min.*1	Max.*1	Unit*1	Test Conditions	
RSPI	RSPCK clock cycle	Master	t_{SPcyc}	2	—	t_{PAcyc}	Figure 49.59	
		Slave		2	—			
	RSPCK clock high pulse width	Master	t_{SPCKWH}	$(t_{SPcyc} - t_{SPCKr} - t_{SPCKf}) / 2 - 3$	—	ns		
		Slave		0.4	0.6	t_{SPcyc}		
	RSPCK clock low pulse width	Master	t_{SPCKWL}	$(t_{SPcyc} - t_{SPCKr} - t_{SPCKf}) / 2 - 3$	—	ns		
		Slave		0.4	0.6	t_{SPcyc}		
	RSPCK clock rise/fall time	Output	t_{SPCKr}, t_{SPCKf}	—	5	ns		
		Input		—	1	µs		
	Data input setup time	Master	t_{SU}	0	—	ns	VCC ≥ 4.5 V	Figure 49.60 to Figure 49.66
				2.5	—		VCC < 4.5 V	
		Slave		2.5	—			
	Data input hold time	Master	t_H	7	—	ns		
		Slave		2.5	—			
	SSL setup time	Master	t_{LEAD}	1	8	t_{SPcyc}		
		Slave		6	—	t_{PAcyc}		
	SSL hold time	Master	t_{LAG}	1	8	t_{SPcyc}		
		Slave		6	—	t_{PAcyc}		
	Data output delay time	Master	t_{OD}	—	4.5	ns	VCC ≥ 4.5 V	
				—	5.5		VCC < 4.5 V	
		Slave		—	14		VCC ≥ 4.5 V	
				—	18		VCC < 4.5 V	
	Data output hold time	Master	t_{OH}	0	—	ns		
		Slave		0	—			
	Successive transmission delay time	Master	t_{TD}	$t_{SPcyc} + 2 \times t_{PAcyc}$	$8 \times t_{SPcyc} + 2 \times t_{PAcyc}$	ns		
		Slave		t_{SPcyc}	—			
	MOSI and MISO rise/fall time	Output	t_{Dr}, t_{Df}	—	5	ns		
		Input		—	1	µs		
	SSL rise/fall time	Output	t_{SSLr}, tSSLf	—	5	ns		
		Input		—	1	µs		
	Slave access time		t_{SA}	—	20	ns	Figure 49.63, Figure 49.64	
	Slave output release time		t_{REL}	—	20	ns		
	TI SSP SS input setup time	Slave	t_{TISS}	4.5	—	ns	Figure 49.65, Figure 49.66	
	TI SSP SS input hold time	Slave	t_{TISH}	2.5	—	ns		
	TI SSP next-access delay time	Slave	t_{TIND}	$2 \times t_{PAcyc}$ + SLNDL × t_{PAcyc}	—	ns		
	TI SSP SS output delay time	Master	t_{TISSOD}	—	7	ns	Figure 49.62	

Note 1. t_{PAcyc}: PCLKA cycle

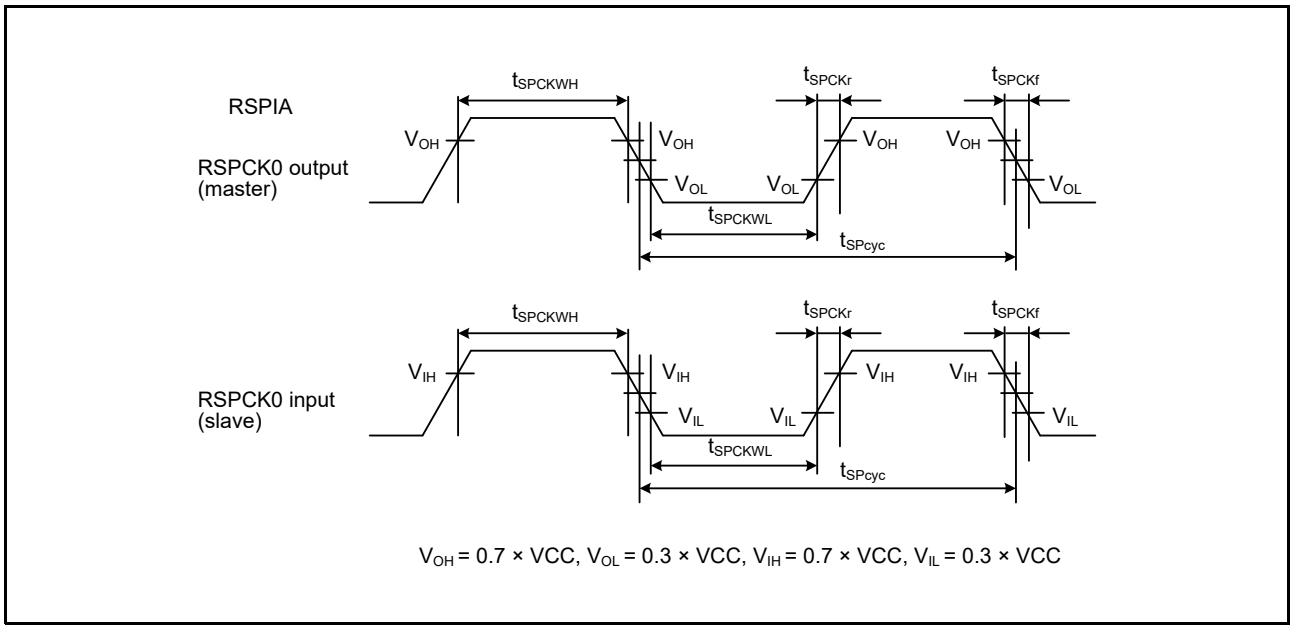


Figure 49.59 RSPIA Clock Timing

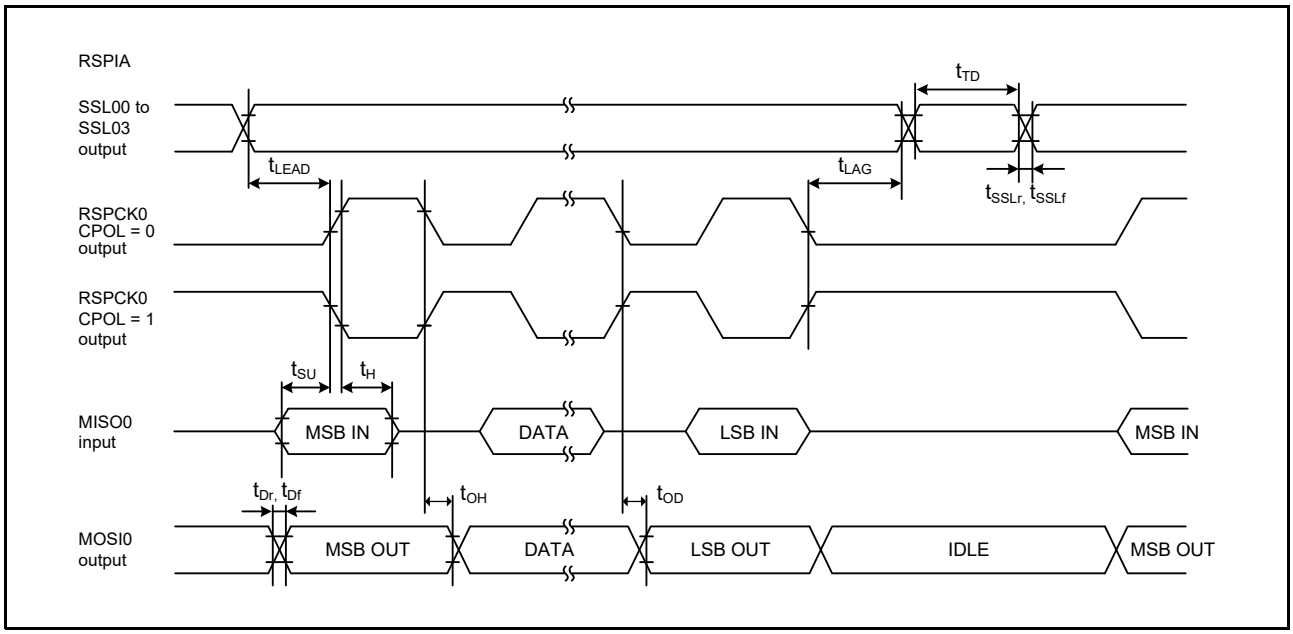


Figure 49.60 RSPIA Timing (Master, Motorola SPI, CPHA = 0)

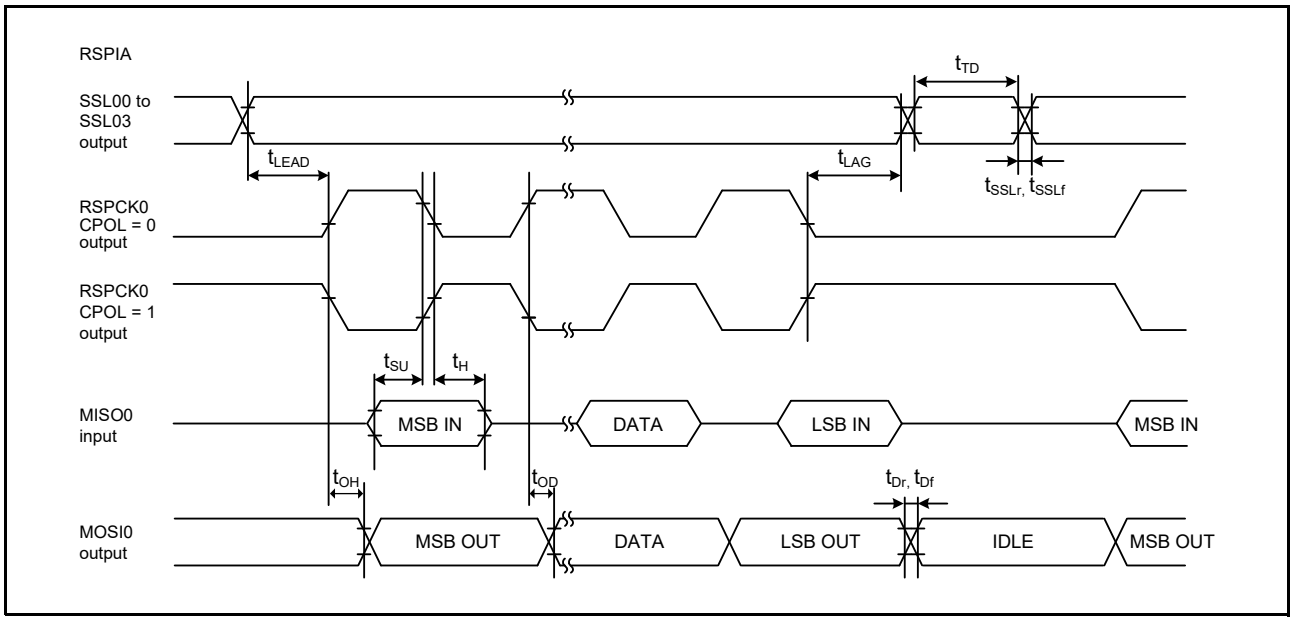


Figure 49.61 RSPIA Timing (Master, Motorola SPI, CPHA = 1)

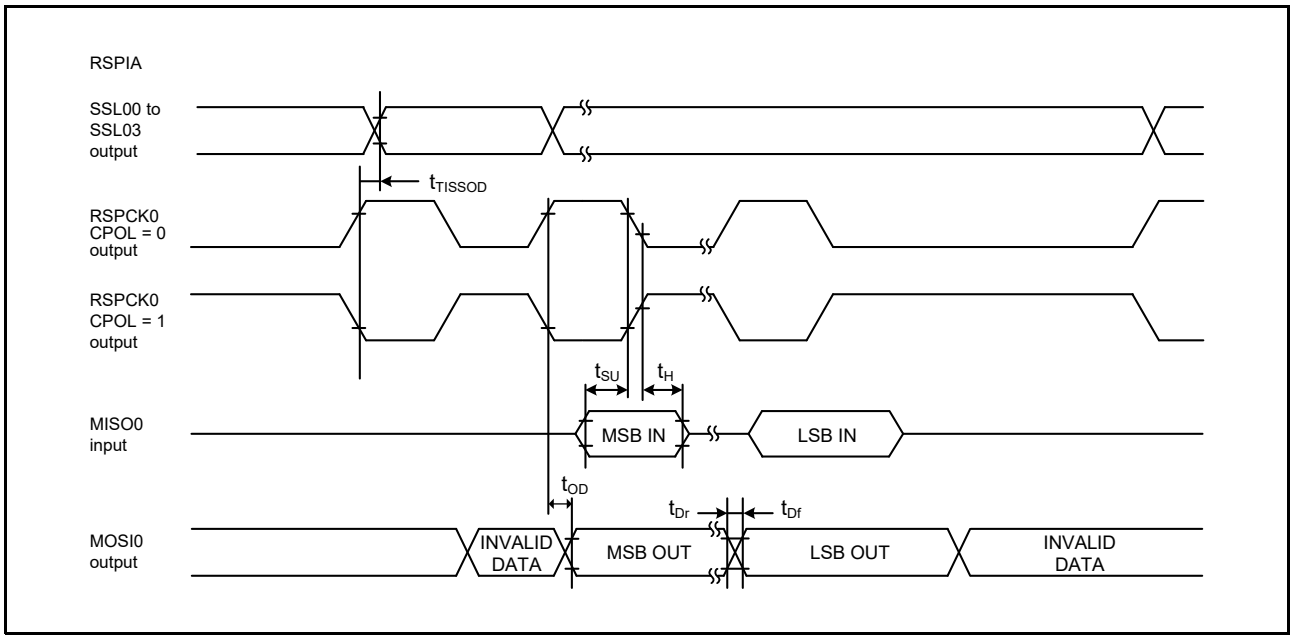


Figure 49.62 RSPIA Timing (Master, TI SSP)

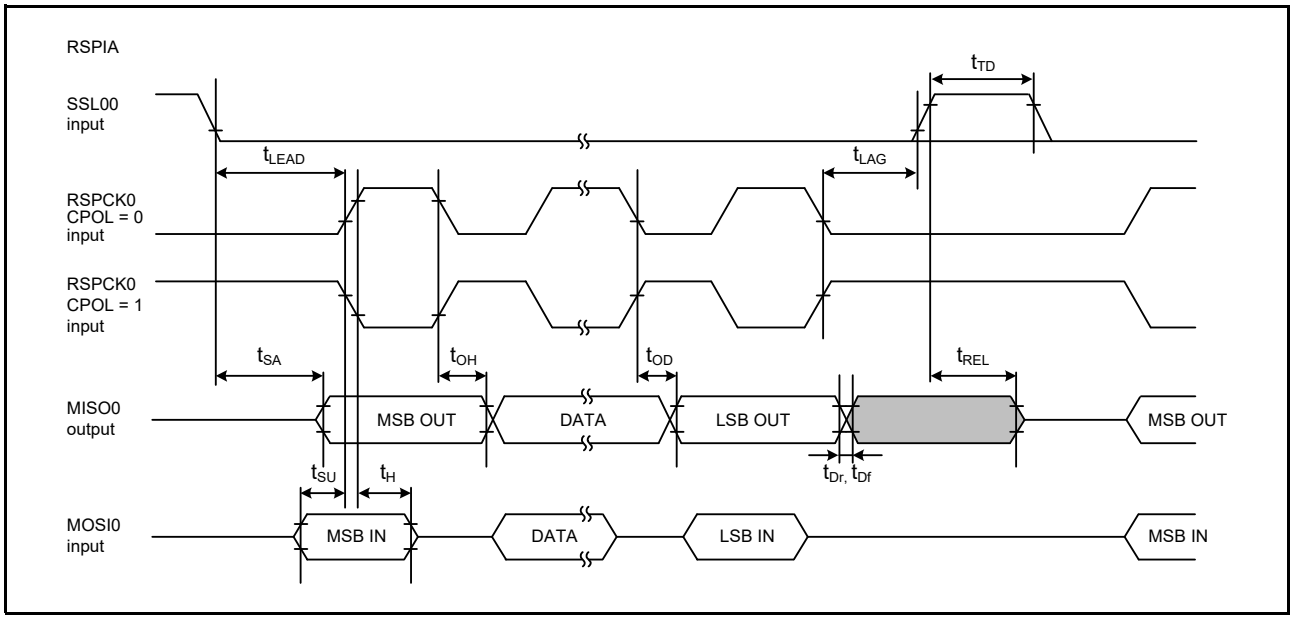


Figure 49.63 RSPIA Timing (Slave, Motorola SPI, CPHA = 0)

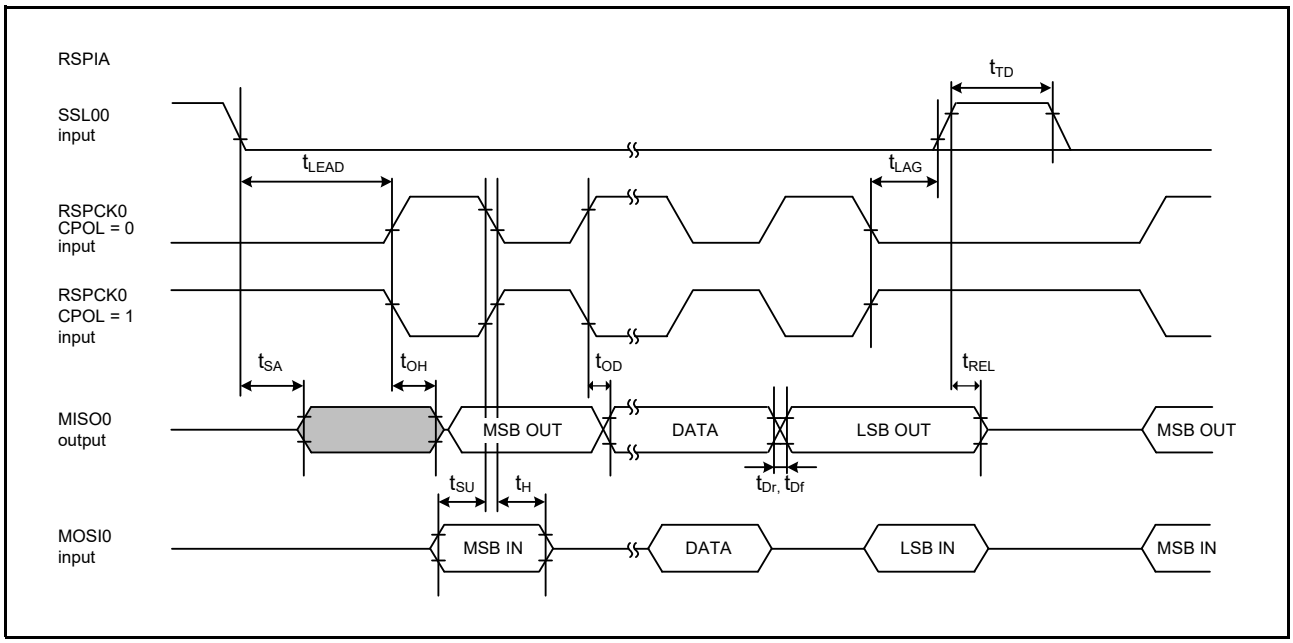


Figure 49.64 RSPIA Timing (Slave, Motorola SPI, CPHA = 1)

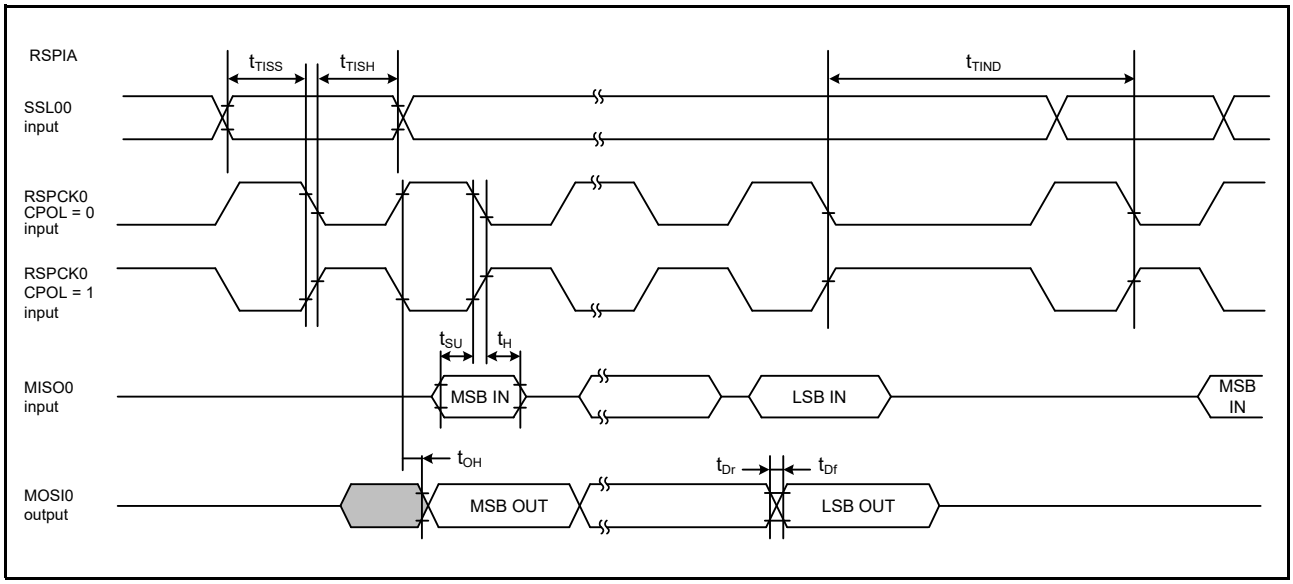


Figure 49.65 RSPIA Timing (Slave, TI SSP, Transmit with Delay between Frames)

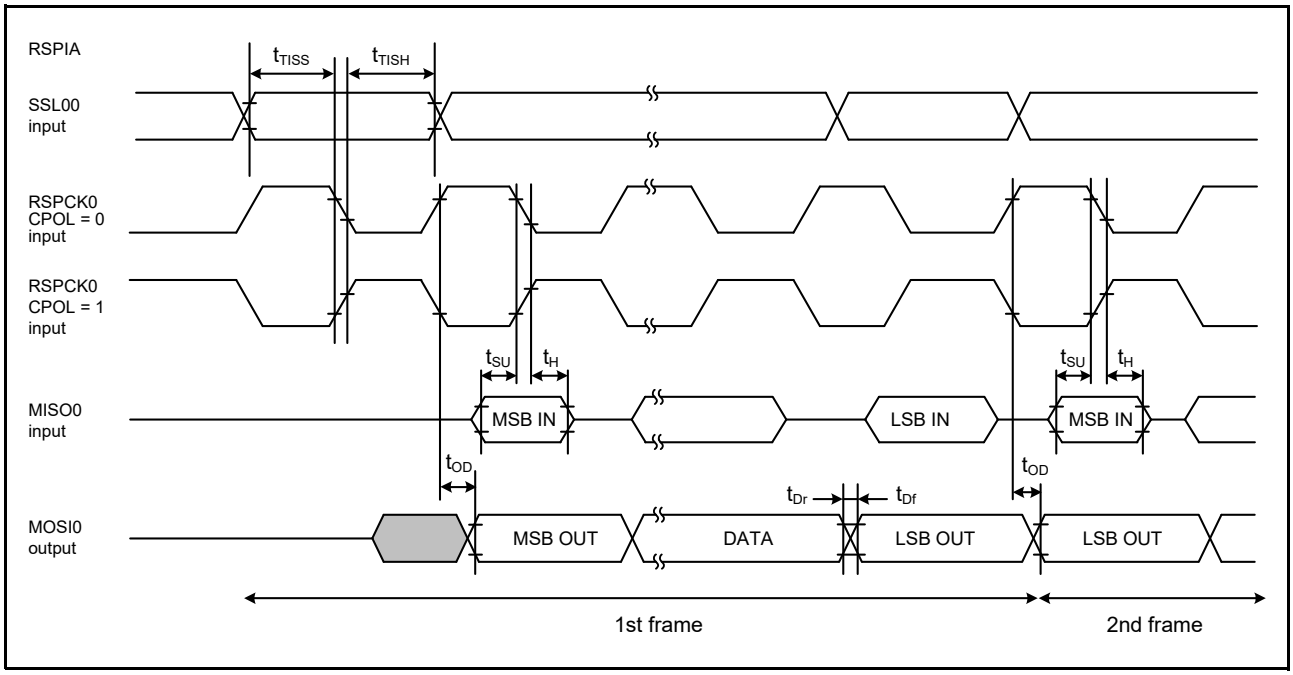


Figure 49.66 RSPIA Timing (Slave, TI SSP, Transmit with No Delay between Frames)

49.4.5.13 RIIC

Table 49.39 RIIC Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = AVCC1 = AVCC2 = 3.0 to 5.5 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V, $T_a = T_{opr}$,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz,
High-drive output is selected by the driving ability control register.

Item		Symbol	Min.*1	Max.*1	Unit	Test Conditions*3
RIIC (Standard-mode, SMBus)	SCL input cycle time	t_{SCL}	6(12) × t_{IICcyc} + 1300	—	ns	Figure 49.67
	SCL input high pulse width	t_{SCLH}	3(6) × t_{IICcyc} + 300	—		
	SCL input low pulse width	t_{SCLL}	3(6) × t_{IICcyc} + 300	—		
	SCL, SDA input rise time	t_{Sr}	—	1000		
	SCL, SDA input fall time	t_{Sf}	—	300		
	SCL, SDA input spike pulse removal time	t_{SP}	0	1(4) × t_{IICcyc}		
	SDA input bus free time	t_{BUF}	3(6) × t_{IICcyc} + 300	—		
	Start condition input hold time	t_{STAH}	t_{IICcyc} + 300	—		
	Restart condition input setup time	t_{STAS}	1000	—		
	Stop condition input setup time	t_{STOS}	1000	—		
	Data input setup time	t_{SDAS}	t_{IICcyc} + 50	—		
	Data input hold time	t_{SDAH}	0	—		
	SCL, SDA capacitive load	C_b*2	—	400	pF	
RIIC (Fast-mode)	SCL input cycle time	t_{SCL}	6(12) × t_{IICcyc} + 600	—	ns	
	SCL input high pulse width	t_{SCLH}	3(6) × t_{IICcyc} + 300	—		
	SCL input low pulse width	t_{SCLL}	3(6) × t_{IICcyc} + 300	—		
	SCL, SDA input rise time	t_{Sr}	20 × (External pull-up voltage/5.5 V)	300		
	SCL, SDA input fall time	t_{Sf}	20 × (External pull-up voltage/5.5 V)	300		
	SCL, SDA input spike pulse removal time	t_{SP}	0	1(4) × t_{IICcyc}		
	SDA input bus free time	t_{BUF}	3(6) × t_{IICcyc} + 300	—		
	Start condition input hold time	t_{STAH}	t_{IICcyc} + 300	—		
	Restart condition input setup time	t_{STAS}	300	—		
	Stop condition input setup time	t_{STOS}	300	—		
	Data input setup time	t_{SDAS}	t_{IICcyc} + 50	—		
	Data input hold time	t_{SDAH}	0	—		
	SCL, SDA capacitive load	C_b*2	—	400	pF	

Note: t_{IICcyc}: RIIC internal reference clock (IICφ) cycle
Note 1. The value within parentheses is applicable when the value of the ICMR3.NF[1:0] bits is 11b while the digital filter is enabled by the setting ICFER.NFE = 1.
Note 2. C_b is the total capacitance of the bus lines.
Note 3. When VCC ≥ 4.5 V, VOLSR.RICVLS = 0
When VCC < 4.5 V, VOLSR.RICVLS = 1

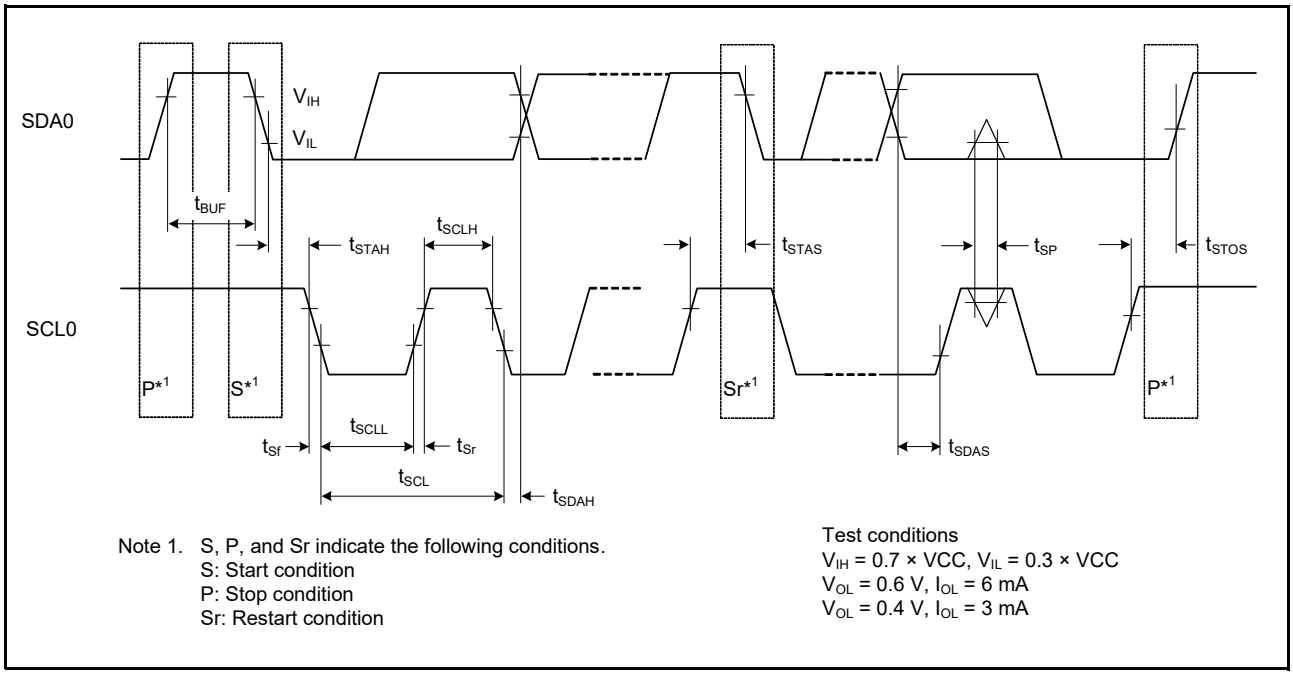


Figure 49.67 RIIC Bus Interface Input/Output Timing

49.4.5.14 RI3C

Table 49.40 RI3C Timing (Open Drain Timing Parameters)

Conditions: VCC = AVCC0 = AVCC1 = AVCC2 = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V, $T_a = T_{opr}$,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
SCL clock low period	t_{LOW_OD}	200*1, *2	—	—	ns	Figure 49.68
	$t_{DIG\_OD\_L}$	t_{LOW_ODmin} + t_{fDA_ODmin}	—	—		
SDA signal fall time	t_{fDA_OD}	t_{CF}	—	33	ns	Figure 49.68
SDA data setup time open drain mode	t_{SU_OD}	3*1	—	—	ns	Figure 49.68, Figure 49.69
Clock after START (S) condition	t_{CAS}	38.4 ns*3	—	For ENTAS0: 1 µs	—	Figure 49.68
			—	For ENTAS1: 100 µs		
			—	For ENTAS2: 2 ms		
			—	For ENTAS3: 50 ms*4		
Clock before STOP (P) condition	t_{CBP}	$t_{CASmin}/2$	—	—	sec	Figure 49.70
Current controller to secondary controller overlap time during handoff	$t_{CRHPOverlap}$	$t_{DIG\_OD\_Lmin}$	—	—	ns	Figure 49.71
Bus available condition	t_{AVAL}	1*5	—	—	µs	
Bus idle condition	t_{IDLE}	1	—	—	ms	
Time internal where new controller not driving SDA low	$t_{NEWCRLock}$	$t_{AVALmin}$	—	—	µs	Figure 49.71

Note 1. This is approximately equal to $t_{LOWmin} + t_{DS\_ODmin} + t_{rDA\_ODtyp} + t_{SU\_ODmin}$.
Note 2. The controller may use a shorter low period if it knows that this is safe, i.e., that SDA is already above V_{IH}.
Note 3. On a legacy bus where I^2C devices need to see start, the t_{CAS} Min value is further constrained.
Note 4. Targets that do not support the optional ENTASx CCCs shall use the t_{CAS} Max value shown for ENTAS3
Note 5. On a mixed bus with Fm legacy I^2C devices, t_{AVAL} is 300 ns shorter than the Fm bus free condition time (t_{BUF})

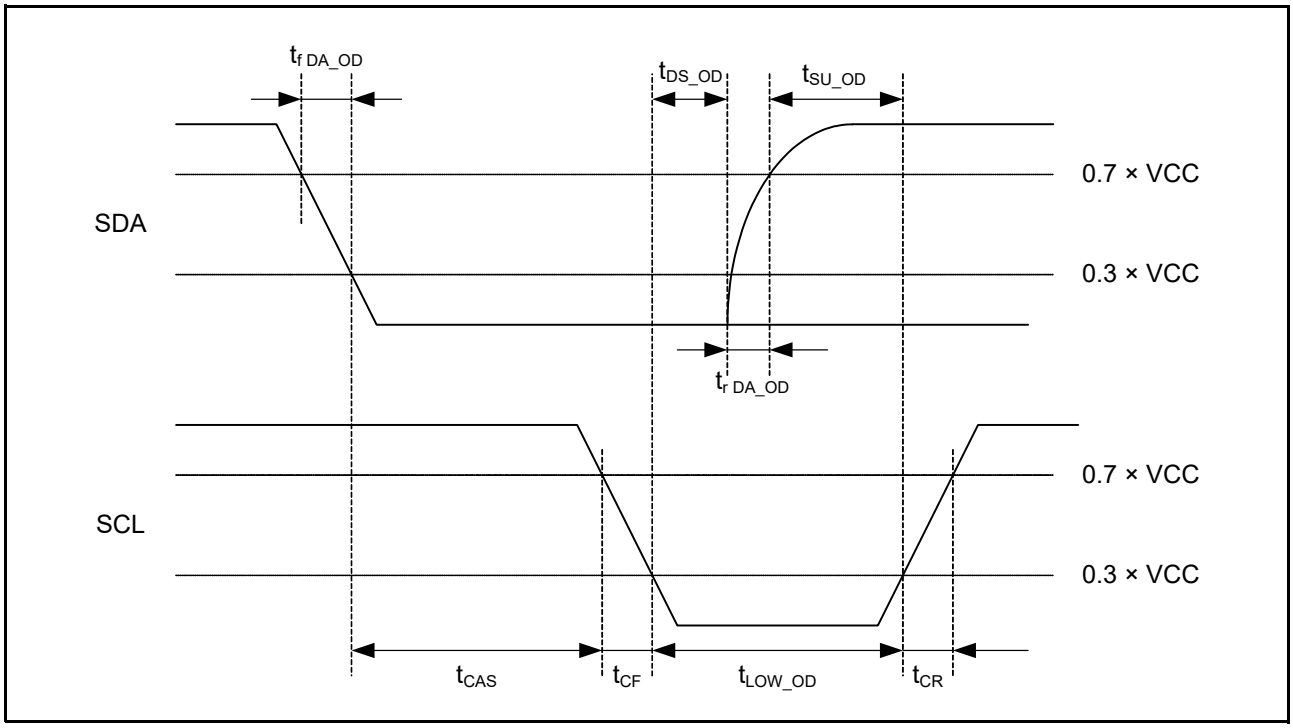


Figure 49.68 RI3C Start Condition Timing

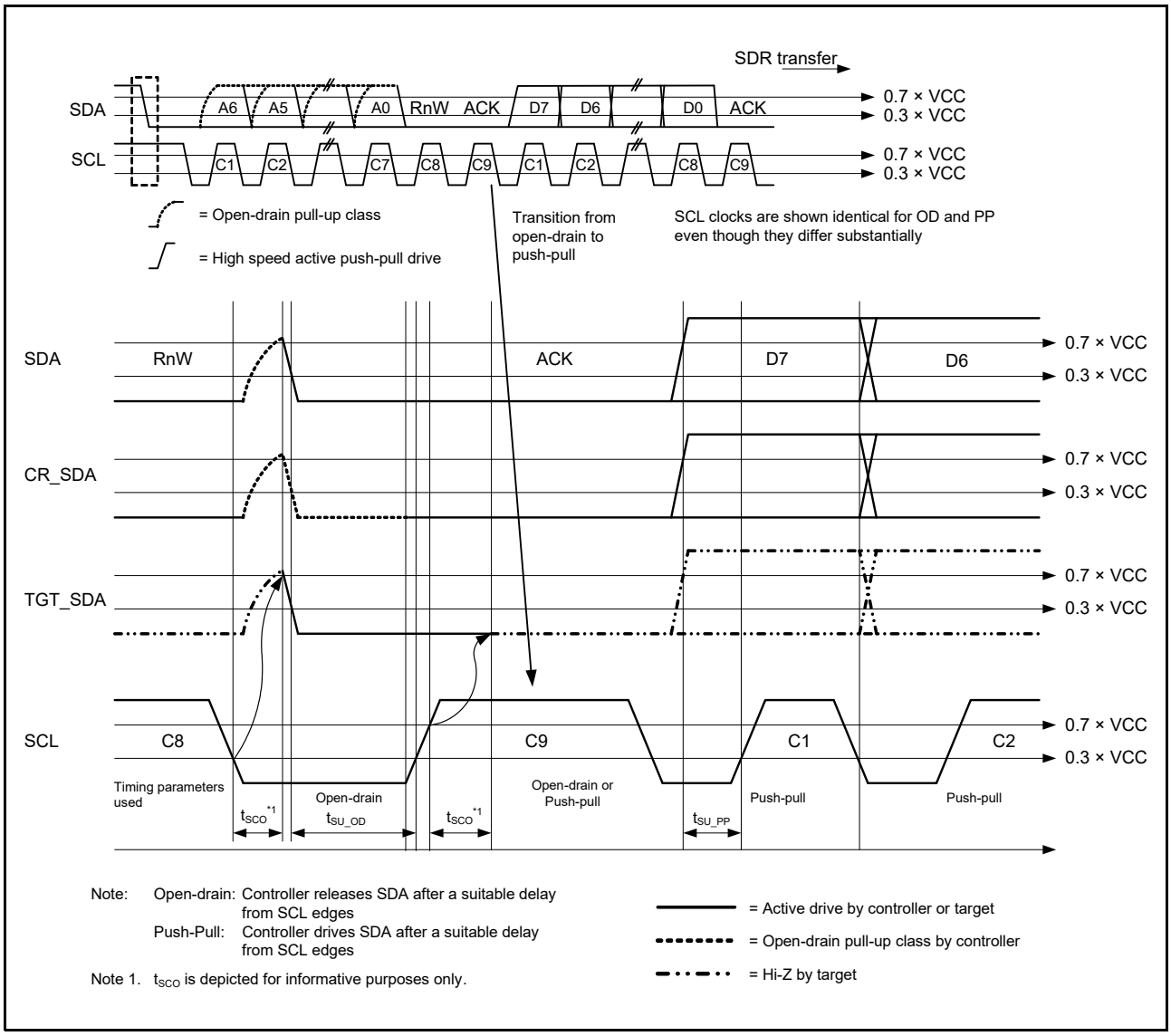


Figure 49.69 RI3C Data Transfer — ACK by Target

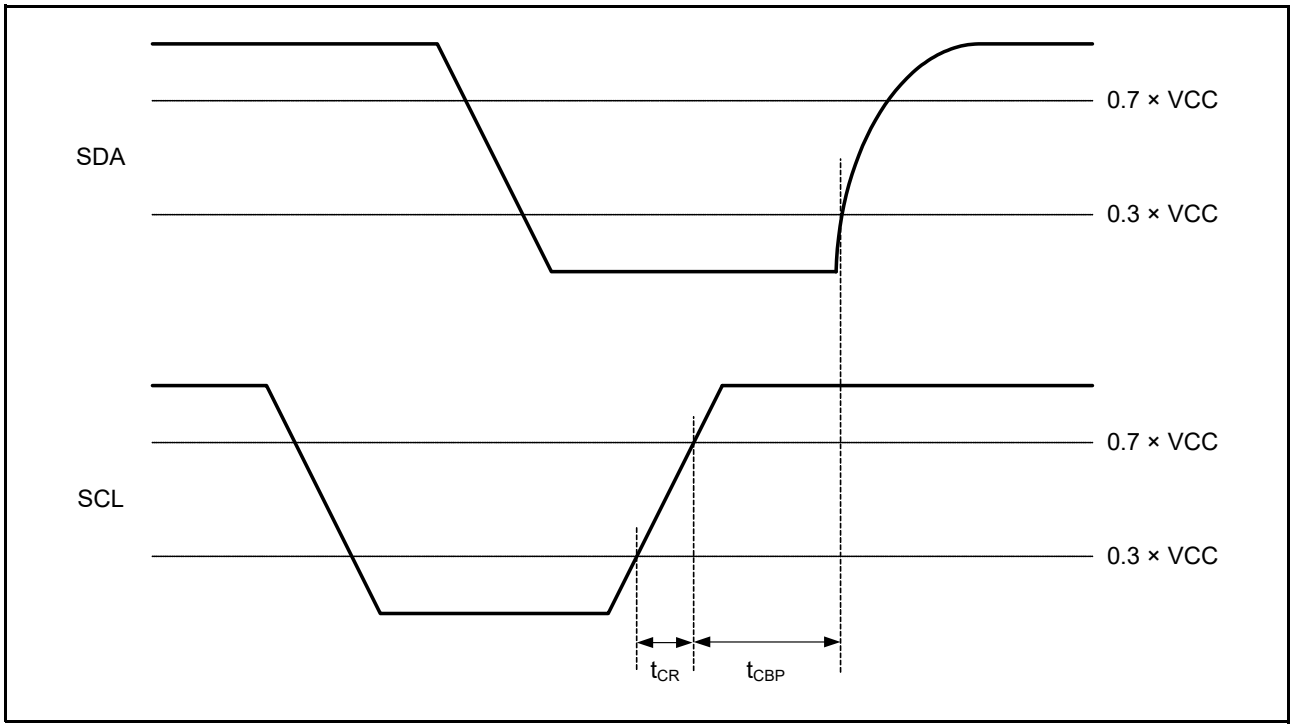


Figure 49.70 RI3C Stop Condition Timing

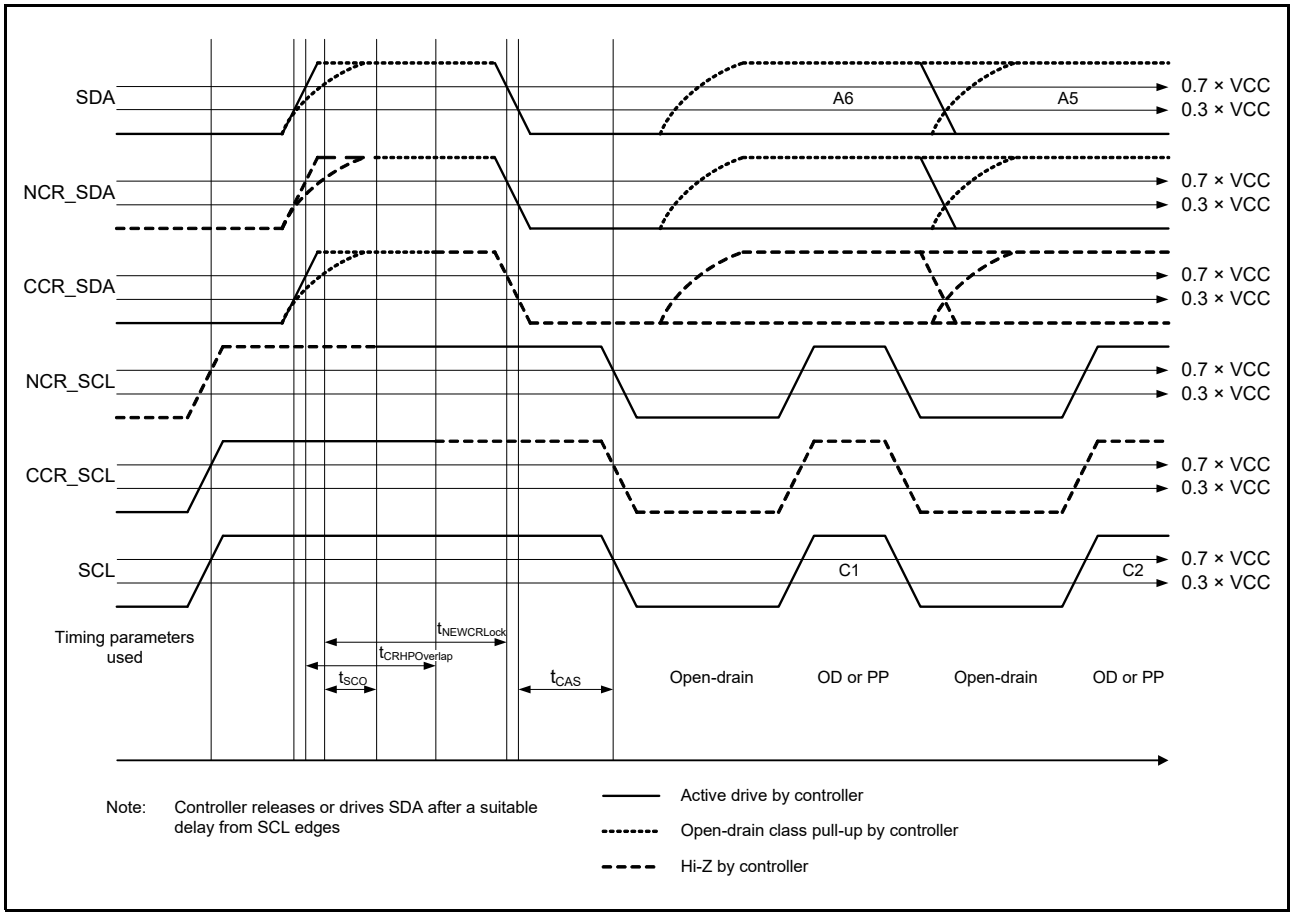


Figure 49.71 RI3C Output Timing

Table 49.41 RI3C Timing (Push-Pull Timing Parameters for SDR)

Conditions: VCC = AVCC0 = AVCC1 = AVCC2 = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V, $T_a = T_{opr}$,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
SCL clock frequency		f_{SCL}	0.01*1	—	10	MHz	
SCL clock low period		t_{LOW}	35	—	—	ns	Figure 49.72
		t_{DIG_L}	50*2, *3	—	—	ns	
SCL clock high period		t_{HIGH}	35	—	—	ns	
		t_{DIG_H}	50*2	—	—	ns	
Clock in to data out for target		t_{SCO}	—	—	42	ns	Figure 49.73
SCL clock rise time		t_{CR}	—	—	150 * 1 /f_{SCL} (capped at 60)	ns	Figure 49.72
SCL clock fall time		t_{CF}	—	—	150 * 1 /f_{SCL} (capped at 60)	ns	
SDA signal data hold in push-pull mode	Controller	t_{HD_PP}	t_{CR} + 3*3, t_{CF} + 3*3	—	—	—	Figure 49.74
	Target	t_{HD_PP}	0	—	—	—	Figure 49.75
SDA signal data setup in push-pull mode		t_{SU_PP}	3	—	—	ns	Figure 49.73, Figure 49.74
Clock after repeated start (Sr)		t_{CASr}	t_{CASmin}/2	—	N/A	ns	Figure 49.76
Clock before repeated start (Sr)		t_{CBSr}	t_{CASmin}/2	—	N/A	ns	Figure 49.76
Capacitive load per bus line (SDA/SCL)		C_b*4	—	—	50	pF	

Note 1. $f_{SCL} = 1 / (t_{DIG\_L} + t_{DIG\_H})$
Note 2. t_{DIG_L} and t_{DIG_H} are the clock low and high periods as seen at the receiver end of the I3C bus using V_{IL} and V_{IH}.
Note 3. As both edges are used, the hold time must be satisfied for the respective edges, for example, t_{CF} + 3 for falling edge clocks, and t_{CR} + 3 for rising edge clocks.
Note 4. C_b is the total capacitance of the bus lines.

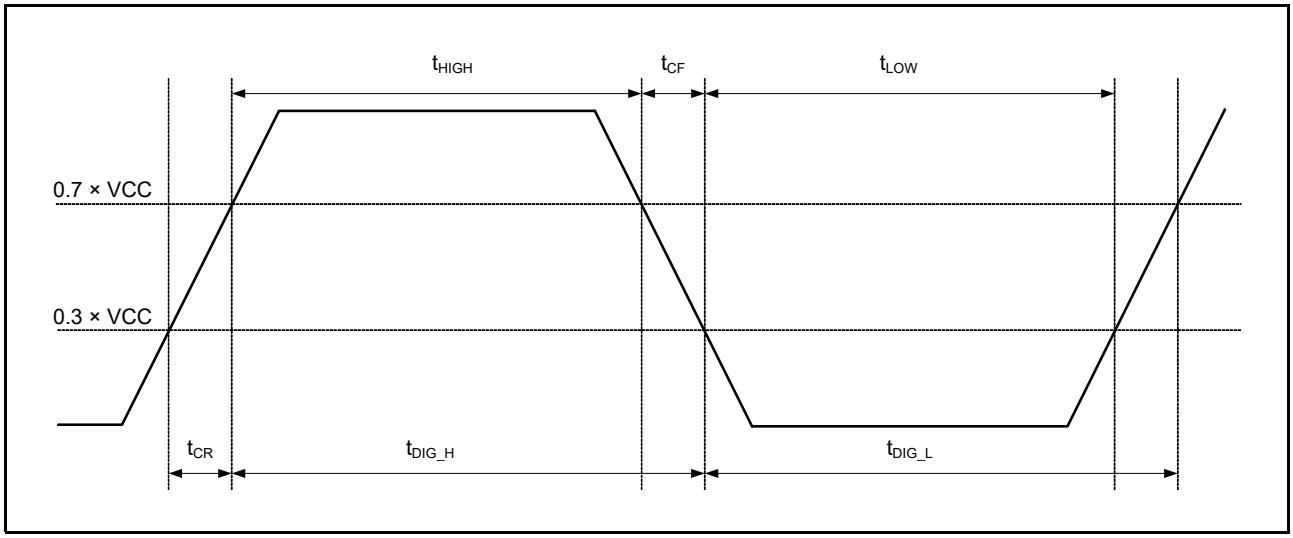


Figure 49.72 t_{DIG_H} and t_{DIG_L}

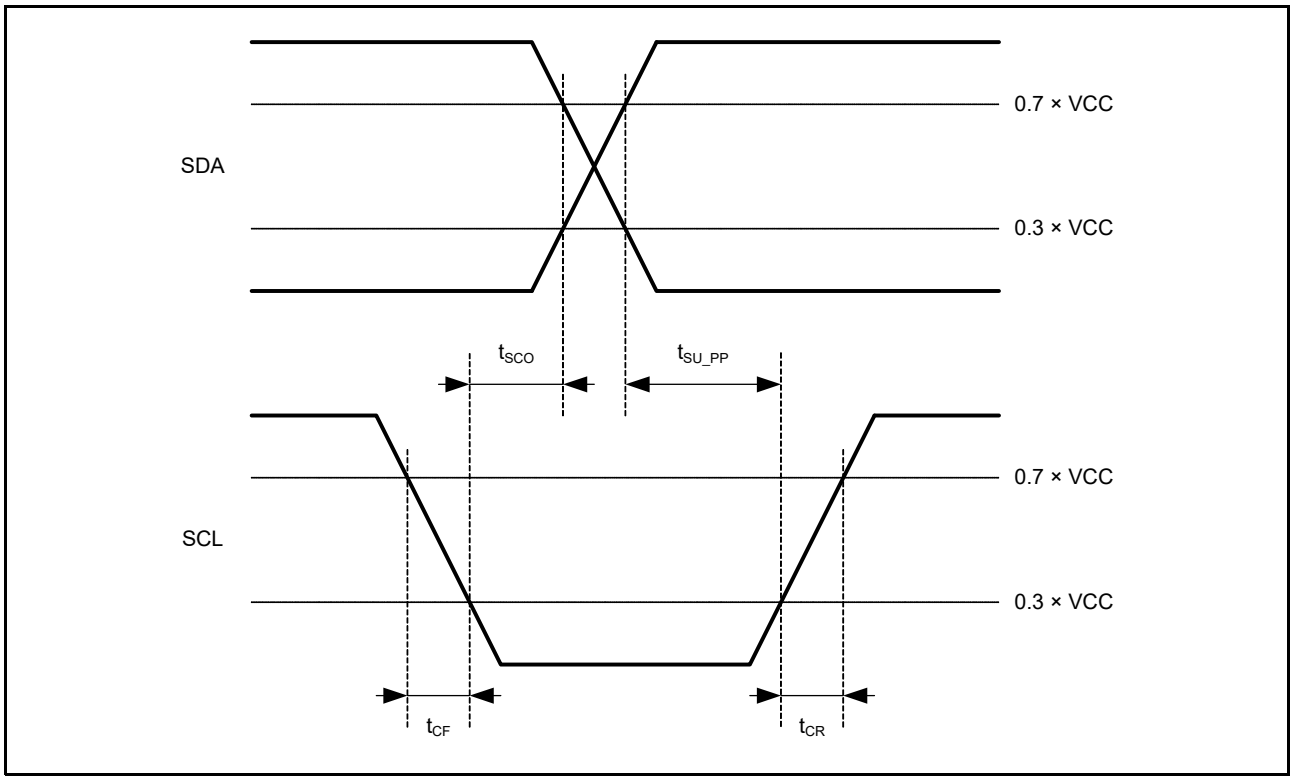


Figure 49.73 RI3C Target Output Timing

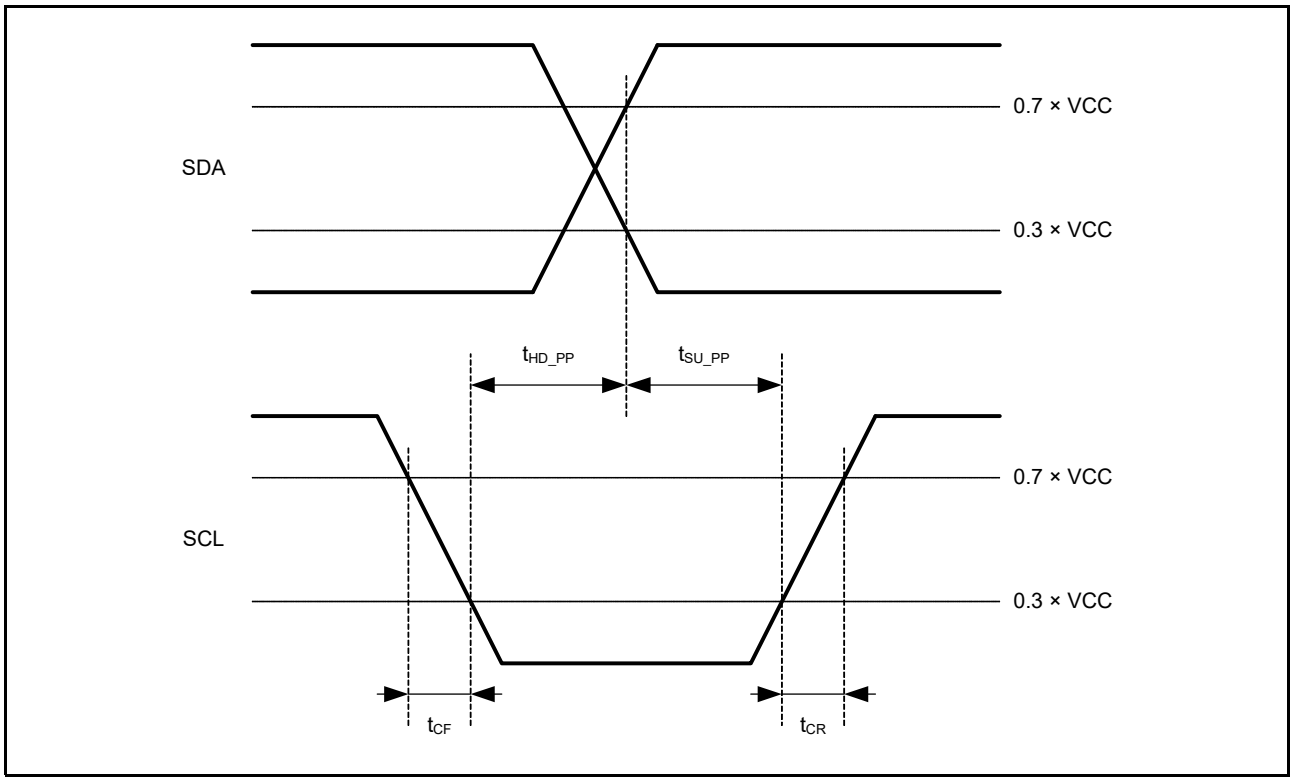


Figure 49.74 RI3C Bus Controller Output Timing

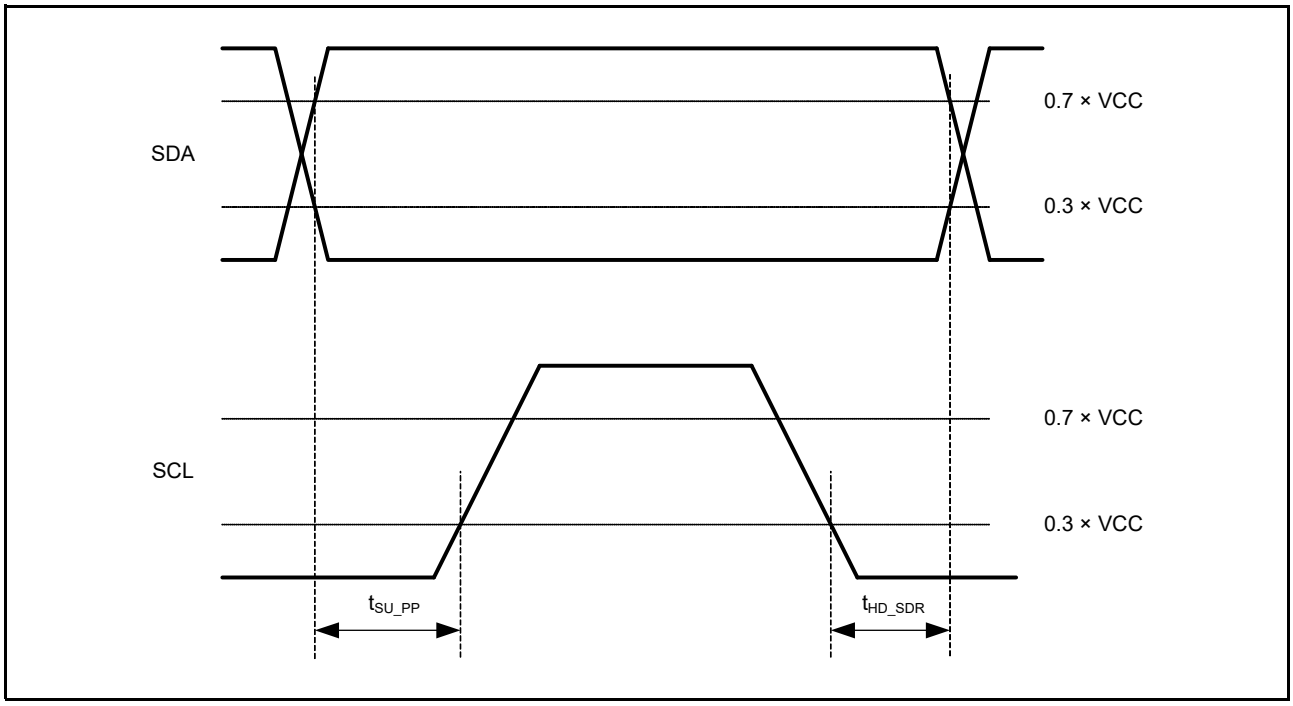


Figure 49.75 Controller SDR Timing

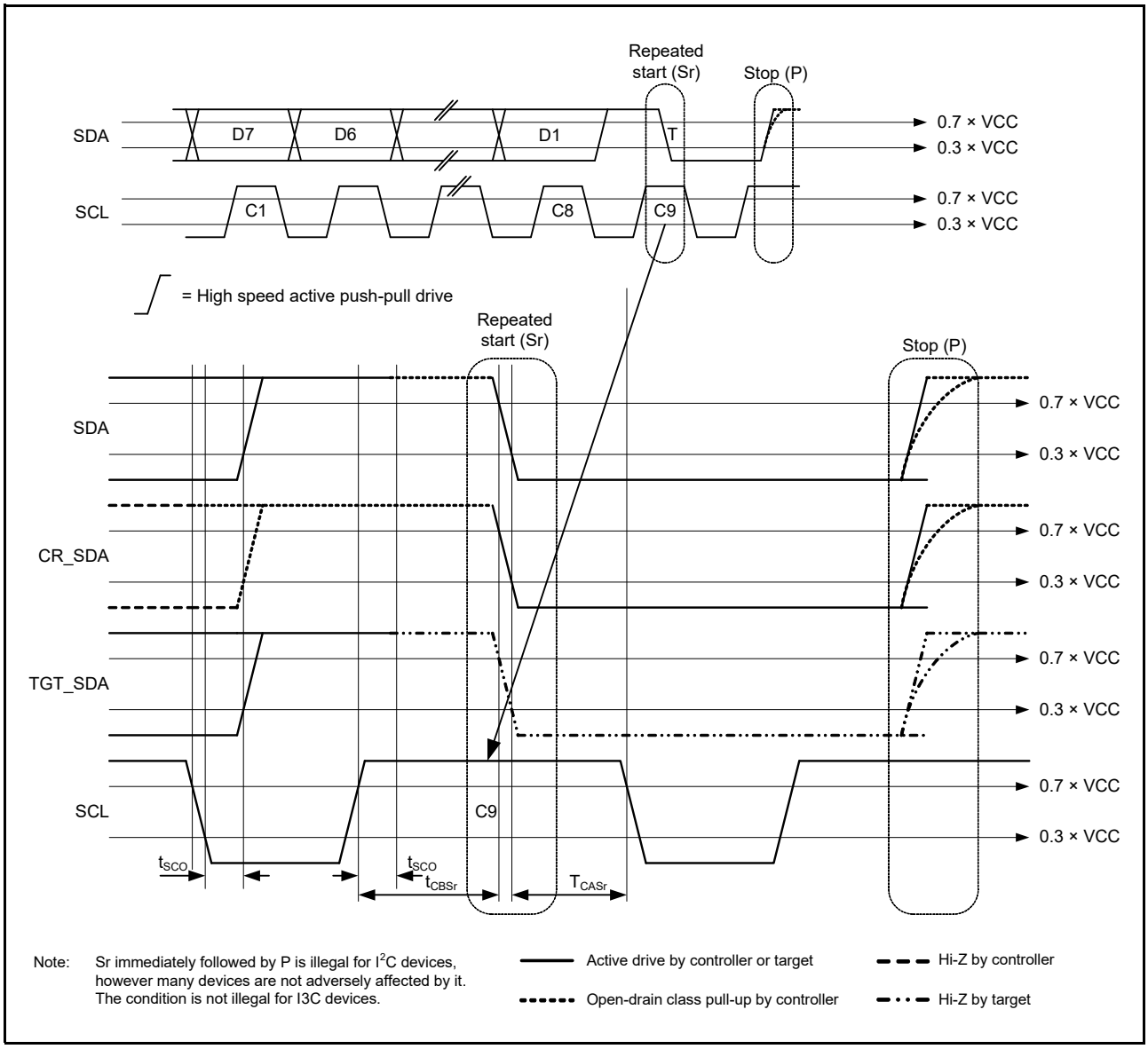


Figure 49.76 T-Bit When Controller Ends Read with Repeated Start and Stop

49.4.5.15 HRPWM

Table 49.42 HRPWM Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = AVCC1 = AVCC2 = 3.0 to 5.5 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V, $T_a = T_{opr}$,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz,
Output load conditions: $V_{OH} = 0.5 \times VCC$, $V_{OL} = 0.5 \times VCC$, C = 30 pF,
High-drive output is selected by the driving ability control register.

Item	Min.	Typ.	Max.	Unit	Test Conditions
Input frequency (f_{IN})	80	—	120	MHz	
Resolution	—	260	—	ps	f_{IN} = 120 MHz
DNL*1	—	±2.0	—	LSB	

Note 1. The value is that difference from code to code normalized by the resolution (1 LSB).

49.4.5.16 CANFD

Table 49.43 CANFD Timing

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = AVCC1 = AVCC2 = 3.0 to 5.5 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V, $T_a = T_{opr}$

Item		Symbol	Min.	Max.	Unit
Classic CAN mode	Bit rate for communications		—	1	Mbps
CAN FD mode	Bit rate for communications		—	1	Mbps
	Bit rate for communications (only for data)		—	5	

49.5 A/D Conversion Characteristics

Table 49.44 12-Bit A/D (Unit 0, 1, 2) Conversion Characteristics (1)

Conditions: VCC = 2.7 to 5.5 V, 4.5 V ≤ AVCC0 = AVCC1 = AVCC2 ≤ 5.5 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V, $T_a = T_{opr}$, PCLKB = PCLKD = 8 to 60 MHz*1,
Source impedance = 1.0 kΩ

Item				Min.	Typ.	Max.	Unit	Test Conditions
Resolution				12	12	12	Bit	
Analog input capacitance				—	—	30	pF	
Conversion time*2 (Operation at PCLKD = 60 MHz)	AN000 to AN002, AN100 to AN102	Channel-dedicated sample-and-hold circuits in use	Constant sampling enabled	1.00	—	—	μs	• Sampling time: 24 PCLKD
			Constant sampling disabled	1.40	—	—		• Sampling time of channel-dedicated sample-and-hold circuits: 24 PCLKD • Sampling time: 24 PCLKD
		Channel-dedicated sample-and-hold circuits not in use		0.90	—	—		• Sampling time: 30 PCLKD
	AN003, AN103			0.90	—	—		• Sampling time: 30 PCLKD
	AN200 to AN211			0.95	—	—		• Sampling time: 33 PCLKD
	AN216 to AN217			1.05	—	—		• Sampling time: 39 PCLKD
Offset error		Channel-dedicated sample-and-hold circuits in use		—	±1.5	±6.0	LSB	AN000 to AN002, AN100 to AN102 = 0.2 V
		Channel-dedicated sample-and-hold circuits not in use		—	±1.5	±5.0		
Full-scale error		Channel-dedicated sample-and-hold circuits in use		—	±1.5	±5.5	LSB	AN000 to AN002 = AVCC0 – 0.2 V AN100 to AN102 = AVCC1 – 0.2 V
		Channel-dedicated sample-and-hold circuits not in use		—	±1.5	±4.5		
Quantization error		Channel-dedicated sample-and-hold circuits in use		—	±0.5	—	LSB	
		Channel-dedicated sample-and-hold circuits not in use		—	±0.5	—		
Absolute accuracy	AN000 to AN002, AN100 to AN102	Channel-dedicated sample-and-hold circuits in use		—	±3.0	±6.0	LSB	
		Channel-dedicated sample-and-hold circuits not in use		—	±2.5	±5.5		
	AN003, AN103			—	±2.5	±5.5		
	AN200 to AN211			—	±2.5	±5.5		
	AN216 to AN217			—	±2.5	±6.5		
DNL differential nonlinearity error		Channel-dedicated sample-and-hold circuits in use		—	±1.0	±2.5	LSB	
		Channel-dedicated sample-and-hold circuits not in use		—	±1.0	±1.5		
INL integral nonlinearity error		Channel-dedicated sample-and-hold circuits in use		—	±1.5	±4.0	LSB	
		Channel-dedicated sample-and-hold circuits not in use		—	±1.5	±2.5		
Holding time of the channel-dedicated sample-and-hold circuit				—	—	20	μs	
Dynamic range	AN000 to AN002	Channel-dedicated sample-and-hold circuits in use		0.2	—	AVCC0 – 0.2	V	
	AN100 to AN102	Channel-dedicated sample-and-hold circuits in use		0.2	—	AVCC1 – 0.2		

Note 1. When PCLKD was higher than 40 MHz, 1000 pF capacitors were placed in parallel with the 0.1-μF capacitors between AVCC0 and AVSS0, AVCC1 and AVSS1, and AVCC2 and AVSS2 for measurement of the A/D conversion characteristics.

Note 2. The conversion time is the sum of the sampling time and the comparison time. The numbers of sampling-clock cycles are indicated as the test conditions.

Table 49.45 12-Bit A/D (Unit 0, 1, 2) Conversion Characteristics (2)

Conditions: VCC = 2.7 to 4.5 V, 3.0 V ≤ AVCC0 = AVCC1 = AVCC2 < 4.5V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V, $T_a = T_{opr}$, PCLKB = PCLKD = 8 to 40 MHz,
Source impedance = 1.0 kΩ

Item				Min.	Typ.	Max.	Unit	Test Conditions
Resolution				12	12	12	Bit	
Analog input capacitance				—	—	30	pF	
Conversion time*1 (Operation at PCLKD = 40 MHz)	AN000 to AN002, AN100 to AN102	Channel-dedicated sample-and-hold circuits in use	Constant sampling enabled	1.35	—	—	μs	• Sampling time: 18 PCLKD
			Constant sampling disabled	1.80	—	—		• Sampling time of channel-dedicated sample-and-hold circuits: 18 PCLKD • Sampling time: 18 PCLKD
		Channel-dedicated sample-and-hold circuits not in use		1.13	—	—		• Sampling time: 21 PCLKD
	AN003, AN103			1.13	—	—		• Sampling time: 21 PCLKD
	AN200 to AN211			1.20	—	—		• Sampling time: 24 PCLKD
	AN216 to AN217			1.28	—	—		• Sampling time: 27 PCLKD
Offset error		Channel-dedicated sample-and-hold circuits in use		—	±1.5	±7.5	LSB	AN000 to AN002, AN100 to AN102 = 0.2 V
		Channel-dedicated sample-and-hold circuits not in use		—	±1.5	±6.5		
Full-scale error		Channel-dedicated sample-and-hold circuits in use		—	±1.5	±7.5		AN000 to AN002 = AVCC0 – 0.2 V AN100 to AN102 = AVCC1 – 0.2 V
		Channel-dedicated sample-and-hold circuits not in use		—	±1.5	±6.5		
Quantization error		Channel-dedicated sample-and-hold circuits in use		—	±0.5	—		
		Channel-dedicated sample-and-hold circuits not in use		—	±0.5	—		
Absolute accuracy	AN000 to AN002, AN100 to AN102	Channel-dedicated sample-and-hold circuits in use		—	±4.0	±8.0		
		Channel-dedicated sample-and-hold circuits not in use		—	±2.5	±7.0		
	AN003, AN103			—	±2.5	±7.0		
	AN200 to AN211			—	±2.5	±7.0		
	AN216 to AN217			—	±2.5	±8.0		
DNL differential nonlinearity error		Channel-dedicated sample-and-hold circuits in use		—	±1.0	±4.5		
		Channel-dedicated sample-and-hold circuits not in use		—	±1.0	±3.5		
INL integral nonlinearity error		Channel-dedicated sample-and-hold circuits in use		—	±2.0	±5.0		
		Channel-dedicated sample-and-hold circuits not in use		—	±1.5	±3.5		
Channel-dedicated sample-and-hold characteristics of hold circuits				—	—	20	μs	
Dynamic range	AN000 to AN002	Channel-dedicated sample-and-hold circuits in use		0.2	—	AVCC0 – 0.2	V	
	AN100 to AN102	Channel-dedicated sample-and-hold circuits in use		0.2	—	AVCC1 – 0.2		

Note 1. The conversion time is the sum of the sampling time and the comparison time. The numbers of sampling-clock cycles are indicated as the test conditions.

Table 49.46 A/D Internal Reference Voltage Characteristics

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = AVCC1 = AVCC2 = 3.0 to 5.5 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V, $T_a = T_{opr}$, PCLKB = PCLKD = 8 to 60 MHz

Item	Min.	Typ.	Max.	Unit	Test Conditions
A/D internal reference voltage	1.20	1.25	1.30	V	

Note: The above specification values apply during normal operations.

49.6 Programmable Gain Amplifier Characteristics

Table 49.47 Programmable Gain Amplifier Characteristics

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = AVCC1 = AVCC2 = 3.0 to 5.5 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V, $T_a = T_{opr}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input offset voltage	V_{IO}	—	3	8	mV	
Single-ended input voltage range	V_{ISR}	V_{OR} (min)/G	—	V_{OR} (max)/G	V	
Output voltage range	V_{OR}	0.10 × AVCCn	—	0.90 × AVCCn		G = 2.000 to 3.636
		0.15 × AVCCn	—	0.85 × AVCCn		G = 4.000 to 6.667
		0.20 × AVCCn	—	0.80 × AVCCn		G = 8.000 to 20.000
Gain	G	2.000	—	20.000	Linear gain	
Gain error	E_G	—	±0.5	±1.5	%	G = 2.000
		—	±0.5	±1.5		G = 2.500
		—	±0.5	±1.5		G = 3.077
		—	±0.5	±1.5		G = 3.636
		—	±0.6	±1.5		G = 4.000
		—	±0.6	±1.5		G = 4.444
		—	±0.7	±1.5		G = 5.000
		—	±0.7	±1.5		G = 6.667
		—	±0.7	±1.5		G = 8.000
		—	±0.7	±2.5		G = 10.000
		—	±1.1	±2.5		G = 13.333
		—	±1.3	±4.0		G = 20.000
Slew rate	SR	10	—	—	V/µs	
Operation stabilization time	t_{start}	—	—	5	µs	

n = 0 and 1

49.7 Comparator Characteristics

Table 49.48 Comparator Characteristics

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = AVCC1 = AVCC2 = 3.0 to 5.5 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V, $T_a = T_{opr}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input offset voltage	V_{IO}	—	8	15	mV	
Reference input voltage range	V_{ref}	0	—	AVCC1	V	CMPSEL1.CVRS[3:0] = 0100b, 1000b
		0	—	AVCC2		CMPSEL1.CVRS[3:0] = 0001b, 0010b
Response time	$t_{tot(r)}$	—	—	200	ns	VOD = 100 mV CMPCTL.CDFS = 0
	$t_{tot(f)}$	—	—	200		
Waiting time for stabilization following switching of the input	t_{cwait}	300	—	—		
Operation stabilization time	t_{cmp}	—	—	1	µs	

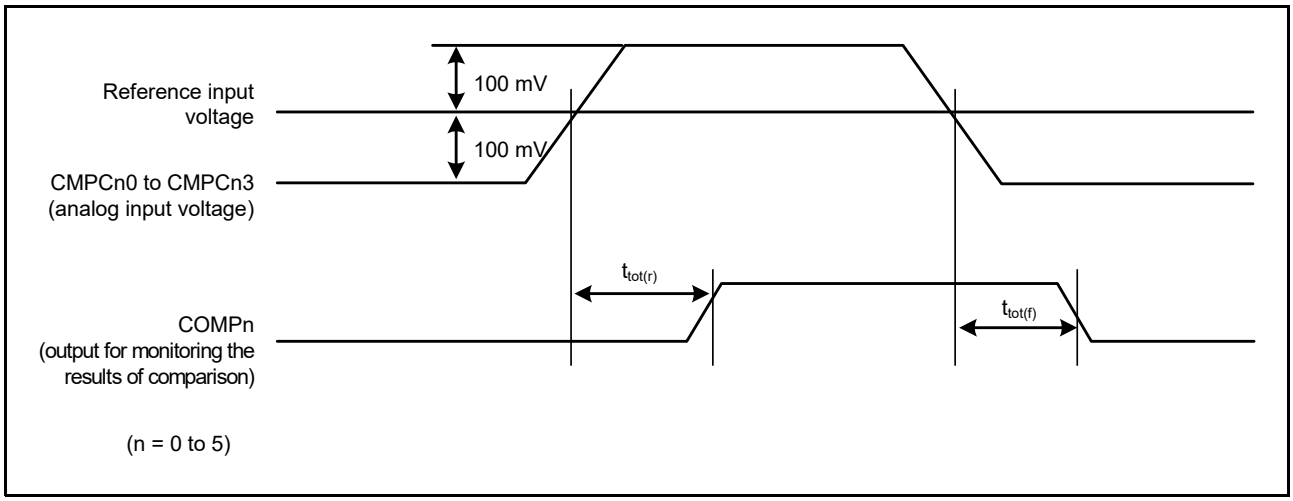


Figure 49.77 Comparator Response Time

49.8 D/A Conversion Characteristics

Table 49.49 D/A Conversion Characteristics

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = AVCC1 = AVCC2 = 3.0 to 5.5 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V, $T_a = T_{opr}$

Item	Min.	Typ.	Max.	Unit	Test Conditions
Resolution	12	12	12	Bit	
Absolute accuracy	—	—	±6.0	LSB	2-MΩ resistive load, 10-bit conversion
Differential nonlinearity error (DNL)	—	±1.0	±2.0	LSB	2-MΩ resistive load
Output resistance (R_o)	—	5.7	—	kΩ	
Conversion time	—	—	3	µs	20-pF capacitive load

49.9 Temperature Sensor Characteristics

Table 49.50 Temperature Sensor Characteristics

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = AVCC1 = AVCC2 = 3.0 to 5.5 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V, $T_a = T_{opr}$,
PCLKB = PCLKD = 8 to 60 MHz

Item	Min.	Typ.	Max.	Unit	Test Conditions
Relative accuracy	—	±1.0	—	°C	
Temperature slope	—	–2.0	—	mV/°C	
Output voltage	—	0.63	—	V	T_a = 25°C
Temperature sensor start time	—	—	200	µs	
Sampling time*1	3	—	—	µs	

Note 1. Set the S12AD2.ADSSTRT register such that the sampling time of the 12-bit A/D converter satisfies this specification.

49.10 Power-on Reset Circuit and Voltage Detection Circuit Characteristics

Table 49.51 Power-on Reset Circuit and Voltage Detection Circuit Characteristics

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = AVCC1 = AVCC2 = 3.0 to 5.5 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V, $T_a = T_{opr}$

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Voltage detection level	Power-on reset (POR)	V_{POR}	2.46	2.58	2.70	V	Figure 49.78
	Voltage detection circuit (LVD0)	V_{det0_1}	4.04	4.22	4.40		Figure 49.79
		V_{det0_2}	2.71	2.83	2.95		
	Voltage detection circuit (LVD1)	V_{det1_0}	4.39	4.57	4.75		Figure 49.80
		V_{det1_1}	4.29	4.47	4.65		
		V_{det1_2}	4.14	4.32	4.50		
		V_{det1_3}	2.81	2.93	3.05		
		V_{det1_4}	2.76	2.88	3.00		
	Voltage detection circuit (LVD2)	V_{det2_0}	4.39	4.57	4.75		Figure 49.81
		V_{det2_1}	4.29	4.47	4.65		
		V_{det2_2}	4.14	4.32	4.50		
		V_{det2_3}	2.81	2.93	3.05		
		V_{det2_4}	2.76	2.88	3.00		
Internal reset time	Power-on reset time	t_{POR}	—	15.5	—	ms	Figure 49.78
	LVD0 reset time	t_{LVD0}	—	0.70	—		Figure 49.79
	LVD1 reset time	t_{LVD1}	—	0.57	—		Figure 49.80
	LVD2 reset time	t_{LVD2}	—	0.57	—		Figure 49.81
Minimum VCC down time		t_{VOFF}	200	—	—	μs	Figure 49.78, Figure 49.79
Response delay time		t_{det}	—	—	200	μs	Figure 49.78 to Figure 49.81
LVD operation stabilization time (after LVD is enabled)		$T_{d(E-A)}$	—	—	20	μs	Figure 49.80, Figure 49.81
Hysteresis width (LVD1 and LVD2)		V_{LVH}	—	80	—	mV	

Note: The minimum VCC down time indicates the time when VCC is below the minimum value of voltage detection levels V_{POR}, V_{det1}, and V_{det2} for the POR/ LVD.

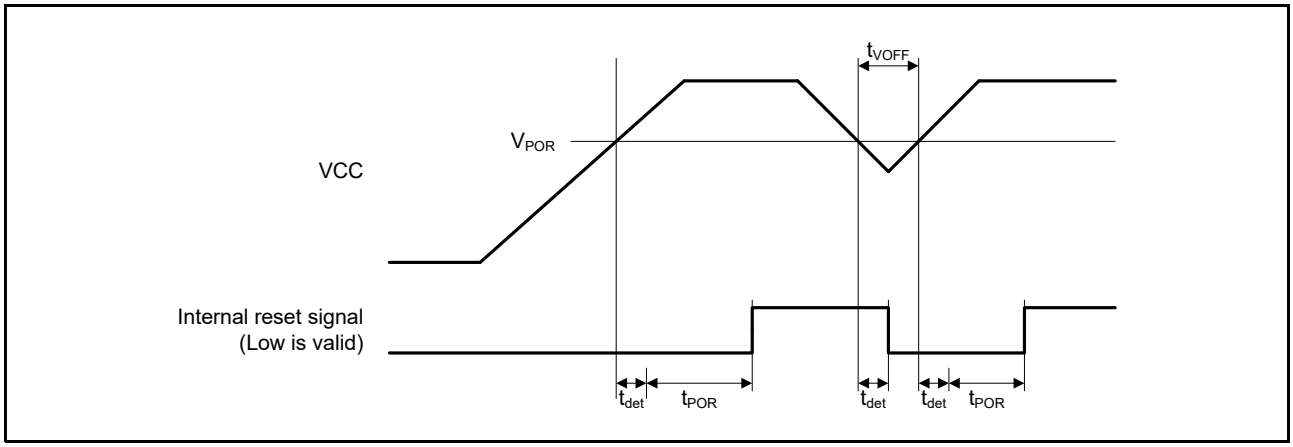


Figure 49.78 Power-on Reset Timing

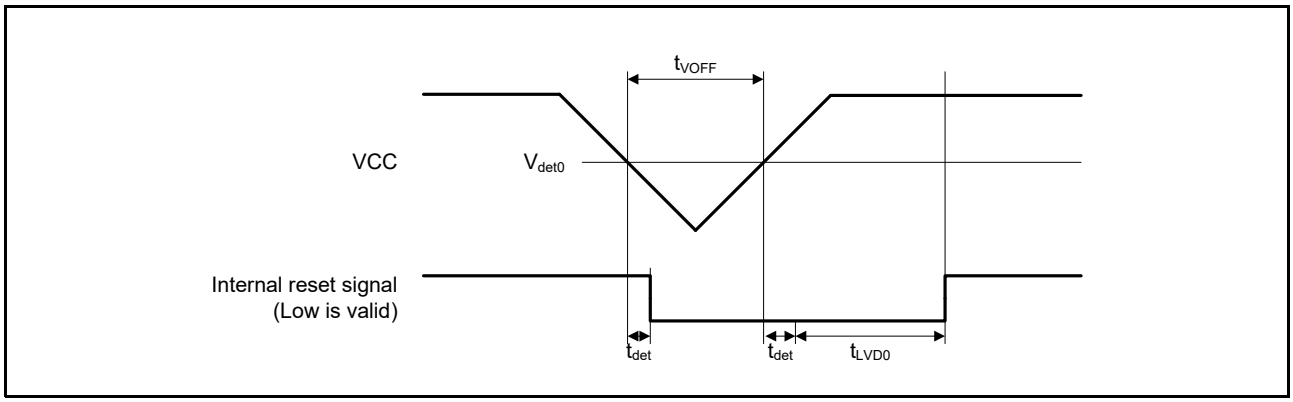


Figure 49.79 Voltage Detection Circuit Timing (V_{det0})

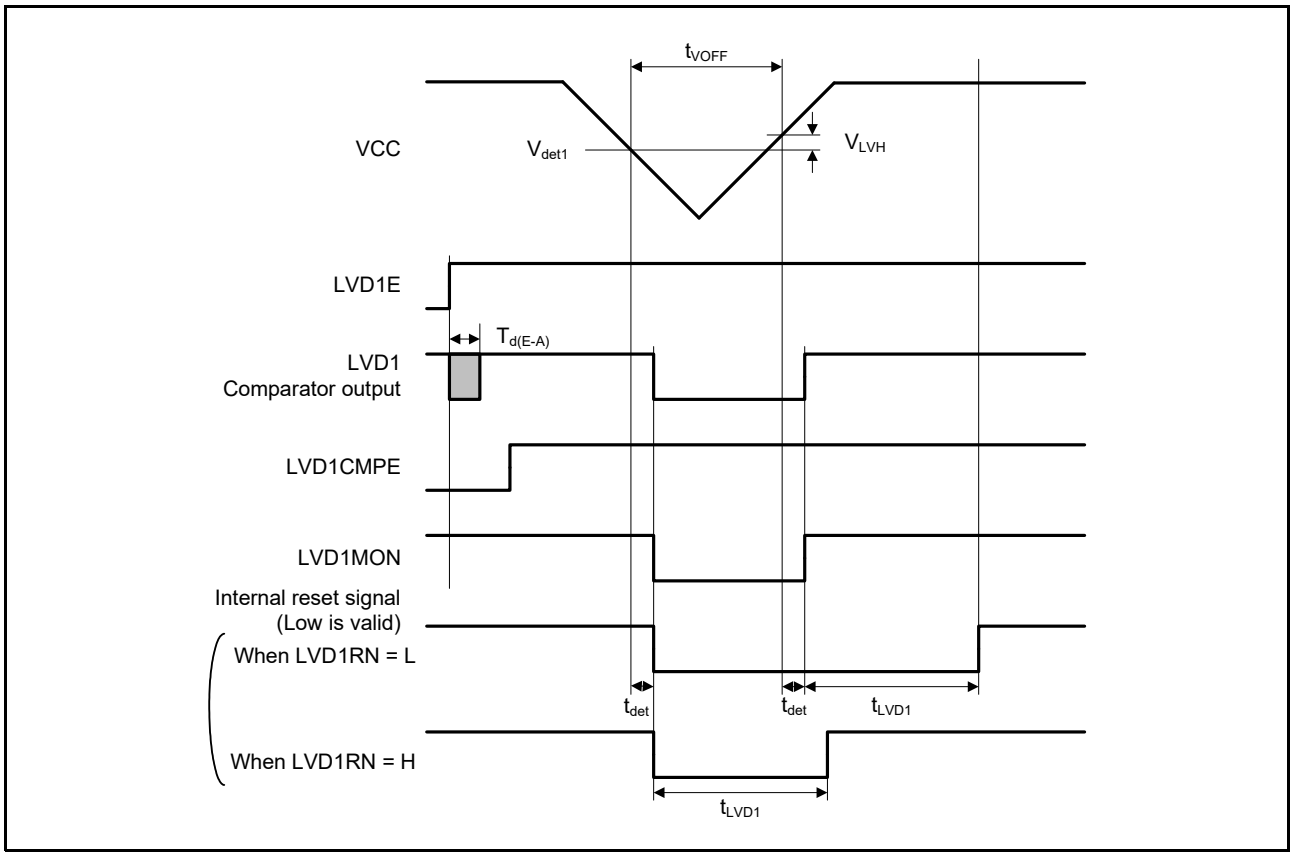


Figure 49.80 Voltage Detection Circuit Timing (V_{det1})

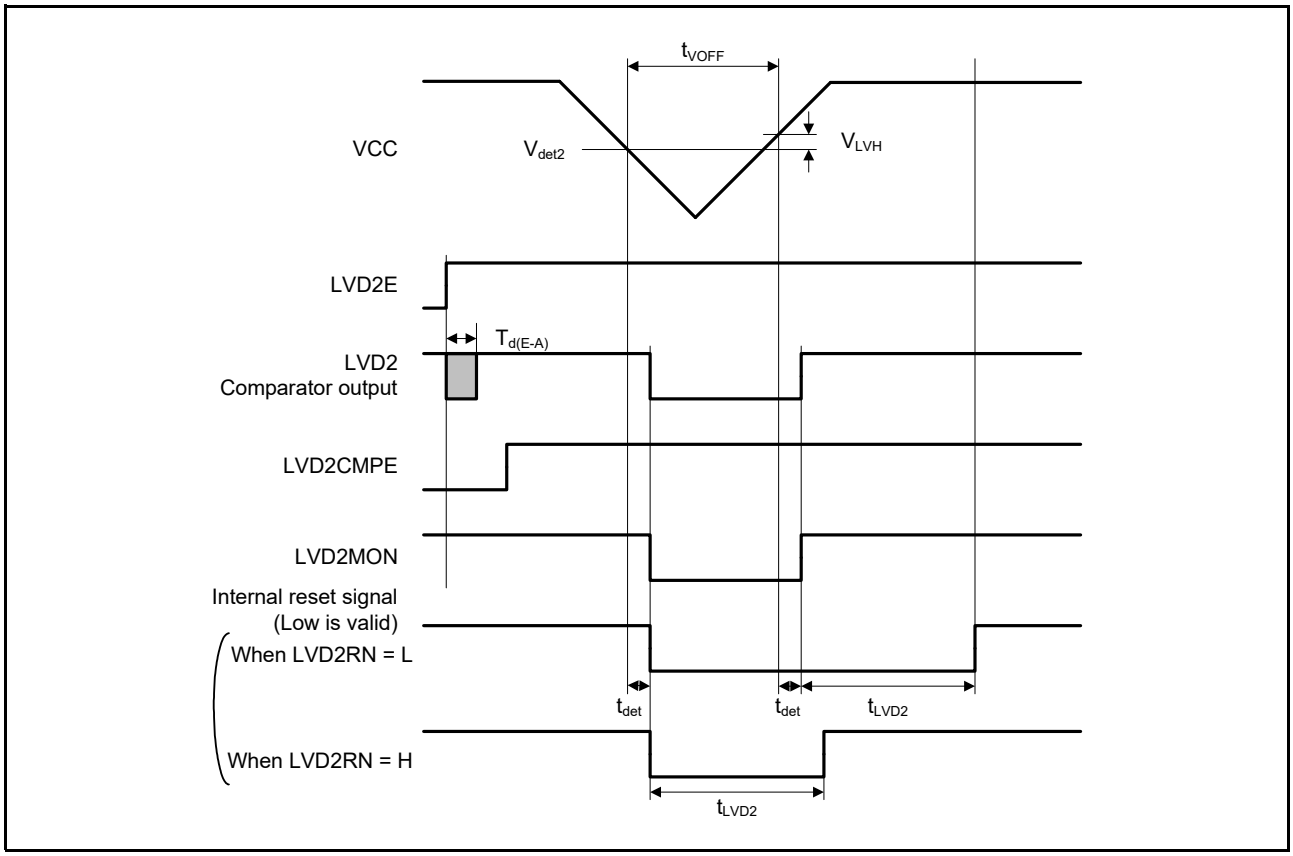


Figure 49.81 Voltage Detection Circuit Timing (V_{det2})

49.11 Oscillation Stop Detection Timing

Table 49.52 Oscillation Stop Detection Circuit Characteristics

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = AVCC1 = AVCC2 = 3.0 to 5.5 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V, $T_a = T_{opr}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Detection time	t_{dr}	—	—	1	ms	Figure 49.82

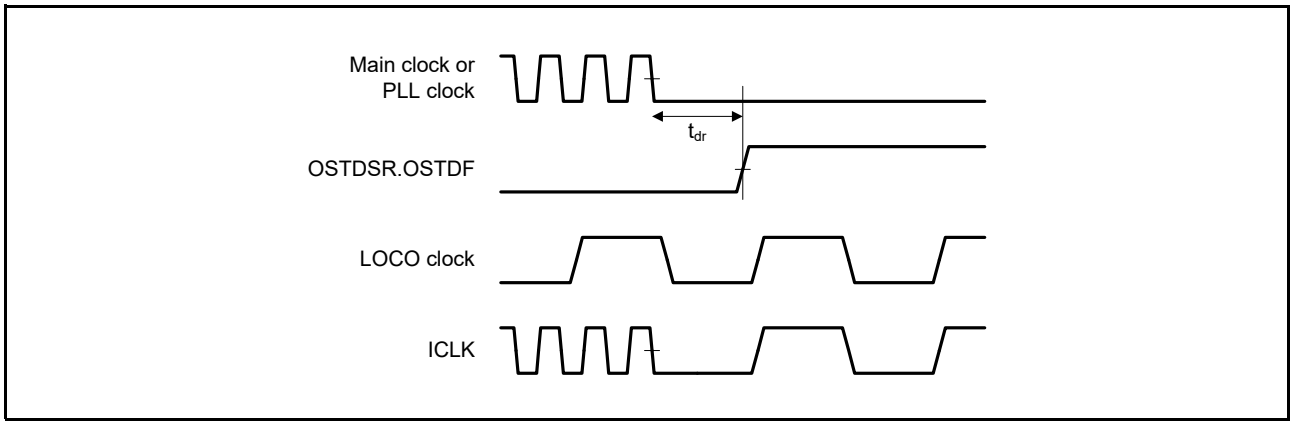


Figure 49.82 Oscillation Stop Detection Timing

49.12 Flash Memory Characteristics

Table 49.53 Code Flash Memory Characteristics

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = AVCC1 = AVCC2 = 3.0 to 5.5 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V,
Temperature range for program/erase: $T_a = T_{opr}$

Item		Symbol	FCLK = 4 MHz			20 MHz ≤ FCLK ≤ 60 MHz			Unit	Test Conditions
			Min.	Typ.	Max.	Min.	Typ.	Max.		
Program time (N_{PEC} ≤ 100 cycles)	128 bytes	t_{P128}	—	0.66	11	—	0.3	5	ms	
	8 Kbytes	t_{P8K}	—	37	176	—	17	80		
	32 Kbytes	t_{P32K}	—	150	704	—	68	320		
Program time (N_{PEC} > 100 cycles)	128 bytes	t_{P128}	—	0.71	13	—	0.32	6	ms	
	8 Kbytes	t_{P8K}	—	46	212	—	21	96		
	32 Kbytes	t_{P32K}	—	185	848	—	84	384		
Erase time (N_{PEC} ≤ 100 cycles)	4 Kbytes	t_{E4K}	—	43	108	—	24	60	ms	
	32 Kbytes	t_{E32K}	—	284	864	—	158	480		
Erase time (N_{PEC} > 100 cycles)	4 Kbytes	t_{E4K}	—	50	130	—	28	72	ms	
	32 Kbytes	t_{E32K}	—	338	864	—	188	480		
Program/erase cycles*1		N_{PEC}	1000*2	—	—	1000*2	—	—	Cycles	
Program suspend latency		t_{SPD}	—	—	264	—	—	120	µs	
Primary erase suspend latency in suspend priority mode		t_{SESD1}	—	—	216	—	—	120		
Secondary erase suspend latency in suspend priority mode		t_{SESD2}	—	—	1.7	—	—	1.7	ms	
Erase suspend latency in erase priority mode		t_{SEED}	—	—	1.7	—	—	1.7	ms	
Forced stop command		t_{FD}	—	—	32	—	—	20	µs	
Data retention*3, *4		t_{DRP}	20	—	—	20	—	—	Year	T_a ≤ 85°C
			10	—	—	10	—	—		T_a ≤ 105°C

Note 1. Definition of program/erase cycle:
The program/erase cycle is the number of erasing for each block. When the number of program/erase cycles is n, each block can be erased n times. For instance, when 256-byte program is performed 32 times for different addresses in 8-Kbyte block and then the block is erased, the program/erase cycle is counted as one. However, the same address cannot be programmed more than once before the next erase cycle (overwriting is prohibited).

Note 2. Characteristics are degraded as the number of program/erase increases. This is the minimum value of program/erase cycles to guarantee all characteristics listed in this table.

Note 3. This shows the characteristic when the flash memory writer or self-programming library from Renesas Electronics is in use, and the number of times programming and erasure proceed does not exceed the specified value.

Note 4. These values are based on the results of reliability testing.

Table 49.54 Data Flash Memory Characteristics

Conditions: VCC = 2.7 to 5.5 V, AVCC0 = AVCC1 = AVCC2 = 3.0 to 5.5 V,
VSS = AVSS0 = AVSS1 = AVSS2 = 0 V,
Temperature range for program/erase: $T_a = T_{opr}$

Item		Symbol	FCLK = 4 MHz			20 MHz ≤ FCLK ≤ 60 MHz			Unit	Test Conditions
			Min.	Typ.	Max.	Min.	Typ.	Max.		
Program time	4 bytes	t_{DP4}	—	0.36	3.8	—	0.16	1.7	ms	
Erase time	64 bytes	t_{DE64}	—	3.1	18	—	1.7	10		
Blank check time	4 bytes	t_{DBC4}	—	—	84	—	—	30	µs	
	64 bytes	t_{DBC64}	—	—	280	—	—	100		
	2 Kbytes	t_{DBC2K}	—	—	6160	—	—	2200		
Program/erase cycles*1		N_{DPEC}	100000 *2	—	—	100000 *2	—	—	Cycles	
Program suspend latency		t_{DSPD}	—	—	264	—	—	120	µs	
Primary erase suspend latency in suspend priority mode		t_{DSESD1}	—	—	216	—	—	120		
Secondary erase suspend latency in suspend priority mode		t_{DSESD2}	—	—	300	—	—	300		
Erase suspend latency in erase priority mode		t_{DSEED}	—	—	300	—	—	300		
Forced stop command		t_{FD}	—	—	32	—	—	20		
Data retention*3, *4		t_{DDRP}	20	—	—	20	—	—	Year	$T_a \leq 85°C$
			10	—	—	10	—	—		$T_a \leq 105°C$

Note 1. Definition of program/erase cycle:
The program/erase cycle is the number of erasing for each block. When the number of program/erase cycles is n, each block can be erased n times. For instance, when 4-byte program is performed 512 times for different addresses in 2-Kbyte block and then the block is erased, the program/erase cycle is counted as one. However, the same address cannot be programmed more than once before the next erase cycle (overwriting is prohibited).

Note 2. Characteristics are degraded as the number of program/erase increases. This is the minimum value of program/erase cycles to guarantee all characteristics listed in this table.

Note 3. This shows the characteristic when the flash memory writer or self-programming library from Renesas Electronics is in use, and the number of times programming and erasure proceed does not exceed the specified value.

Note 4. These values are based on the results of reliability testing.

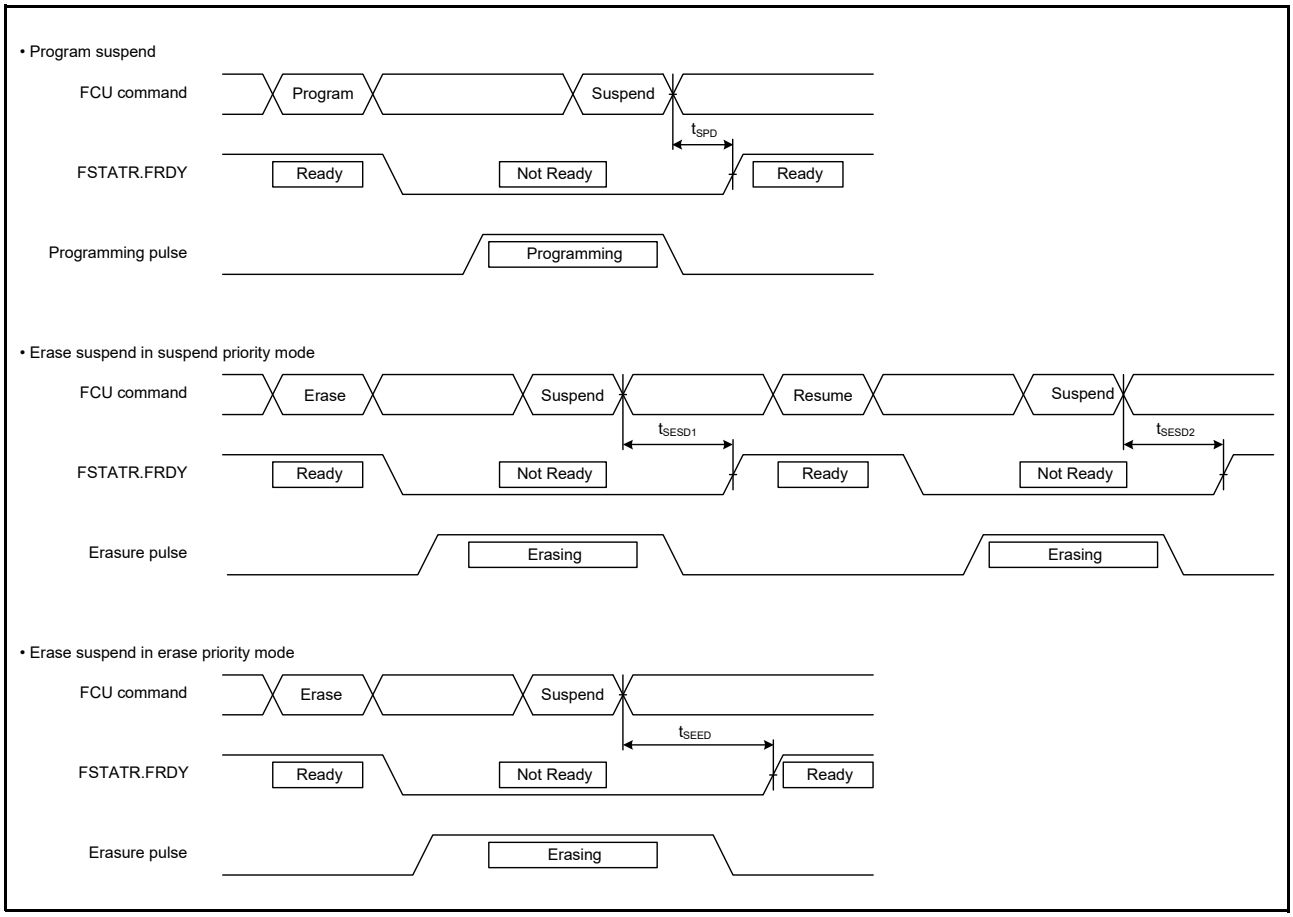


Figure 49.83 Flash Memory Program/Erase Suspend Timing

Appendix 1. Port States in Each Processing Mode

Table 1.1 Port States in Each Processing State (1 / 2)

Port Name Pin Name	Operating Mode According to Registers Setting		Reset	Software Standby Mode
P00/IRQ2, P01/IRQ4	Single-chip mode		Hi-Z	Keep-O*1
P10/IRQ0, P11/IRQ1	Single-chip mode		Hi-Z	Keep-O*1
P20/IRQ7, P21/IRQ6, P22/IRQ10, P23/IRQ11, P24/IRQ4, P27/IRQ15	Single-chip mode		Hi-Z	Keep-O*1
P30/IRQ7, P31/IRQ6, P32/IRQ12, P33/IRQ13	Single-chip mode		Hi-Z	Keep-O*1
P36, P37	Single-chip mode		Hi-Z	Keep-O
P40, P41, P42, P43, P44, P45, P46, P47	Single-chip mode		Hi-Z	Keep-O
P50, P51	Single-chip mode		Hi-Z	Keep-O
P52/IRQ0, P53/IRQ1, P54/IRQ2, P55/IRQ3	Single-chip mode		Hi-Z	Keep-O*1
P60/IRQ4, P61/IRQ5, P62/IRQ6, P63/IRQ7	Single-chip mode		Hi-Z	Keep-O*1
P64/IRQ8	Single-chip mode	DA0 output (DAOE0 = 1)	Hi-Z	DA output retained
		Other than the above (DAOE0 = 0)		Keep-O*1
P65/IRQ9	Single-chip mode	DA1 output (DAOE1 = 1)	Hi-Z	DA output retained
		Other than the above (DAOE1 = 0)		Keep-O*1
P70/IRQ5	Single-chip mode		Hi-Z	Keep-O*1
P71, P72, P73, P74, P75, P76	Single-chip mode		Hi-Z	Keep-O
P80/IRQ5	Single-chip mode		Hi-Z	Keep-O*1
P81	Single-chip mode		Hi-Z	Keep-O
P82/IRQ3	Single-chip mode		Hi-Z	Keep-O*1

Table 1.1 Port States in Each Processing State (2 / 2)

Port Name Pin Name	Operating Mode According to Registers Setting	Reset	Software Standby Mode
P90, P91, P92	Single-chip mode	Hi-Z	Keep-O
P93/IRQ14	Single-chip mode	Hi-Z	Keep-O*1
P94	Single-chip mode	Hi-Z	Keep-O
P95/IRQ1, P96/IRQ4	Single-chip mode	Hi-Z	Keep-O*1
PA0	Single-chip mode	Hi-Z	Keep-O
PA1/IRQ14	Single-chip mode	Hi-Z	Keep-O*1
PA2, PA3, PA4	Single-chip mode	Hi-Z	Keep-O
PA5/IRQ1	Single-chip mode	Hi-Z	Keep-O*1
PB0/IRQ8, PB1/IRQ4	Single-chip mode	Hi-Z	Keep-O*1
PB2	Single-chip mode	Hi-Z	Keep-O
PB3/IRQ9, PB4/IRQ3	Single-chip mode	Hi-Z	Keep-O*1
PB5	Single-chip mode	Hi-Z	Keep-O
PB6/IRQ2	Single-chip mode	Hi-Z	Keep-O*1
PB7	Single-chip mode	Hi-Z	Keep-O
PD0, PD1, PD2, PD3	Single-chip mode	Hi-Z	Keep-O
PD4/IRQ2, PD5/IRQ6, PD6/IRQ5, PD7/IRQ8	Single-chip mode	Hi-Z	Keep-O*1
PE0/IRQ7, PE1/IRQ15	Single-chip mode	Hi-Z	Keep-O*1
PE2/NMI/ IRQ0	Single-chip mode	Hi-Z	Keep-O*1
PE3/IRQ2, PE4/IRQ1, PE5/IRQ0	Single-chip mode	Hi-Z	Keep-O*1
PN6	Single-chip mode	Pull up*2	Keep-O
PN7/IRQ5	Single-chip mode	Pull down*3	Keep-O*1

Keep-O: Output pins retain their previous values, and input pins become high-impedance. Note that the input pull-up resistor remains enabled when it is enabled by the pull-up resistor control register.

Hi-Z: High-impedance

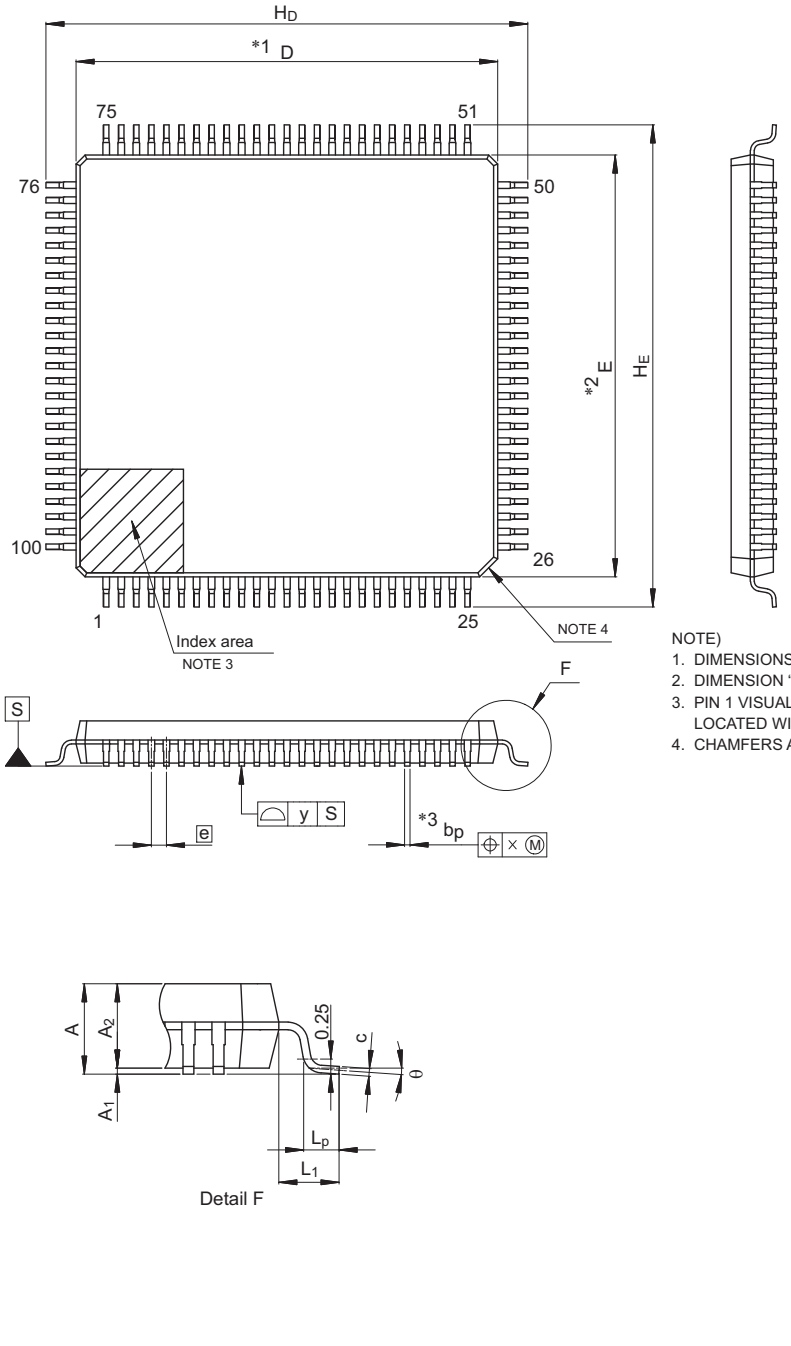
Note 1. Input is enabled if the pin is specified as the software standby cancelling source while it is used as an external interrupt pin.
Note 2. This pin functions as MD after release from the reset state, and the pull-up resistor connected to the MD pin is enabled.
Note 3. This pin functions as EMLE after release from the reset state, and the pull-down resistor connected to the EMLE pin is enabled.

Appendix 2. Package Dimensions

Information on the latest version of the package dimensions or mountings has been displayed in “Packages” on Renesas Electronics Corporation website.

JEITA Package Code	RENESAS Code	Previous Code	MASS (Typ) [g]
P-LFQFP100-14x14-0.50	PLQP0100KB-B	—	0.6

Unit: mm



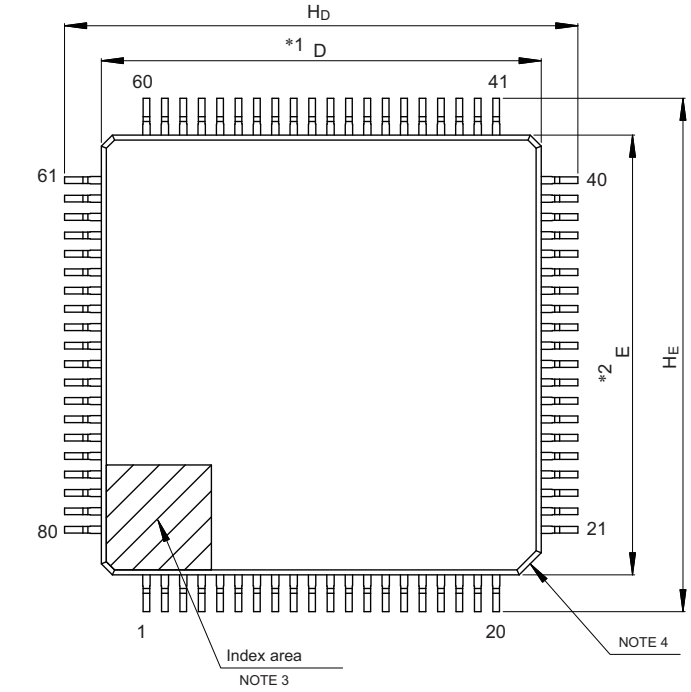
NOTE)
1. DIMENSIONS “*1” AND “*2” DO NOT INCLUDE MOLD FLASH.
2. DIMENSION “*3” DOES NOT INCLUDE TRIM OFFSET.
3. PIN 1 VISUAL INDEX FEATURE MAY VARY, BUT MUST BE LOCATED WITHIN THE HATCHED AREA.
4. CHAMFERS AT CORNERS ARE OPTIONAL, SIZE MAY VARY.

Reference Symbol	Dimensions in millimeters		
	Min	Nom	Max
D	13.9	14.0	14.1
E	13.9	14.0	14.1
A_2	—	1.4	—
H_D	15.8	16.0	16.2
H_E	15.8	16.0	16.2
A	—	—	1.7
A_1	0.05	—	0.15
b_p	0.15	0.20	0.27
c	0.09	—	0.20
θ	0°	3.5°	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.08
L_p	0.45	0.6	0.75
L_1	—	1.0	—

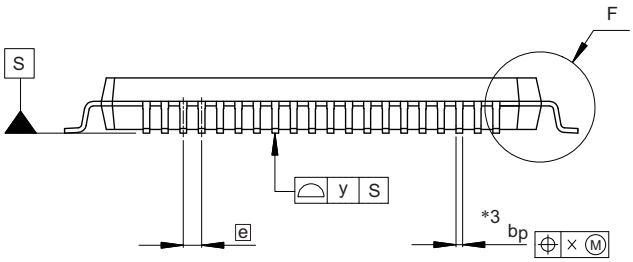
Figure A 100-Pin LFQFP (PLQP0100KB-B)

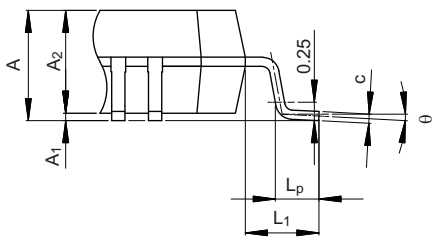
JEITA Package Code	RENESAS Code	Previous Code	MASS (Typ) [g]
P-LFQFP80-12x12-0.50	PLQP0080KB-B	—	0.5

Unit: mm



NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.
3. PIN 1 VISUAL INDEX FEATURE MAY VARY, BUT MUST BE LOCATED WITHIN THE HATCHED AREA.
4. CHAMFERS AT CORNERS ARE OPTIONAL, SIZE MAY VARY.





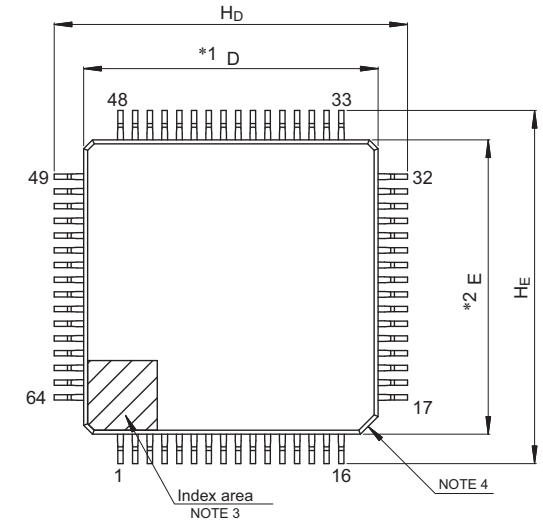
Detail F

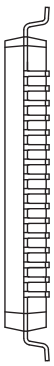
Reference Symbol	Dimensions in millimeters		
	Min	Nom	Max
D	11.9	12.0	12.1
E	11.9	12.0	12.1
A_2	—	1.4	—
H_D	13.8	14.0	14.2
H_E	13.8	14.0	14.2
A	—	—	1.7
A_1	0.05	—	0.15
b_p	0.15	0.20	0.27
c	0.09	—	0.20
θ	0°	3.5°	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.08
L_p	0.45	0.6	0.75
L_1	—	1.0	—

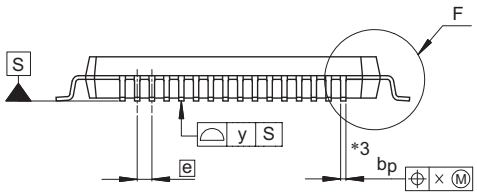
Figure B 80-Pin LFQFP (PLQP0080KB-B)

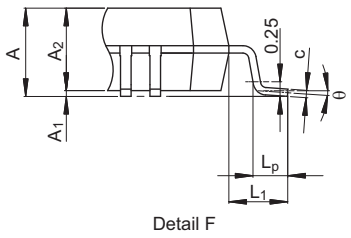
JEITA Package Code	RENESAS Code	Previous Code	MASS (Typ) [g]
P-LFQFP64-10x10-0.50	PLQP0064KB-C	—	0.3

Unit: mm









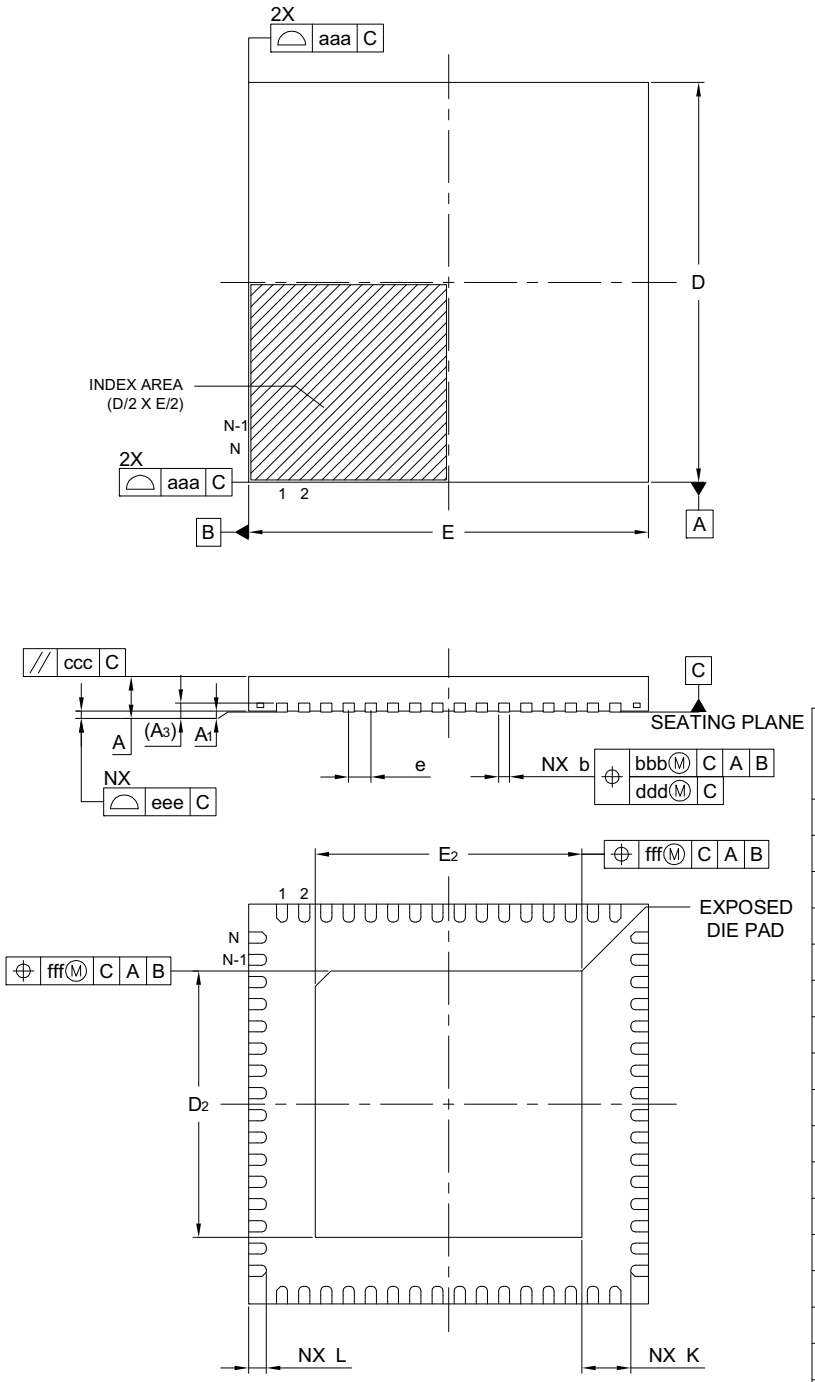
Detail F

NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.
3. PIN 1 VISUAL INDEX FEATURE MAY VARY, BUT MUST BE LOCATED WITHIN THE HATCHED AREA.
4. CHAMFERS AT CORNERS ARE OPTIONAL, SIZE MAY VARY.

Reference Symbol	Dimensions in millimeters		
	Min	Nom	Max
D	9.9	10.0	10.1
E	9.9	10.0	10.1
A_2	—	1.4	—
H_D	11.8	12.0	12.2
H_E	11.8	12.0	12.2
A	—	—	1.7
A_1	0.05	—	0.15
b_p	0.15	0.20	0.27
c	0.09	—	0.20
θ	0°	3.5°	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.08
L_p	0.45	0.6	0.75
L_1	—	1.0	—

Figure C 64-Pin LFQFP (PLQP0064KB-C)

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-HWQFN064-9x9-0.50	PWQN0064KF-A	0.17

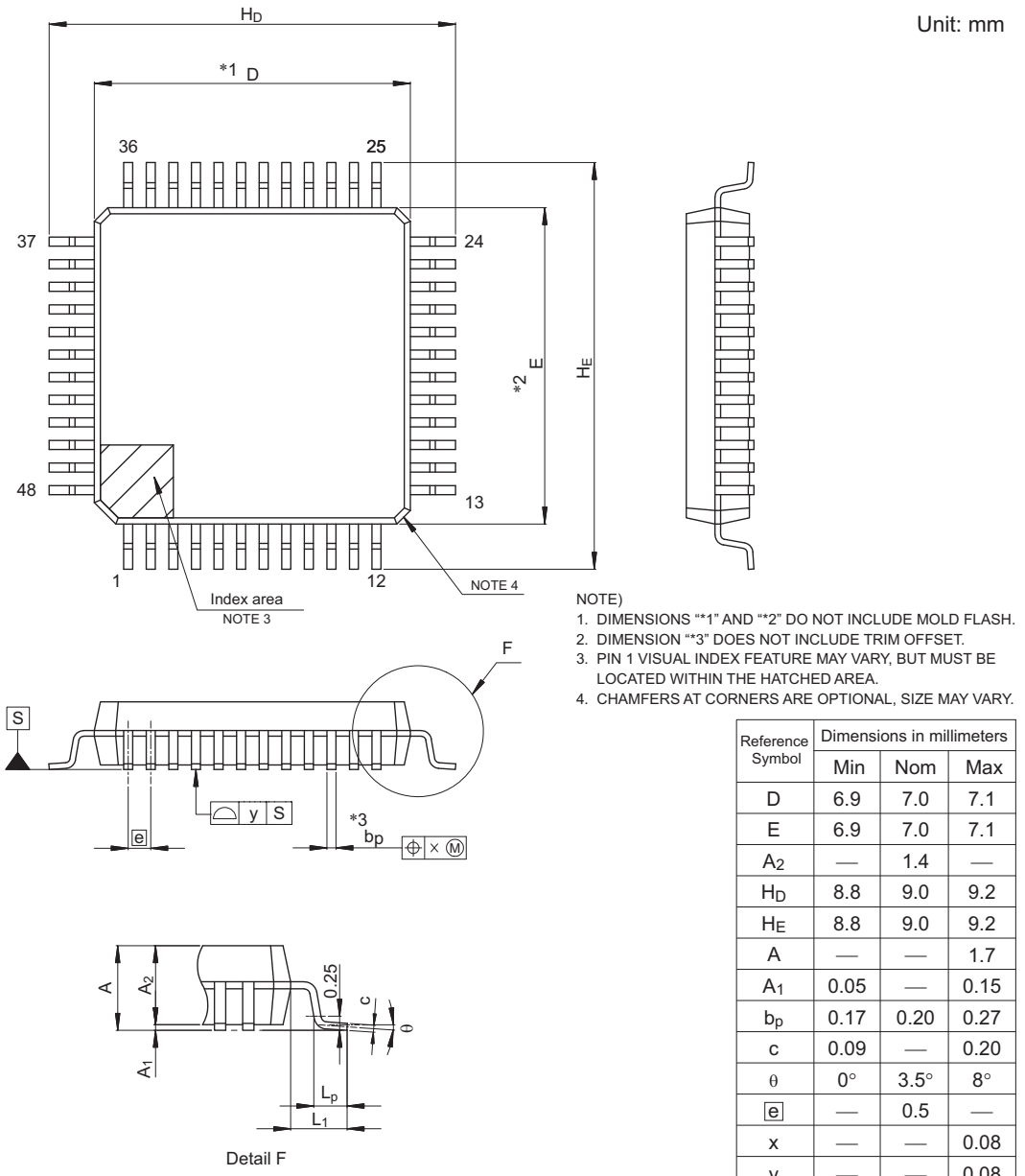


Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	—	—	0.80
A_1	0.00	0.02	0.05
A_3		0.203 REF.	
b	0.18	0.25	0.30
D		9.00 BSC	
E		9.00 BSC	
e		0.50 BSC	
N		64	
L	0.35	0.40	0.45
K	0.20	—	—
D_2	5.95	6.00	6.05
E_2	5.95	6.00	6.05
aaa	—	—	0.15
bbb	—	—	0.10
ccc	—	—	0.10
ddd	—	—	0.05
eee	—	—	0.08
fff	—	—	0.10

Figure D 64-Pin HWQFN (PWQN0064KF-A)

JEITA Package Code	RENESAS Code	Previous Code	MASS (Typ) [g]
P-LFQFP48-7x7-0.50	PLQP0048KB-B	—	0.2

Unit: mm

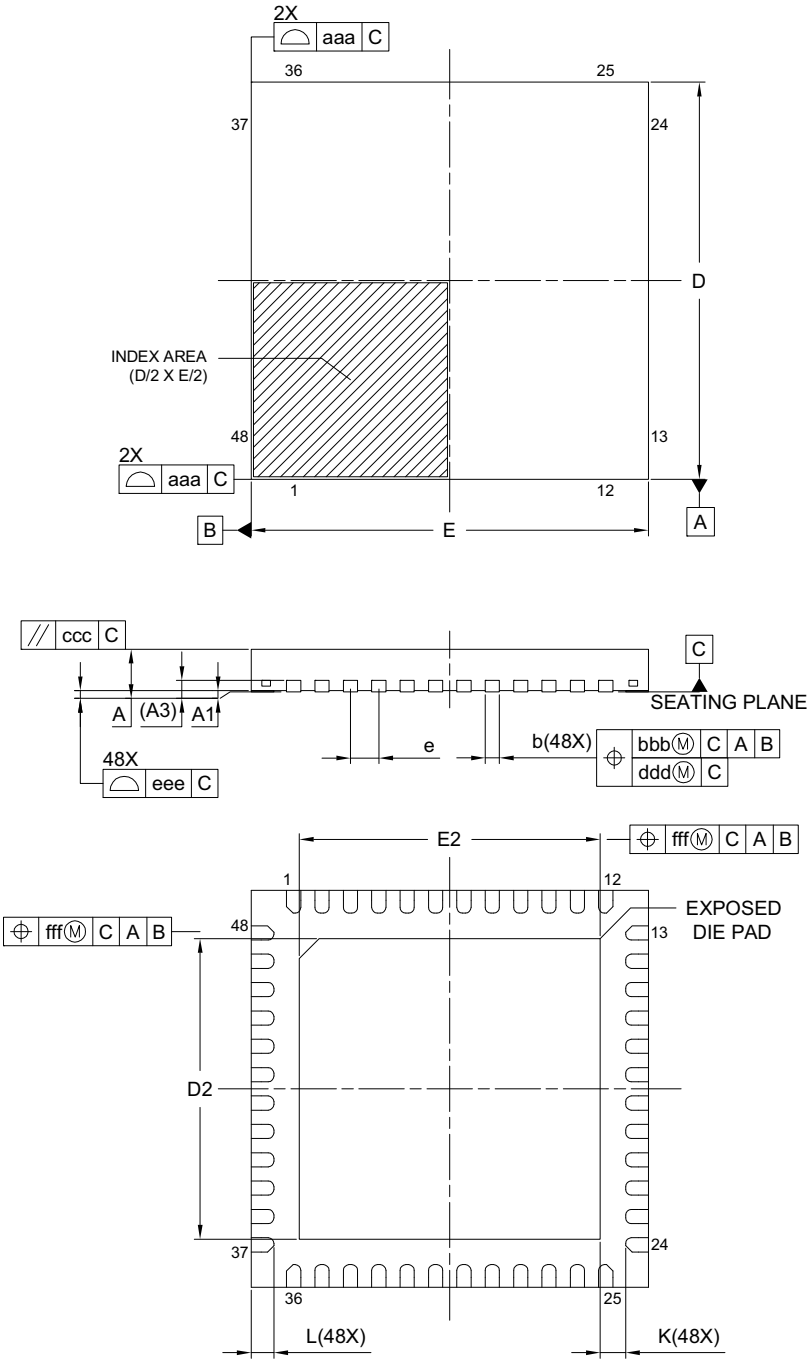


NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.
3. PIN 1 VISUAL INDEX FEATURE MAY VARY, BUT MUST BE LOCATED WITHIN THE HATCHED AREA.
4. CHAMFERS AT CORNERS ARE OPTIONAL, SIZE MAY VARY.

Reference Symbol	Dimensions in millimeters		
	Min	Nom	Max
D	6.9	7.0	7.1
E	6.9	7.0	7.1
A_2	—	1.4	—
H_D	8.8	9.0	9.2
H_E	8.8	9.0	9.2
A	—	—	1.7
A_1	0.05	—	0.15
b_p	0.17	0.20	0.27
c	0.09	—	0.20
θ	0°	3.5°	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.08
L_p	0.45	0.6	0.75
L_1	—	1.0	—

Figure E 48-Pin LFQFP (PLQP0048KB-B)

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-HWQFN048-7x7-0.50	PWQN0048KC-A	0.13 g



Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	—	—	0.80
A_1	0.00	0.02	0.05
A_3		0.203 REF.	
b	0.20	0.25	0.30
D		7.00 BSC	
E		7.00 BSC	
e		0.50 BSC	
L	0.30	0.40	0.50
K	0.20	—	—
D_2	5.25	5.30	5.35
E_2	5.25	5.30	5.35
aaa		0.15	
bbb		0.10	
ccc		0.10	
ddd		0.05	
eee		0.08	
fff		0.10	

Figure F 48-Pin HWQFN (PWQN0048KC-A)

REVISION HISTORY	RX26T Group User's Manual: Hardware

Classifications

- Items with Technical Update document number: Changes according to the corresponding issued Technical Update
- Items without Technical Update document number: Minor changes that do not require Technical Update to be issued

Rev.	Date	Description		Classification
		Page	Summary	
1.00	Jan 16, 2023	—	First edition, issued	
1.01	Mar 01, 2023	Features		
		62	Package, changed	
		1. Overview		
		74 to 76	Table 1.3 List of Products, changed	
		7. Option-Setting Memory (OFSM)		
		230	7.2.9 Flash Access Window Setting Register (FAW), changed	
		16. Memory-Protection Unit (MPU)		
		423	16.4.4 Processing in Response to Memory-Protection Errors (2) When an instruction memory-protection error is generated, changed	
		24. General PWM Timer (GPTWa)		
		1001	Table 24.2 GPTW Functions (n = 0 to 7), changed	
		44. Temperature Sensor (TEMPS)		
		2674	44.3.2 Setting of 12-Bit A/D Converter Unit 2, changed	
		2675	Figure 44.3 Procedure for Using the Temperature Sensor, changed	
		45. Comparator C (CMPCa)		
		2688	Figure 45.6 Comparator Operation Termination Flowchart, changed	
		49. Electrical Characteristics		
		2832	Table 49.10 Permissible Output Currents, changed	
		2835	Table 49.13 Thermal Resistance Value (Reference), changed	
1.10	Aug 10, 2023	1. Overview		
		71	Table 1.1 Outline of Specifications (9/9), changed	
		74 to 76	Table 1.3 List of Products, changed	
		77	Figure 1.1 How to Read the Product Part Number, changed	
		8. Voltage Detection Circuit (LVDA)		
		236	Table 8.1 Voltage Detection Circuit Specifications, changed	
		19. Event Link Controller (ELC)		
		514	Table 19.2 Correspondence between the ELSRn Register and the Peripheral Modules, changed	
		—	19.4.1 Setting ELSRn Register (2) Setting ELSR24, ELSR25, ELSR26, and ELSR27 Registers, deleted	
		22. Multi-Function Timer Pulse Unit 3 (MTU3d)		
		735	Figure 22.47 Example of Complementary PWM Mode Setting Procedure, changed	
		24. General PWM Timer (GPTWa)		
		1017 to 1020	24.2.7 General PWM Timer Clear Source Select Register (GTCSR), changed	TN-RX*-A0268A/E
		1033 to 1038	24.2.12 General PWM Timer Control Register (GTCR), changed	
		1085, 1086	24.2.37 General PWM Timer Inter Channel Logical Operation Function Setting Register (GTICLF), changed	
		1094 to 1096	24.2.42 General PWM Timer Operation Enable Bit Simultaneous Control Register (GTSECR), changed	TN-RX*-A0268A/E
		1105 to 1107	24.2.46 Output Phase Switching Control Register (OPSCR), changed	
		1124	Table 24.11 The GTPR Buffer Transfer Timing in Complementary PWM Mode, changed	TN-RX*-A0268A/E
		1184	Table 24.17 Immediate Single Buffer Transfer from the GTCCRD Register in Complementary PWM Mode 4 (1), changed	
		1225	Table 24.23 Settings of the Registers Used to Set Sources for Synchronous Clearing, added	

Rev.	Date	Description		Classification
		Page	Summary	
1.10	Aug 10, 2023	1303	24.10.3 Range of Settings for the GTPBR and GTPDBR Registers in the Complementary PWM Mode, added	TN-RX*-A0268A/E
		1305	24.10.7 Note on Counter Clearing in the Complementary PWM Mode, added	
			24.10.8 Note on Disabling PWM Initial Output After Synchronous Clearing in the Complementary PWM Mode, added	
		25. High Resolution PWM Waveform Generation Circuit (HRPWM)		
		1318	25.4.3 Notes on Delay Settings for HRPWM, changed	
		26. GPTW Port Output Enable (POEG)		
		1321	Figure 26.2 POEG Block Diagram, changed	
		1328	26.3.1 Request to Stop Output in Response to Detection of Active Level or Edge Input on the Corresponding GTETRGn Pin (n = A to D), changed	
		1332	26.3.7 Requests to Stop Output in Response to Detected Signals and Canceling the Requests, changed	
		32. Serial Communications Interface (SCIk, SCIh)		
		1470	Table 32.16 Maximum Bit Rate for Each Operating Frequency (Asynchronous Mode) (for SCI1, SCI5, and SCI6), added	
		33. Serial Communications Interface (RSCI)		
		1603	Table 33.1 RSCI Specifications (3/3), changed	TN-RX*-A0268A/E
		1628	Table 33.18 Examples of BRR[7:0] Bits Settings for Various Bit Rates (Clock Synchronous Mode, Simple SPI Mode), changed	
		1650	33.2.13 DE Signal Control Register (DECR), changed	
		1807	33.14 RS-485 Driver Control Function, changed	
		1816	Table 33.49 RSCI Event Link Signal List, changed	TN-RX*-A0268A/E
		—	33.18 Event Linking (5) Receive Data Match Event Output, deleted	
		—	(6) Receive Data Unmatch Event Output, deleted	
		—	(7) Active Edge Detection Event Output, deleted	
		35. I3C Bus Interface (RI3C)		
		1981, 1982	35.3.1.4 Combo Transfer Command, changed	
		1985	35.3.2 Response Descriptor (1) I3C Controller Mode, changed	
		1996	Figure 35.11 Bus Conditions, changed	
		1997	Figure 35.12 Example of RI3C Initialization Flowchart, changed	
		1998	Figure 35.13 I3C Communication Flow, changed	
		2008	Figure 35.28 I3C Controller CRR Processing Flow, changed	
		2019	Figure 35.46 I3C Target CRR Processing Flow, changed	
		2021	Figure 35.48 I3C Target Receives GET CCC Command while Data Exists by Writing from the ICDR Register to the Transmit Buffer, changed	TN-RX*-A0268A/E
		2035	Figure 35.63 Controller Role Request: ACK, changed	
			Figure 35.64 Controller Role Request: NACK (Not Match the DADR[7:0] Bits) and RCRRN = 1, changed	
		2036	Figure 35.65 Controller Role Request: NACK (ROLE[1:0] = 00b (I3C Target)) and RCRRN = 0, changed	
			Figure 35.66 Controller Role Request: NACK (CRRRJ = 1) and RCRRN = 1 (1/2), changed	
		2037	Figure 35.67 Controller Role Request: NACK (CRRRJ = 1) and RCRRN = 1 (2/2), changed	
		36. CAN FD Module (CANFD)		
		2055 to 2058	36.2.2 Channel Control Register (CHCR), changed	
		2241	Figure 36.58 Global Interrupt Block Diagram, changed	
		2242	Figure 36.59 Channel Interrupt Block Diagram, changed	
		40. Arithmetic Unit for Trigonometric Functions (TFUv2)		
		2468	Table 40.1 TFU specifications, changed	
		2469	40.2.1 Fixed-Point Sincos Input and Output Setting Register (FXSCIOC), changed	
		2470	40.2.2 Fixed-Point Atanhypot_k Input and Output Setting Register (FXATIOC), changed	
		2475	Table 40.4 Input/Output value of FXSCDT0 and FXSCDT1, changed	
		2477	Table 40.5 Input/Output value of FXATDT0 and FXATDT1, changed	

Rev.	Date	Description		Classification
		Page	Summary	
1.10	Aug 10, 2023	2479, 2480	40.3.2.1 Single-Precision Floating-Point Numbers, changed	
		2481	40.3.3 Units of Angles, changed	
		42. 12-Bit A/D Converter (S12ADHa)		
		2604	42.3.3.4 Channel Selection and Self-Diagnosis (Without Channel-Dedicated Sample-and-Hold Circuits), changed	
		48. Flash Memory (FLASH)		
		2713	48.1 Overview, changed	
		2714	Figure 48.1 Block Diagram of Flash Memory Related Modules, changed	
		2723	48.4.5 FACI Command Processing Start Address Register (FSADDR), changed	
		2724	48.4.6 FACI Command Processing End Address Register (FEADDR), changed	
		2781	Table 48.17 Address Used by Configuration Set Command, changed	
		2826, 2827	48.8.28 Rewriting Flash Memory When Using Dual Mode, added	TN-RX*-A0267A/E
		2830	Table 48.24 Conditions under which Background Operation is Usable, changed	
		49. Electrical Characteristics		
		2835	Table 49.6 DC Characteristics (3) (Products with 64 Kbytes of RAM), changed	
		2836	Table 49.7 DC Characteristics (3) (Products with 48 Kbytes of RAM), changed	

RX26T Group User's Manual: Hardware

Publication Date: Rev.1.00 Jan 16, 2023
Rev.1.10 Aug 10, 2023

Published by: Renesas Electronics Corporation

RX26T Group

Renesas Electronics Corporation

R01UH0979EJ0110